TRS8BIT



BACK IN THE LATE 1970'S, I BUMPED INTO AN OLD FRIEND, BRIAN PAIN, WHO MENTIONED THAT HE HAD JUST PURCHASED A TRS-80, MODEL 1, AND WITH ANOTHER COLLEAGUE, LEON HELLER, WAS SETTING UP A USER-GROUP FOR ANYONE INTERESTED IN TRS-80'S. HAVING JUST BOUGHT A MODEL 1, I JUMPED AT THE OPPORTUNITY TO JOIN WITH A GROUP OF LIKE-MINDED INDIVIDUALS.
ORIGINALLY CALLED THE NATIONAL TRS-80 USER GROUP, THE GROUP WENT FROM STRENGTH TO STRENGTH AND BY THE MID 80'S HAD OVER 900 MEMBERS. WORKSHOPS WERE HELD ON A REGULAR BASIS THROUGHOUT THE COUNTRY, TOGETHER WITH PUBLISHING A MONTHLY NEWSLETTER, 'NATGUG NEWS'.

30 YEARS ON, I ASSUMED THAT ALL THINGS 'TRS-80' HAD LONG SINCE PASSED AWAY, AND I, ALONG WITH MY MODEL 1, WAS IN A MINORITY OF ONE. THEN I CAME ACROSS IRA GOLDKLANG'S TRS-80 WEB SITE -

WWW.TRS-80.COM

I DON'T THINK I'VE SEEN SO MUCH 'TANDY' INFORMATION IN ONE PLACE (EVEN DURING THE 1980'S).
IF YOU'VE THE SLIGHTEST INTEREST IN TRS-80'S, TRY AND VISIT THE SITE, IT'S A REAL EYE OPENER!
I THEN WONDERED IF THERE WOULD BE ANY INTEREST IN AN 8-BIT NEWS-SHEET (OR POINT OF CONTACT) RELATING TO ANY OF THE TANDY 8-BIT COMPUTERS WITHIN (BUT NOT EXCLUSIVELY) THE UK.

THE EASIEST WAY TO FIND OUT, I THOUGHT, IS TO PRODUCE A SMALL NEWS-SHEET, TRY AND PASS THE WORD AROUND AND SEE WHAT, (IF ANYTHING), HAPPENS.

THIS FIRST, 4 PAGE 'STARTER' NEWS-SHEET IS AVAILABLE FOR DOWNLOADING AT

TRS-80.ORG.UK

IT'S IN PDF FORMAT. IF YOU ENCOUNTER PROBLEMS, CONTACT ME AT -

DUSTYM@BEEB.NET

AND I WILL DO MY BEST TO SEND YOU A COPY ASAP, EITHER BY EMAIL OR GOOD-OLD SNAIL-MAIL.

AFTER THIS, THE FIRST ISSUE, ANY FURTHER PUBLICATIONS WILL DEPEND ON THE RESPONSE I RECEIVE.
ITEMS FOR PUBLICATION, CAN BE ARTICLES, FRIENDS-REUNITED QUESTS ETC. EMAIL TO ME ANYTHING THAT YOU MIGHT THINK IS OF INTEREST (SO LONG AS IT LEGAL!) AND I'LL PUBLISH IT.
WITH ANY LUCK, A SECOND EDITION COULD APPEAR.

--== oØo ==--

IT WAS MY INTENTION TO PRODUCE THE NEWS-SHEET USING MY MODEL1 AND 'PENCIL' BUT I CAN'T FIND MY CASSETTE COPY OF PENCIL, AND MY OLD TELETYPE PRINTER WAS VICTIM OF A CLEAR-OUT, MANY MOONS AGO.
SO, MS PUBLISHER AND MY FAVOURITE TRUETYPE FONT 'TELEPRINTER' CAME TO THE RESCUE.



BYE FOR NOW

DUSTY

AT THE
READY>
PROMPT



IT REALLY IS QUITE LIKE OLD TIMES, SAT IN FRONT OF A MODEL1. THE 'FUN' RETURNS TO COMPUTING THE MINUTE MY MODEL 1'S SWITCHED ON.
THE SAME EFFECT CAN BE ACHIEVED WHEN RUNNING MATHEW REED'S TRS-80 EMULATOR ON YOUR PC.
I PURCHASED THE DOS VERSION SOME TIME AGO AND AM NOW JUST TRYING OUT THE WINDOWS VERSION. THIS VERSION EMULATES THE MODELS 1,3 & 4, WHEREAS THE DOS VERSION HAS JUST MODELS 1 & 3. THE SHAREWARE VERSIONS ARE AVAILABLE FROM -

WWW.ARROWWEB.COM/MKR

IT'S QUITE A REMARKABLE PIECE OF SOFTWARE. THE WINDOWS VERSION EVEN ALLOWS A MODEL 1 TO USE A VIRTUAL CASSETTE FOR SAVING AND LOADING PROGRAMS. IT WORKS FIRST TIME, EVERY TIME! (I CAN'T REMEMBER THAT HAPPENING OFTEN!)

--== o0o ==--

IN THE TRUE SPIRIT OF THE MODEL1, I SET ABOUT THE OLD CHESTNUT, A PROGRAM TO CALCULATE THE DATE OF EASTER.

IT SEEMS EASTER FALLS ON THE 1ST SUNDAY FOLLOWING THE ARBITRARY 'PASCHAL FULL MOON', WHICH DOES NOT NECESSARILY COINCIDE WITH A REAL OR ASTRONOMICAL MOON.
THE 'KEY' TO A PASCHAL FULL MOON IS CALCULATED BY ADDING 1 TO THE REMAINDER OBTAINED BY DIVIDING THE YEAR BY 19 AND THEN APPLYING PRE-DETERMINED DATE INFORMATION.
THUS, FOR THE YEAR 2000, THE KEY IS 6 (THE 6TH DATA STATEMENT) OR THE 18TH APRIL.
IT IS THEN NECESSARY TO CALCULATE THE DAY OF THE WEEK FOR THE 18TH APRIL, WHICH HAPPENS TO BE A TUESDAY. THEN ADD 5 TO THE DATE BRINGING IT TO THE NEXT SUNDAY. IF THE PASCHAL FULL MOON FALLS ON A SUNDAY, THEN EASTER IS THE FOLLOWING SUNDAY.

I'VE USED A NUMBER OF IDEAS FROM LEWIS ROSENFELDER'S WONDERFUL BOOK 'BASIC, FASTER & BETTER' AND HAVE WRITTEN THE PROGRAM SO THAT IT CAN BE RUN ON A LEVEL 2, MODEL 1. (I NEVER MANAGED TO AFFORD DISKS ON MY ORIGINAL SET-UP).

BRIEFLY, THE PROGRAM RUNS AS FOLLOWS -

```
100 INPUT THE YEAR
110 GET THE PASCHAL KEY
120 READ THE DATA
130 GET THE DAY & MONTH
140 ASSIGN DAY & MONTH
150 USE ROSENFELDER'S DAY
    NUMBER CALCULATION (FROM
    01JAN1900)
160 CALCULATE THE DAY OF THE
    WEEK (0 = FRIDAY)
170-200 ADD THE 'EXTRA' DAYS
    UP TO THE NEXT SUNDAY
    AND CHECK FOR DATES >
    31ST MAR
300 PRINT THE ANSWER
```

THE PROGRAM SEEMS TO WORK OK. IT SHOULD BE CORRECT FOR THE 20TH & 21ST CENTURIES. CAN YOU IMPROVE IT? MAKE IT SMALLER OR RUN FASTER? IS THERE A BETTER WAY? GET BASHING THOSE KEYS AND LET ME HEAR FROM YOU.

--== o0o ==--

PROGRAM TO CALCULATE THE DATE OF EASTER FOR A GIVEN YEAR

```
100 INPUT "WHICH YEAR FOR EASTER ";Y
110 E=Y-INT(Y/19)*19+1
120 FOR X=1TOE:READA$:NEXTX
130 D=VAL(LEFT$(A$,2)):M=VAL(RIGHT$(A$,2))
140 IF M=3 THEN DP$="MAR" ELSE DP$ = "APR"
150 N=Y*365+INT((Y-1)/4)+(M-1)*28+VAL(MID$("000303060811131619212426",
(M-1)*2+1,2))-((M>2)AND((YANDNOT-4)=0))+D
160 A=N-INT(N/7)*7
170 IF A>1 THEN D=D+(9-A)
180 IF A=0 THEN D=D+2
190 IF A=1 THEN D=D+A
200 IF D>31 THEN D=D-31:DP$="APR"
300 PRINT"EASTER IS THE ";D;DP$
900 DATA 1404,0304,2303,1104,3103,1804,0804,2803,1604,0504
920 DATA 2503,1304,0204,2203,1004,3003,1704,0704,2703
```

--== o0o ==--

NOW HERE'S AN ARTICLE FROM A PHOTOCOPY WHICH I HAVE HAD FOR OVER 25 YEARS!
IT'S SO OLD, IT'S ON THAT 'WAXIE' TYPE PAPER WE ALL LOVED. IT IS IN SUCH POOR CONDITION DUE TO FADING IN THE LIGHT, IT WILL NOT COPY WELL. SO PLEASE, DON'T ZAP YOUR ONLY OR MASTER COPY

IT'S THREE MODS TO THE DISK VERSION OF ELECTRIC PENCIL TO SPEED THINGS UP A BIT AND ALSO TO SAVE YOU BUTCHERING YOUR KEYBOARD TO ADD ON A CONTROL KEY. I ALWAYS THOUGHT THE DESIGN OF THE 80 WAS TOO NEAT TO MESS ABOUT WITH. THAT ADDITIONAL KEY JUST DIDN'T LOOK RIGHT!

USING YOUR ELECTRIC PENCIL WITH THE RADIO SHACK LOWER CASE MODIFICATION

THE ADAPTATION OF THE ELECTRIC PENCIL TO THE RADIO SHACK COMPUTER REQUIRES A CONTROL KEY TO ACCESS THE COMMAND TABLE. SINCE THE RADIO SHACK LOWER CASE MODIFICATION DOES NOT INCLUDE A CONTROL KEY, THIS
MAKES IT IMPOSSIBLE TO USE ELECTRIC PENCIL WITH THE RADIO SHACK MODIFICATION.
FOR MANY PEOPLE THIS IS NOT A PROBLEM, AS RADIO SHACK SCRIPSIT IS BETTER THAN THE ELECTRIC PENCIL FOR MOST USERS AND COSTS LESS.
HOWEVER THERE ARE STILL SOME USES FOR PENCIL, AND SOME PEOPLE PREFER THE MORE EXPENSIVE LESS POWERFUL SYSTEM.
THESE MODIFICATIONS TO THE ELECTRIC PENCIL WILL ALLOW YOU TO USE THE BREAK KEY FOR A CONTROL KEY, AND THE CLEAR KEY FOR THE FUNCTIONS THAT THE BREAK KEY IS NORMALLY USED FOR.
THESE MODIFICATIONS WERE PROVIDED BY JEFF BROWN ON THE SOURCE, TESTED BY CLAY SCHNEIDER, AND VERIFIED BY GEORGE BLANK.
THIS ARTICLE WAS WRITTEN USING THE MODIFIED PENCIL.
THE MODIFICATIONS ALSO SKIP OVER THE UPPER/LOWER CASE QUESTIONS, THE TITLE, CHANGE THE DEFAULT PRINT MARGIN TO 5, AND START YOU IN THE "CONTROL K" SUB COMMAND MODE.

USE THE DFS FUNCTION OF SUPERZAP 3.0 TO MAKE THE FOLLOWING CHANGES TO PENCIL/CMD: (MAKE THESE CHANGES ON

A COPY OF YOUR PROGRAM TO PROTECT THE ORIGINALS)

ZAP 1 WILL:-
AT THE START-UP, SKIP THE TITLE CARD.
ALLOW FOR START-UP DEFAULT PRINT MARGIN GREATER THAN Ø.
START-UP WITH KEYBOARD IN LOWERCASE-ENTRY MODE.
START-UP IN CONTROL "K" SUBCOMMAND MODE.
*THE "Ø5" IS THE START-UP DEFAULT MARGIN AND MAY BE CHANGED TO WHATEVER VALUE YOU LIKE.

ZAP 2 WILL:-
MAKE THE BREAK KEY INTO THE CONTROL KEY, INSTEAD OF THAT OBNOXIOUS HIDDEN LITTLE BUTTON. USE CLEAR WHENEVER BREAK IS NORMALLY CALLED FOR (EXITING CONTROL "K" ETC.).
IMPORTANT
ZAP 1 MUST BE APPLIED FIRST

ZAP 3 WILL:-
SKIP OVER THE "LOWERCASE KIT INSTALLED" QUESTION, AUTOMATICALLY ANSWERING "Y" OR "N"
** "4E" FOR NO.

ZAP 1
F00743 FROM 5A 21 C9 3D 11 6E 59 CD CF 67 CD 79 65
TO 5A 3E Ø5* 32 2A 5A CD 6D 65 C3 D6 61 65

F000A4 FROM 22 B1 5C 21
TO 22 00 00 21

ZAP 2
F01047 FROM 3A 7F 38 B7 28 28 D9 CD
TO 3A 3F 38 B7 C3 CA 59 CD
F01BC FROM C0 CB 61 20 F6
TO C0 C3 BD 59 F6
F01D9 FROM 5D 1B 17
TO 5D 00 17
F010E1 FROM 5F 1E 5B
TO 5F 00 5B
F010F0 FROM FE 1B CA
TO FE 5D CA
F0069F FROM 00 54 48 45 20 45 4C 45 43 54 52 49 43 20 50 45
TO 00 E5 21 40 38 CB 56 E1 C2 FC 65 C3 06 66 28 04
CONT. FROM 4E 43 49 4C 20 20 28 43 29 20 31 39 37 39 20
TO D9 C3 93 65 3A 40 38 E6 FB 20 F5 C3 BA 65 20

ZAP 3
F016E6 FROM 6C CD CF 67 21
TO 6C 00 00 00 21
F016F5 FROM 5A CD 79 65 E6
TO 5A 3E 59** 00 E6

--== oØo ==--

MANY THANKS FOR PERSEVERING THROUGH ALL 4 PAGES! I HOPE IT WAS OF INTEREST AND HAS BROUGHT BACK A FEW HAPPY MEMORIES OF NEARLY 30 YEARS AGO! (WHERE HAVE THE YEARS FLOWN EH?)
HOW ABOUT SOMEONE (ANYONE?) COMING UP WITH A 'ONE-LINER' BASIC PROGRAM FOR THE NEXT ISSUE TO AMAZE AND OR AMUSE US? THEY NEVER CEASED TO CAPTURE MY IMAGINATION.

PLEASE EMAIL ANY CONTRIBUTION TO ME AT DUSTYM@BEEB.NET

TRS8BIT



MANY THANKS TO ALL OF YOU WHO HAVE EMAILED ME. YOUR KIND WORDS OF ENCOURAGEMENT ARE MOST WELCOME.

I AM DELIGHTED TO PRODUCE THIS, THE SECOND EDITION OF TRS8BIT AND EVEN THOUGH CONTRIBUTIONS ARE IN SHORT SUPPLY, I HOPE THERE'S SOMETHING YOU WILL FIND OF INTEREST.

I'VE TRIED TO SORT-OUT BITS FOR HARDWARE AND SOFTWARE FANS EVERYWHERE.

I'VE RESURRECTED AN ARTICLE ON HOW TO MAKE YOUR OWN 'EL-CHEAPO' SERIAL INTERFACE FOR THE MODEL 1. IT WORKS FINE WITH ELECTRIC PENCIL (AND ANY OTHER PROGRAM THAT USES BIT Ø OF PORT FFH, 255 DECIMAL FOR SERIAL OUTPUT TO A PRINTER (THE CASSETTE OUTPUT LINE). I'VE JUST PURCHASED ALL THE COMPONENTS AND STILL HAD CHANGE FROM £3! (ALTHOUGH THE 5-PIN DIN PLUGS WERE CANNIBALISED FROM AN OLD CASSETTE LEAD I FOUND AT A CAR-BOOT). I UNDERSTAND 5-PIN DINS ARE GETTING DIFFICULT TO FIND. I'VE A COUPLE OF SPARES IF YOU ARE HAVING PROBLEMS SOURCING THEM - PLEASE EMAIL ME AND I'LL SEND ONE OFF TO YOU. THEY ARE FREE BUT HELP WITH THE POSTAGE COSTS WOULD BE APPRECIATED.

I'VE HAD A REQUEST, VIA EMAIL, ASKING IF I KNOW OF ANYONE WHO IS ABLE TO OFFER A POSTAL REPAIR SERVICE (WITHIN THE UK) ON MODEL 1'S AND 3'S. IF YOU CAN HELP, PLEASE EMAIL ME WITH CONTACT DETAILS. AND I'LL PASS THEM ON.
FOR STARTERS, I'VE AN EXPANSION INTERFACE THAT SEEMS TO BE BEYOND MY MEAGRE ABILITIES!

THANKS TO ROGER STORRS, OF PC-NATGUG, I'VE MANAGED TO OBTAIN THE MISSING ISSUES OF NATGUG-NEWS WHICH ENABLES ME TO COMPLETE THE INDEX FOR THE FIRST SIX VOLUMES, (COVERING THE PERIOD 1978 TO 1985).
THE INDEX IS AVAILABLE FOR DOWNLOADING FROM THE WEB SITE.

WWW.TRS-80.ORG.UK

--== o0o ==--

THE MODEL 1 IS 30 YEARS OLD
- HAPPY BIRTHDAY -

--== o0o ==--

ALTHOUGH NO ONE HAS SUBMITTED A 'ONE-LINER' I'VE HAD A QUICK GO MYSELF. IT'S THE USUAL GAME OF 'CATCH A CHARACTER' IT EVEN MANAGES TO KEEP THE SCORE FOR YOU!
I HOPE YOU TYPE IT IN AND HAVE A GO.
IT'S CRYING-OUT FOR SOME IMPROVEMENT(S)/ENHANCEMENTS SO I'D LOVE TO SEE WHAT YOU GUYS CAN DO TO IT.

I'VE RECENTLY ACQUIRED (AT A CAR-BOOT) A LARGE BOX OF 5.25 DISKS AND CASSETTES. ALTHOUGH THE DISKS ARE SECOND-HAND, THEY SHOULD BE SUITABLE FOR USE AS 40TRACK DOUBLE SIDED SINGLE DENSITY. (AFTER RE-FORMATTING). IF YOU'RE SHORT OR IN NEED OF SOME DISKS, PLEASE LET ME KNOW AND I'LL GET SOME OFF TO YOU ASAP.
THE CASSETTES ARE BRAND NEW AND THE CASES ARE THE ONES WITH 5 SCREWS HOLDING THEM TOGETHER. ONCE AGAIN, BOTH ITEMS ARE FREE BUT HELP WITH THE POSTAGE COSTS WOULD BE APPRECIATED.

PLEASE EMAIL ME FOR FURTHER DETAILS.

A REGULAR CONTRIBUTOR TO TRS-TIMES WAS DR MICHAEL ECKER WHO WAS, UNTIL ITS DEMISE IN NOVEMBER 2006, THE EDITOR IN CHIEF OF THE MAGAZINE 'REC' (RECREATIONAL & EDUCATIONAL COMPUTING).
HE IS OFFERING VARIOUS DVD DEALS OF PAST ISSUES (INCLUDING PROGRAMS) ON HIS WEB SITE -

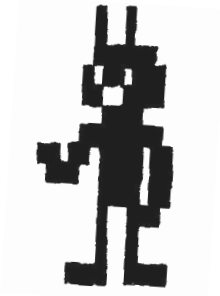
HTTP://MEMBERS.AOL.COM/DRMWECKER/REC.HTML

IT'S WELL WORTH A LOOK. THERE ARE SOME REALLY EXCITING ARTICLES REFERRED TO. (YET ANOTHER ITEM TO GO ON MY WISH-LIST)!

I'VE HAD AN EMAIL FROM MATHEW REED REPLYING TO VARIOUS QUESTIONS I HAD RAISED WITH HIM REGARDING THE DOS AND WINDOWS EMULATORS. HE INTIMATED THAT HE MAY GO AHEAD AND INCLUDE WITHIN HIS MODEL 1 AND 3 EMULATOR, FACILITIES FOR USING THE EXATRON STRINGY-FLOPPY. IF THAT IS TO BE THE CASE, I ASKED HIM TO BEAR IN MIND THAT WE IN THE UK WERE MORE FAMILIAR WITH THE ACULAB FLOPPY-TAPE. SO IF THE SYSTEM USED RELIES ON A ROM IMAGE BEING PRESENT IN THE CORRECT DIRECTORY ON THE PC, ALL I (WE) NEED TO DO IS GET OUR HANDS ON AN ACULAB, TO DOWN LOAD THE ROM (AS WITH THE LEVEL 2 ROM) AND BINGO! @ZEN AND @PENCIL RIDE AGAIN.



BYE FOR NOW
DUSTY.



AT THE
READY>
PROMPT



AS YOU CAN SEE BELOW, I'VE COME UP WITH A ONE-LINER, JUST TO START THE BALL ROLLING! THE OFT LAMENTED PHRASE I WAS SUBJECTED TO AT SCHOOL, "COULD DO BETTER IF TRIED HARDER" SPRINGS TO MIND!
'GRAND THEFT AUTO' IT AINT, BUT A REASONABLE FIRST ATTEMPT I HOPE?
IT'S QUITE EASY TO RUN, JUST KEEP PRESSING THE NUMBER KEYS TO MOVE THE CATCHER, ON THE TOP ROW, TO THE RIGHT.
IT'S 1 POINT FOR AN "!" MARK AND 10 FOR A "*" EVERY TIME THEY GO THROUGH THE GAP.

THE FOLLOWING WAS AMONG MY BITS OF NOTES I'VE ACCUMULATED OVER TIME. (MY HANDWRITING WAS ALWAYS JUST AS BAD IT SEEMS!)

ASSEMBLER (OR EVEN BASIC) SOUND ROUTINES NORMALLY SEND THE VALUES 1 AND 0 ALTERNATELY OUT THE CASSETTE PORT (255) WITH A VIBRATION COUNT TO CREATE SOME SOUND.
IF THE PROGRAMME DISPLAY RUNNING THE ROUTINE USES DOUBLE SIZED CHARACTERS, CHR$(23), AS SOON AS YOU SEND THE 1 OR 0 OUT OF THAT PORT, IF FLIPS THE DOUBLE SIZE CHARACTERS TO SINGLE SIZE AND WRECKS THE SCREEN IMAGE. HOWEVER, IF YOU SEND 8 AND 9 INSTEAD OF 0 AND 1 THE DOUBLE SIZE DISPLAY REMAINS INTACT.

--== o0o ==--

```
1 CLS:A$=CHR$(143)+CHR$(143)+"-> <-"+CHR$(143)+CHR$(143):FORC=0T
O1STEP0:B$=INKEY$:E=VAL(B$)*5:FORD=1TO7:S=S+PEEK(15364+E)-32:PRI
NT@E,A$;:PRINT@54,"SCORE";S;:X=RND(124)+895:PRINT@X,"  !!!  ":PR
INT@X,"! * !":NEXTD:NEXTC
```

ULTRA-SIMPLE SERIAL PRINTER INTERFACE.

THE ORIGINAL AND BRILLIANT IDEA FOR A SIMPLE SERIAL PRINTER INTERFACE WAS DEVISED BY LEON HELLER, CO-FOUNDER, ALONG WITH BRIAN PAIN, OF NATUG. MY OLD PAL, DAVID GODDARD, AND I, DECIDED TO SLIGHTLY AMEND AND THEN BUILD THE INTERFACE AS DETAILED ON THE PRINTED CIRCUIT OVER-LEAF AND I RAN VARIOUS SERIAL PRINTERS FROM IT ALL THE WHILE I HAD MY ORIGINAL MODEL 1.

IT IS A VERY SIMPLE, LOW-COST CIRCUIT. IT DERIVES ITS POWER FROM THE MODEL 1 POWER SUPPLY VIA A BRIDGE RECTIFIER AND CAPACITOR COMBINATION DELIVERING ABOUT +9 & -9 VOLTS. THE ACTUAL INTERFACE CONSISTS OF A 741 OPERATIONAL AMPLIFIER CONNECTED AS A COMPARATOR WITH A 100K POT. PROVIDING THE THRESHOLD ADJUSTMENT. IT IS ADVISABLE TO USE A 10-TURN CERMET TRIMPOT FOR THE FUNCTION AS, IF I REMEMBER CORRECTLY, THE ADJUSTMENT CAN BE A LITTLE TRICKY.

TO SET THE CIRCUIT UP MERELY CONNECT IN LINE WITH THE POWER SUPPLY AND KEYBOARD AND PLUG IN THE CASSETTE OUTPUT LEAD FROM THE KEYBOARD UNIT. DON'T CONNECT THE PRINTER AT THIS STAGE. USE THE APPROPRIATE COMMAND TO ACTIVATE THE PRINTER AND MONITOR THE OUTPUT FROM THE INTERFACE WITH A VOLTMETER OR A 'SCOPE. ADJUST THE 100K TRIMPOT UNTIL YOU GET THE OUTPUT SWINGING BETWEEN + AND - 9 VOLTS, ENSURING THAT WHEN THE OUTPUT IS BETWEEN CHARACTERS (A MARK CONDITION) AND IDLING, -9 VOLTS IS ON THE OUTPUT FROM THE INTERFACE. CONNECT UP THE PRINTER AND YOU SHOULD BE AWAY

CONSTRUCTION IS NON-CRITICAL AND ANY CONVENIENT CONSTRUCTION TECHNIQUE CAN BE EMPLOYED. OURS WAS BUILT UPON A SMALL PRINTED CIRCUIT BOARD AND MOUNTED IN A SMALL PLASTIC SWEET BOX! I CAN'T GUARANTEE THAT THE RS232 SPECIFICATION IS MET 100% BUT IT WORKED WELL WITH ELECTRIC PENCIL AND BASIC. (ALTHOUGH BASIC NEEDED A DRIVER IN PROTECTED MEMORY).

THE OLD TELETYPE I HAD RAN AT 110 BAUD AND MADE A FABULOUS NOISE, WITH OUTPUT VERY SIMILAR TO THIS PUBLICATION! I LATER UPGRADED TO A 'TERMIPRINTER' WHICH HAD LOWER CASE AND NECESSITATED THE MODEL 1 LOWERCASE MOD (DONE, IF I REMEMBER CORRECTLY, BY OZ HOUSE).

I REMEMBER THAT LEON DROVE THE RS232C INTERFACE FROM HIS MODEL 2 TO HIS MODEL 1 USING THIS INTERFACE QUITE HAPPILY.

I WAS AMAZED THAT ALL OF THE PARTS WERE STILL SO EASILY AVAILABLE (EXCEPT THE 5-PIN DINS AND AS I SAID EARLIER, I'VE A COUPLE OF SPARES IF YOU ARE HAVING PROBLEMS LOCATING THEM).
HAVING JUST PURCHASED ALL THE REQUIRED BITS, I SHALL HAVE TO GIVE DAVE A NUDGE AND SEE IF HE'S STILL GOT A WARM SOLDERING IRON! (ONLY KIDDING DAVE!)

I'LL TRY AND FIND A COPY OF THE BASIC PROGRAM NEEDED TO ENABLE USE OF THE INTERFACE FROM WITHIN LEVEL 2 BASIC.



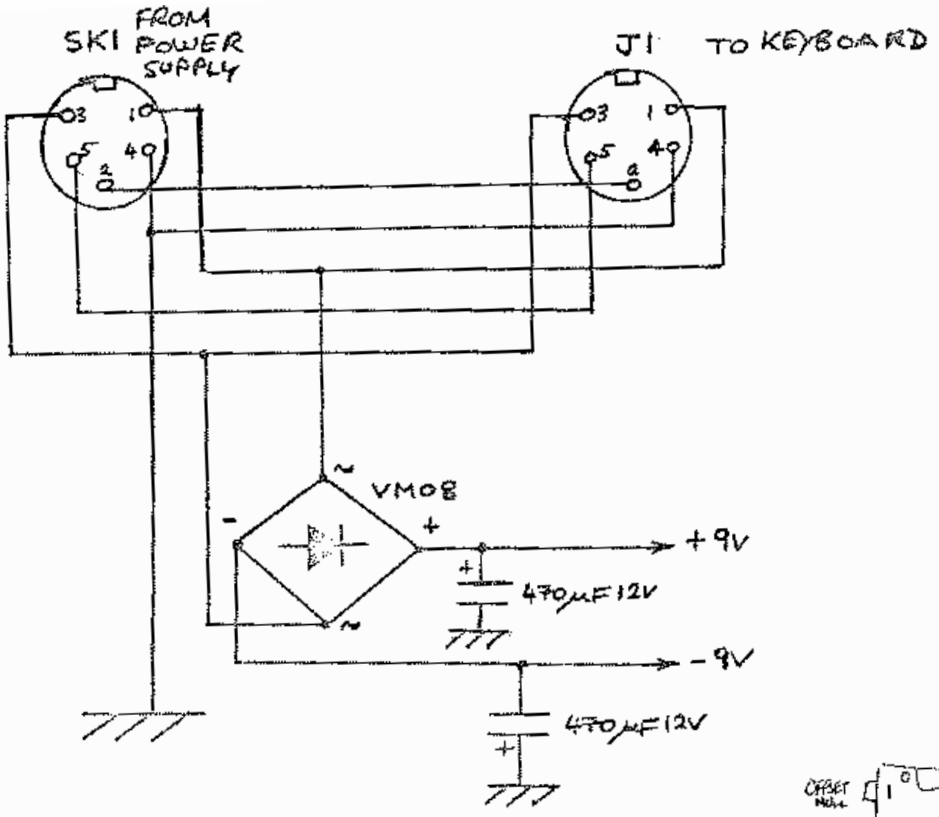
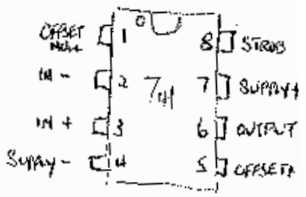


FIG. 1 - Power supply



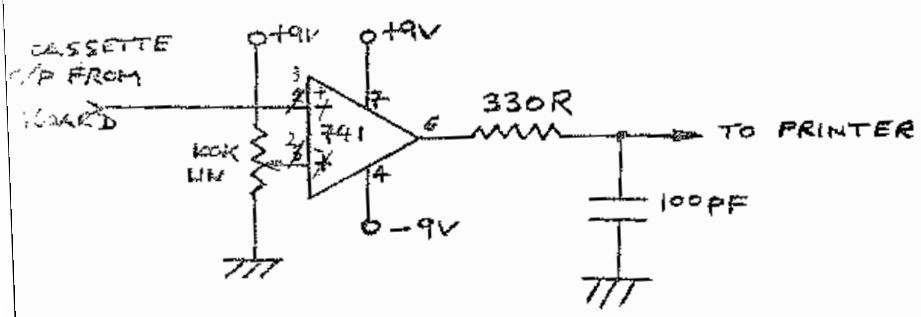


FIG. 2 - Interface circuit

---== o0o ==---

I HOPE YOU'VE ENJOYED ISSUE NUMBER 2, ENOUGH TO 'HAVE A GO' YOURSELF, EITHER WITH A ONE-LINER OF YOUR OWN, OR BUILDING THE SERIAL INTERFACE. PLEASE LET ME KNOW HOW YOU GET ON, I'D LOVE TO HEAR FROM YOU WITH ANY COMMENTS, ARTICLES OR SUGGESTIONS. WITH ANY LUCK, ISSUE 3 SHOULD 'HIT THE STREETS' DURING SEPTEMBER '07 IN THE MEANTIME, YOU CAN, AS USUAL, CONTACT ME AT

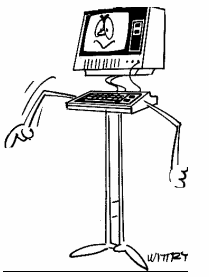
DUSTYM@BEEB.NET

TRS8BIT



WELCOME TO THE THIRD EDITION OF TRS8BIT. ONCE AGAIN I'VE TRIED TO FIND BITS (NEW AND OLD) OF INTEREST FOR EVERYONE.

I HAVEN'T RECEIVED ANY CONTRIBUTIONS FOR THIS ISSUE BUT I HOPE I'VE MANAGED TO FIND QUITE A FEW INTERESTING LITTLE BITS NONETHELESS. I USUALLY TEST ALL PROGRAMS ON MATHEW REED'S (DOS) MODEL 1 EMULATOR SO DON'T BE SHY IN TRYING THEM FOR YOURSELF. AFTER ALL, THERE'S A SHAREWARE VERSION AVAILABLE FROM HIS WEB-SITE.



ALL AND ANY CONTRIBUTIONS, FOR THE NEXT ISSUE (WHICH WILL BE THE XMAS ONE), WILL BE GRATEFULLY RECEIVED!

THERE'S A COUPLE OF ONE-LINERS JUST FOR THE FUN OF IT. VARIOUS HINTS AND TIPS FOR BASIC PROGRAMMERS AND A SUPER MACHINE CODE ROUTINE THAT LAURIE (ZEN) SHIELDS DEVISED AS PART OF ONE OF HIS TRAINING SESSIONS. IT'S A MACHINE CODE SOURCE OF A PROGRAM TO DUMP THE SCREEN TO A PRINTER. THE OBJECT CAN THEN BE CALLED VIA A USER(N) FROM BASIC. IT'S SUITABLE FOR THE MODEL 1 (WITH OR WITHOUT LOWER CASE) AND THE MODEL 3. THERE'S ALSO A (THE?) SOLUTION TO THE GAME OF ANDROID NIM. (TAKEN FROM AN ORIGINAL ARTICLE BY DR A.R. MANSON).

EVERYWHERE I SEEM TO LOOK AT THE MOMENT, THERE ARE EMAILS AND NOTES ABOUT THE STOPPING OF DOWNLOADS FROM IRA'S WEB-SITE. FROM WHAT I GATHER, HE'S UPSET AT SOMEONE SELLING A CD CONTAINING SOME OF THE DOWNLOADS AVAILABLE FROM HIS SITE.
I DON'T THINK STOPPING DOWNLOADS WILL HELP THE SITUATION IN ANY WAY. IT WILL, IN MY OPINION, MAKE THE CD'S FOR SALE MORE VALUABLE. AFTER ALL, THE INFORMATION THEY CONTAIN IS NO LONGER AVAILABLE FOR FREE!!
IRA IS DOING A REMARKABLE JOB IN RUNNING TRS-80.COM. IT MUST TAKE AN INORDINATE AMOUNT OF TIME AND MONEY TO RUN THE SITE, AND PERHAPS HE FELT THAT THIS WAS 'THE LAST STRAW'.
I DO FEEL SORRY ABOUT IT THOUGH, HAVING THE DOWN LOADS AVAILABLE WAS SUCH A UNIQUE AND VALUABLE FACILITY FOR ALL THE TRS-80 COMMUNITY.

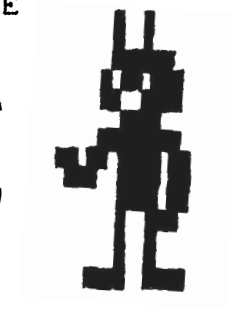
I'VE HAD AN EMAIL FROM ALAN POUND, THE FOUNDER OF THE ACULAB AND THE FLOPPY TAPE. I CONTACTED HIM TO SEE IF HE HAD ANY TECHNICAL INFORMATION ABOUT THE FT BUT WAS SORRY TO HEAR THAT HE HAS NO INFORMATION REGARDING THE FLOPPY TAPE WHATSOEVER. IT'S ALL DISAPPEARED INTO THE MISTS OF TIME.
I PROMISED THAT, IF ANYTHING OF INTEREST COMES TO LIGHT, I WOULD LET HIM KNOW, SO CAN YOU HELP? SO FAR I'VE MANAGED TO FIND A COPY OF THE OPERATOR MANUAL, A COPY OF THE PROGRAM DIRTOP/BAS AND A REVIEW OF XBAS, BUT THAT'S ABOUT IT FOR THE MOMENT.
HAVE YOU ANYTHING AT HOME

TUCKED AWAY IN THE LOFT, SUCH AS ADVERTS, MANUALS FOR BOTH SOFTWARE OR HARDWARE ETC? PLEASE CONTACT ME IF YOU HAVE SIGHT OF ANY.

STOP PRESS! (SORT OF). I'VE JUST PURCHASED A QUICK-PRINTER 2 ON EBAY COMPLETE WITH MANUAL. I'LL SEND A PDF COPY OF THE MANUAL TO IRA AT TRS-80.COM, AS I KNOW HE HASN'T ONE. AS DOWN LOADS ARE STILL PROHIBITED I'LL SEE IF THERE IS ENOUGH SPACE ON TRS-80.ORG.UK AND SET IT UP FOR ANYONE WHO WANTS A COPY.



BYE FOR NOW DUSTY.



--== o0o ==--

AT THE **READY>** PROMPT



IF YOU NEED TO DETERMINE A PRINT@ POSITION AND ONLY KNOW THE X AND Y CO-ORDINATES, THIS IS THE SIMPLE FORMULAE REQUIRED -

```
P=INT(Y/3)*64+INT(X/2)
```

AND TO REVERSE THE PROCEDURE ONLY KNOWING THE PRINT@ NUM-BER -

```
Y=3*INT(P/64)
X=2*(P-64*Y/3)
```

FOR THOSE VISICALC AND ELECTRIC PENCIL USERS DON'T FORGET THAT SAVING A VC FILE USING CMD"P" THEN "FILE", THE VC FILE IS SAVED WITH THE SUFFIX /PRF. JUST RENAMING IT TO /PCL WILL ENABLE IT TO LOAD INTO ELECTRIC PENCIL.

PRINTING BLOCKS OF GRAPHIC CHARACTERS TO THE SCREEN CAN BE QUITE SLOW, (EVEN WHEN POKEING) SO THE FOLLOWING MIGHT BE OF INTEREST. NOT AS QUICK AS MACHINE CODE BUT NEVER THE LESS, QUITE IMPRESSIVE FOR SHORT DISPLAY RE-BUILDING ETC. HERE'S AN EXAMPLE -
PRINT@N,STRING$(N,CHR$(191))
HUMMM .. MUST BE WORTHY OF INCLUSION IN A ONE-LINER DON'T YOU THINK?

HERE'S A TIP I PICKED UP FROM LAURIE SHIELDS, FOR ONE OF THE SMALLEST MACHINE CODE UTILITIES AVAILABLE ON ANY DISK SYSTEM. IT'S FOR USERS OF THE DISK VERSION OF ELECTRIC PENCIL. THE PROGRAM HAS NO CODING AND IS SIMPLY AN EXECUTION ADDRESS OF 5CC5H (PENCIL'S RE-ENTRY POINT). SO IF EVER YOU'RE IN THE MIDDLE OF WRITING A PENCIL FILE AND THE SYSTEM LOCKS OR REBOOTS, SIMPLY KEY-IN PENCRASH/CMD (OR WHATEVER NAME YOU'VE GIVEN THE PROGRAM) AND BINGO!

THIS IS A VERY ADDICTIVE ONE LINER. IT'S FROM AN ORIGINAL IDEA BY JAMES PETIVAN FROM NEW ORLEANS OVER 25 YEARS AGO! I THINK IT IS A VERY NEAT BIT OF CODING. IF YOU WANT TO MAKE IT EASIER, WIDEN THE GAP BETWEEN THE EXCLAMATION MARKS AND THE FULL STOP.

```
10 CLS:J=27:P=15391:FORM=0TO1STEP0:T=RND(99):FORL=RND(T)TOTSTEPR
ND(9)/9:J=50-ABS(ABS(J+SIN(L))-50):A=PEEK(14400)/32:P=P-((AAND1)
-(A/2AND1))*(PEEK(14464)+1):PRINTTAB(J)"!!!    .    !!!":IFPEEK(
P)=33PRINT"SC0RE:"SELSEPOKEP,191:S=S+1:NEXTL,M
```

YOU'RE BACK IN BUSINESS. ONE OR TWO CHARACTERS ON THE LINE WITH THE CURRENT BLINKING CURSOR MAY BE ALTERED, BUT THE REST OF THE TEXT WILL BE THERE. THE SOURCE CODE IN EDTASM FORMAT IS

```
    END 5CC5H
```

OR, IF YOU PREFER (AND WHO WOULDN'T) ZEN FORMAT

```
    EXEC    5CC5H
    END
```

YOU CAN EVEN CREATE A CMD PROGRAM USING BASIC BY TYPING THE FOLLOWING

```
10 OPEN '0',1,"PENCRASH/CMD"
20PRINT#1,CHR$(2);CHR$(2);CH
R$(&HC5);CHR$(&HC5)
30 CLOSE:END
```

FOR LEVEL 2 CASSETTE USERS, IF PENCIL LOCKS-UP ON YOU, ALL YOU NEED TO DO AFTER RESETTING BACK TO THE READY> PROMPT IS

```
SYSTEM <ENTER>
/17318 <ENTER>
```

WITH ONLY 4K, OR EVEN 16K AT YOUR DISPOSAL, EVERY BYTE CAN COUNT! IF YOUR PROGRAM LINE ENDS, FOR EXAMPLE, PRINT "BYE", DON'T FORGET, YOU CAN DROP THE LAST ". THE PROGRAM STILL WORKS OK AND IT SAVES YOU ONE WHOLE BYTE!

--== o0o ==--

ANDROID NIM

FROM AN ARTICLE BY
DR. A.R. MANSON

I WAS CAPTIVATED BY THE GAME ''ANDROID NIM'', BY LEO CHRISTOPHERSON, AND SO WERE MY KIDS AND NOW, EVEN MY GRANDKIDS TOO! WERE YOU AWARE THAT IT IS A SPECIAL CASE OF THE ANCIENT CHINESE GAME NIM?
THE MOST GENERAL GAME OF NIM MAY HAVE ANY NUMBER OF ROWS AND ANY NUMBER OF ELEMENTS (ANDROIDS?) PER ROW. DETERMINATION OF THE WINNING PLAYER IN THE GAME DEPENDS ONLY ON TWO ITEMS, AFTER ONE LEARNS THE WINNING STRATEGY. THESE TWO ITEMS ARE: (1) WHICH PLAYER MOVES FIRST, AND (2) WHETHER A PLAYER ''WIN'' IS DEFINED TO OCCUR WHEN THE PLAYER REMOVES THE LAST ELEMENT OR WHEN THE PLAYER FORCES THE OPPOSING PLAYER TO REMOVE THE LAST ELEMENT. THE GENERAL RULES OF "ANDROID NIM" ARE SIMPLE, AS I'M SURE YOU KNOW, A PLAYER MAY REMOVE ANY NUMBER OF ELEMENTS FROM ANY SINGLE ROW IN HIS TURN, SO LONG AS HE REMOVES AT LEAST ONE AND AT MOST THE NUMBER REMAINING ON THE ROW SELECTED. A WIN IS DEFINED TO OCCUR WHEN THE PLAYER IS ABLE, (OR FORCE HIS OPPONENT) TO REMOVE THE



HERE'S ANOTHER ONE LINER. I'VE TRIED TO CREATE A TWIRLING DOUBLE HELIX EFFECT. SEE IF YOU CAN YOU IMPROVE ON IT!

```
1 A$=CHR$(149):FORB=9TO26:FORC=1TO50:A=A+.25:PRINTTAB(28+SIN(A)*
B);A$:PRINTTAB(28+COS(A)*B);A$:NEXTC,B
```

LAST ELEMENT IN HIS TURN. THEN THE GENERAL SOLUTION TO THE R (ROWS) PROBLEM WITH C (COLUMN) ANDROIDS IS AS FOLLOWS. EXPRESS THE ORIGINAL NUMBER OF ANDROIDS ON EACH ROW AS A BINARY NUMBER (Ø'S AND 1'S).
IF THE NUMBER OF L'S IN EACH COLUMN OF THE ROWS OF BINARY NUMBERS IS EVEN, THEN THE PLAYER WHO GOES FIRST WILL LOSE, PROVIDED HIS OPPOSING PLAYER USES THE FOLLOWING SIMPLE STRATEGY: WHATEVER NUMBER OF ANDROIDS ARE REMOVED BY THE FIRST PLAYER THE SECOND PLAYER SHOULD RE-EXPRESS THE REMAINING ANDROIDS AS ROWS OF BINARY NUMBER, SCAN THE COLUMNS OF THESE BINARY NUMBERS, AND REMOVE SUFFICIENT ANDROIDS FROM THAT ROW WHICH WILL MAKE THE BINARY REPRESENTATIONS OF THE ROWS HAVE AN EVEN NUMBER OF 1'S IN ALL COLUMNS. CONTINUATION OF THIS STRATEGY WILL ALWAYS RESULT IN A WIN FOR THE PLAYER WHO GOES SECOND.

IF THE NUMBER OF 1'S IN AT LEAST ONE COLUMN IS ODD, THEN THE PLAYER WHO TAKES THE FIRST TURN CAN ALWAYS WIN. THIS IS ACCOMPLISHED BY REMOVING THE APPROPRIATE NUMBER OF ANDROIDS FROM THAT ROW WHICH WILL MAKE THE BINARY REPRESENTATIONS OF THE COLUMNS HAVE AN EVEN NUMBER OF L'S. AFTER THE OPPOSING PLAYER REMOVES ELEMENTS. THE AFOREMENTIONED STRATEGY IS REPEATED UNTIL THE WIN IS OBTAINED.
THIS PROCESS IS EASY TO DEMONSTRATE. FOR EXAMPLE ANDROID NIM HAS R = 3 ROWS WITH C1 = 7, C2 = 5, AND C3= 3. IN TERMS OF BINARY NUMBERS THESE C1 MAY BE EXPRESSED AS:

1 1 1 = 7
1 Ø 1 = 5
Ø 1 1 = 3

NOTE THAT THE THIRD COLUMN OF THE BINARY REPRESENTATIONS CONTAINS AN ODD NUMBER OF 1'S (I.E. 3). THUS THE FIRST PLAYER MERELY REMOVES ONE ELEMENT FROM ANY OF THE THREE ROWS TO GIVE AN EVEN NUMBER OF 1'S IN EACH COLUMN. PLAYER NUMBER 1, THEREAFTER CAN FORCE A SIMILAR SITUATION REGARDLESS OF PLAYER 2'S CHOICE AND THEREFORE CAN ALWAYS WIN. TRY IT!

I HOPE THAT THE GAME SOLUTION GIVEN HERE WILL NOT DETRACT FROM YOUR ENJOYMENT OF NIM; HOWEVER, THERE IS NO WAY TO MAKE NIM A FAIR GAME BY CHANGING ROWS OR COLUMNS, EXCEPT BY ALTERNATING WHICH PLAYER HAS THE OPTION OF THE FIRST MOVE.

--== oØo ==--

THE SOURCE CODE OVERLEAF IS A SCREEN DUMP PROGRAM WHICH AN OLD COLLEAGUE, LAURIE SHIELDS, DEVISED AS PART OF A MACHINE CODE PROGRAMMING TUTORIAL HE GAVE MANY MOONS AGO. IT'S WRITTEN FOR USE WITH THE 'ZEN' EDITOR ASSEMBLER (WHO WOULD USE ANY OTHER?). THE OBJECT PROGRAM IS DESIGNED TO RUN WITHIN BASIC AS A USER CALL. I REMEMBER ZEN WAS REALLY GREAT AT TURNING MEMORY INTO DATA STATEMENTS. PERHAPS THIS IS A THEME I COULD EXPAND ON IN FUTURE ISSUES?

```
1 CLS:C=RND(9)+3:FORA=0TO6.28STEP.03:R=6*SIN(C*A):X=R*7*COS(A)+6
4.5:Y=R*3*SIN(A)+24.5:SET(X,Y):NEXT:RUN
```

```
                                    ;SCRNPRNT
                                    ;FROM LAURIE (ZEN) SHIELDS
                                    ;
PRINTER:    EQU     3BH             ;ROM CALL ROUTINE
            ORG     8000H           ;CAN BE ANYWHERE
            LOAD    8000H
START:      XOR     A               ;A REGISTER = ZERO

            LD      HL,3C80H        ;START OF VIDEO

LOOP:       LD      A,L
            AND     3FH             ;END OF LINE?
            LD      A,0DH           ;CARRIAGE RETURN CODE
            CALL    Z,PRINTER       ;SEND C/R IF NEEDED

            LD      A(HL)           ;GET NEXT CHARACTER
            BIT     6,H             ;IS H = 40H
            JR      NZ,DONE         ;YES SO FINISH
            INC     HL              ;ADD 1 TO POINTER

            CP      192             ;CANT PRINT IF > 191
            JR      C,OK1
            LD      A,"."           ;CHANGE GRAPHIC TO A DOT

OK1:        CP      20H             ;TANDY LOWER CASE MOD?
            JR      NC,OK2          ;OR MODEL 3 FUNNIES
            ADD     A,64            ;FOR MODEL 3 LD A,"."

OK2:        PUSH    AF              ;SAVE CHARACTER
            LD      A,(3840H)       ;CONTROL KEYS
            BIT     2,A             ;(BREAK)
            JR      NZ,BREAK

            POP     AF              ;RESTORE CHARACTER
            CALL    PRINTER
            JR      LOOP            ;GO BACK FOR NEXT ONE

BREAK:      LD      A,(3840H)
            BIT     2,A             ;WAIT UNTIL BREAK
            JR      NZ,BREAK        ;KEY RELEASED

            LD      A,0DH           ;CARRIAGE RETURN
            CALL    PRINTER         ;CLEAR PRINTER BUFFER
            POP     AF              ;RESTORE THE STACK
DONE:       RET
            END
```

---== o0o ==---

THE ERROR CORRECTIONS OVER LEAF I'VE ONLY RECENTLY FOUND IN A TANDY PUBLICATION. I'VE HAD MY LEVEL 2 MANUAL OVER 25 YEARS AND WASN'T AWARE OF THEM!!

YET ANOTHER ONE LINER, JUST TO BRIGHTEN THINGS UP A BIT. SORRY, BUT I'VE NO ACCREDITATION FOR THIS ONE.

```
10 CLEAR195:R=RND(159)+32:A$=STRING$(64,R):FORX=0TO1023STEP64:PR
INT@X,A$;:NEXTX:S=RND(159)+32:C$=STRING$(64,S):FORX=960T00STEP-6
4:PRINT@X,C$;:NEXTX:RUN10
```

Derived Function List Correction

It has come to our attention that our list of derived functions in the Level II Manual (both editions) and in our Double Precision Subroutine Package (26-1704) contain some misprints. All of these lists were taken from the same source and have the same errors.

Please make note of the following corrections and additions to these functions:

The list of Derived Functions contains typographical errors in the following functions:

```
HYPERBOLIC COSINE:    COSH(X) = (EXP(X) +
                      EXP(-X)) / 2
HYPERBOLIC TANGENT:   TANH(X) = -EXP(-X)
                      / (EXP(X) + EXP(-X))
                      * 2 + 1
HYPERBOLIC COTANGENT: COTH(X) = EXP(-X)
                      / (EXP(X) - EXP(-X))
                      * 2 + 1
```

In addition to the typographical errors, you should be aware of the appropriate range of values for which certain of these functions are valid:

```
INVERSE SINE                    : -1 < X < 1
INVERSE COSINE                  : -1 < X < 1
INVERSE SECANT                  : X < -1 OR
                                  X > 1
INVERSE HYPER. COSINE           : X > 1
INVERSE HYPER. TANGENT          : X * X < 1
INVERSE HYPER. SECANT           : 0 < X < 1
INVERSE HYPER. COSECANT         : X <> 0
INVERSE HYPER. COTANGENT        : X * X > 1
```

The following values are mathematically undefined but, due to round-off error and limited precision, you may get a value from these functions.

TANGENT OF 90 DEGREES AND 270 DEGREES
SECANT OF 90 DEGREES AND 270 DEGREES
COTANGENT OF 0 DEGREES AND 180 DEGREES
COSECANT OF 0 DEGREES AND 180 DEGREES

An example of this type error is:

TAN(1.5708) returns a value

TAN(90 * .01745329) returns a division by zero error indicating that no valid value exists.

1.5708 = 90 * .01745329 in single precision arithmetic, so we would have expected both of the above to have returned /0 errors.

Other values which you may want but which our functions will not provide are:

```
ARCSIN(-1)   = - PI/2
ARCSIN( 1)   =   PI/2
ARCCOS(-1)   =   PI
ARCCOS( 1)   =   0
ARCSEC(-1)   = - PI
ARCSEC( 1)   =   0
ARCCSC(-1)   = - PI/2
ARCCSC( 1)   =   PI/2
```

PI = 3.14159 26535 89793 2

Values and formulas derived or confirmed using CRC STANDARD MATH TABLES, 16th STUDENT EDITION.

WELL, THAT'S ABOUT IT FOR THIS, THE 3RD EDITION. HOPE YOU FOUND IT OF SOME INTEREST. HAVE YOU ANY IDEAS OR SUGGESTIONS FOR ARTICLES YOU'D LIKE TO SEE IN THE NEXT EDITION OF TRS8BIT?
FEEDBACK IS DEFINITELY IN SHORT SUPPLY, SO DON'T BE BASHFUL, EMAIL ME AT DUSTYM@BEEB.NET

THE NEXT ISSUE, WHICH WILL BE THE CHRISTMAS 2007 ISSUE, SHOULD HIT THE STREETS (WELL THE WEB-SITE AT LEAST) IN EARLY DECEMBER. IN THE MEANTIME, PLEASE KEEP AN EYE ON THE WEB-SITE. I'LL UPDATE IT, AS AND WHEN INTERESTING (8)BITS OF TRS-80 INFORMATION COMES TO LIGHT.

TRS8BIT



DEE AND I WISH ALL OUR READERS AND SUPPORTERS A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR. MAY ALL YOUR PROBLEMS TURN INTO INTERESTING CHALLENGES DURING 2008.

WELCOME TO THE CHRISTMAS 2007 EDITION OF TRS8BIT. (I CAN'T BELIEVE IT DECEMBER ALREADY. WHERE HAS THE YEAR FLOWN?)

I HOPE I'VE FOUND SOMETHING OF INTEREST FOR EVERYONE IN THIS, THE XMAS ISSUE.

I HAVE AN EXCELLENT ARTICLE BY ROY T BECK, WHICH FIRST APPEARED IN TRS-TIMES. ROY, IN HIS USUAL EXPERT MANNER, PRODUCED AN ARTICLE WITH THE MODEST TITLE OF 'SOME THOUGHTS ON RADIO SHACK PRINTERS'. IT IS ONE OF THE MOST REMARKABLE PIECES OF RESEARCH ON TANDY PRINTERS I'VE COME ACROSS! ROY EVEN QUOTES THE RS RIBBON CATALOGUE NUMBER!. I'VE REFERRED TO THIS ON A NUMBER OF RECENT OCCASIONS WHEN PRINTERS COME UP FOR SALE ON EBAY AND THOUGHT IT WAS WELL WORTH STAR BILLING ON A REPRINT.

THERE'S THE USUAL COUPLE OF ONE-LINERS, (JUST FOR THE HECK OF IT). I HAVE NOTICED THAT NINTENDO ARE ADVERTISING 'BRAIN TRAINING' PROGRAMS ON THEIR DS COMPUTER. SO NOT TO BE OUTDONE, I'VE INCLUDED A ONE-LINER THAT HELPS WITH MENTAL ARITHMETIC. (JUST TRY AND CONSISTENTLY ANSWER THE QUESTIONS UNDER THE COUNT OF 50! IT CAN BE DONE BUT I FOUND THAT I NEEDED TO CONCENTRATE REALLY HARD!)

THERE IS, OF COURSE, THE USUAL COUPLE OF SNIPPETS IN 'AT THE READY PROMPT' AND TO FINISH OFF, PUTTING EVERYONE IN THE XMAS MOOD THERE'S A SMALL PROGRAM WHICH PRINTS A NUMBER OF 'MERRY XMAS'S SO THAT IF YOU SQUINT AT THE PRINTOUT, IT LOOKS LIKE A COWBOY! THIS PROGRAM WAS WRITTEN USING MATHEW REEDS M1 WINDOWS EMULATOR (RUNNING UNDER WINDOWS ME) WHICH I HAVE RECENTLY PURCHASED. THE PRINTOUT WAS SENT DIRECTLY TO MY HP LASERJET PRINTER. THE EMULATOR WORKED SPLENDIDLY AND THE VIRTUAL CASSETTE IS A REAL JOY TO USE.

WHO SAYS CHRISTMAS DOESN'T COME EARLY? I'VE JUST MANAGED TO WIN, (BUY, DEE KEEPS REMINDING ME!), ON EBAY, 17 80-MICROCOMPUTINGS FROM THE EARLY 1980'S. THEY CAME FROM AN EBAY SHOP CALLED '8-BIT HEAVEN'. IT SEEMS THEY QUITE OFTEN HAVE BITS OF OLD TANDY RELATED ITEMS FOR SALE SO, I'VE PUT A LINK TO THEM ON THE WEB-SITE.

I HAVEN'T HEARD FROM ANYONE YET WITH INFORMATION ON THE ACULAB FLOPPY TAPE. I LIVE IN HOPE. CAN ANYONE HELP? ANYTHING OF INTEREST WOULD BE MOST HELPFUL.



THAT'S ABOUT IT FOR NOW -

TAKE CARE

DUSTY.

AT THE
READY>
PROMPT



THERE ARE A NUMBER OF OPTION WITHIN PRINT USING WHICH ALWAYS SEEM TO GET OVER LOOKED. FOR INSTANCE, THE EXCLAMATION MARK. YOU CAN TAKE TWO FULL CHRISTIAN NAMES AND A SURNAME AND PRINT THEM OUT AS TWO INITIALS AND A SURNAME. I KNOW IT CAN BE DONE WITH LEFT$, BUT THE EXAMPLE BELOW IS MUCH EASIER TO CODE AND JUST MORE FUN!.

```
10A$="DUSTY":B$="CLINT":C$="
MILLER"
20 PRINT USING "!";A$;B$;"
";
30 PRINT C$
```

TYPE RUN AND DC MILLER IS PRINTED. (LEAVING A$ AND B$ INTACT). NATURALLY IT ALSO WORKS WITH LPRINT SO IT'S QUITE HANDY FOR LABEL AND ENVELOPE PRINT RUNS.

FOR DISK BASIC (AND ACULAB XBAS) HERE'S A TIP. APART FROM THEIR APPLICATION IN DIRECT ACCESS FILE HANDLING LSET AND RSET CAN BE USED TO LEFT OR RIGHT JUSTIFY A STRING WITHIN ITS EXISTING LENGTH. APART FROM THEIR CONVENIENCE, THESE FUNCTION RE-USE THE EXISTING STRING STORAGE, AND THEREFORE HELP TO AVOID DELAYS CAUSED BY GARBAGE COLLECTION.

EVER WONDERED IF YOU'RE IN LEVEL II OR DISK BASIC? THE QUICKEST WAY TO FIND OUT WHICH IS TO PEEK (16396). IF THE VALUE = 201 THEN IT'S LEVEL II.



--== o0o ==--

THIS WAS ORIGINALLY INTENDED TO BE A 'NORMAL' SIZE PROGRAM (HAVING BEEN INSPIRED BY THE NINTENDO 'DS' ADVERTISEMENT FOR 'BRAIN-TRAINING') WITH ALL SORTS OF BELLS AND WHISTLES, BUT I JUST COULDN'T RESIST THE URGE TO MAKE IT A ONE-LINER!
IT LOSES A BIT OF FINESSE IN THE PROCESS, BUT AFTER ALL, THAT'S WHAT A ONE-LINER IS ALL ABOUT!
JUST TYPE THE PROGRAM IN (THE LARGE AMOUNT OF SPACE IS CREATED BY THE DOWN-ARROW, NOT THE SPACE BAR). TYPE IN YOUR ANSWER AND PRESS ENTER.
IT EVEN HAS A BUILT-IN TIMER. KEEP TRYING, YOU WILL GET QUICKER I PROMISE!

WHILE I WAS TYPING IN THESE BRIEF NOTES, I SUDDENLY REALISED THAT THE PROGRAM, (MINUS THE TIMER AND THE RANDOM NUMBER BITS ETC.) COULD EASILY BE USED AS A LEVEL 2 LINEINPUT COMMAND. CHECKING TO SEE IF THE ENTER KEY IS PRESSED TO TERMINATE INPUT IS ACHIEVED BY PEEKING(14400) AND WAITING FOR IT TO BE EQUAL TO 1. NEAT EH? I'LL PERHAPS EXPAND ON THIS IN THE NEXT ISSUE.

```
10 CLS:A=RND(985)+15:B=1000-A:PRINT"SUBTRACT ";A;"FROM 1000";:FO
RX=0TO1STEP0:C=C+1:PRINT@192,"TIME TAKEN ";C;:A$=INKEY$:B$=B$+A$
:IFPEEK(14400)=1THENPRINT"

YOU TYPED ";B$"ANSWER IS";B:INPUT"
ENTER TO GO AGAIN";Z$:RUNELSENEXTX
```



SOME THOUGHTS ON RADIO SHACK PRINTERS

BY ROY T BECK

REPRODUCED (WITH PERMISSION) FROM TRSTIMES, VOL8 NO.6

EVERY SO OFTEN YOU WILL SEE ONE OF THE OLD BATTLESHIP-GREY R/S PRINTERS AT A SWAP MEET OR SALVAGE STORE, AND YOU WONDER WHAT KIND OF A BARGAIN IT MIGHT BE. HOW DO YOU KNOW WHAT ITS CHARACTERISTICS ARE? YOU COULD WRITE TO FT. WORTH FOR A MANUAL, BUT THAT'S A BUMMER, CONSIDERING THE COST AND DELAY.

I RECENTLY GOT TO THINKING ABOUT THE WIDE VARIETY OF PRINTERS AND PRINTER-LIKE DEVICES RADIO SHACK HAS PRODUCED OVER THE YEARS. I HAVE MOST OF THE CATALOGUES IN THE RSC SERIES, RUNNING FROM RSC-2 THROUGH RSC-22A (I AM MISSING RSC-1 AND RSC-13). THIS LIBRARY IS VERY USEFUL IN SEARCHING OUT DATA ABOUT VARIOUS R/S PRODUCTS. SINCE THIS COVERS THE PERIOD FROM 1978 TO 1991, (THE TRS ERA), I DECIDED TO TABULATE ALL OF THE PRINTERS BY THEIR CHARACTERISTICS.

AS YOU READ THE TABLE, YOU WILL NOTICE A NUMBER OF 'HOLES'. I AM SORRY ABOUT THESE, BUT THE CATALOGUE LISTING OF SOME OF THE MACHINES DOESN'T ALWAYS PROVIDE ALL THE EXPECTED INFORMATION. ALSO, I HAVE TO INSERT THE USUAL LEGAL ESCAPE CLAUSE. I'M NOT RESPONSIBLE FOR ANY ERRORS YOU MAY FIND IN THE ARTICLE. I HAVE DONE THE BEST I CAN, BUT I'M ONLY HUMAN AND I DO MAKE ERRORS. (R/S IS ALSO KNOWN TO MAKE OCCASIONAL ERRORS!).

YOU MAY NOT EVEN RECOGNISE SOME OF THESE PRINTERS, BUT, YES, RADIO SHACK DID PRODUCE ALL OF THEM. BEFORE TABULATING THEM, I WOULD LIKE TO DESCRIBE THE TYPES OF PRINTERS R/S HAS PRODUCED. TAKING THEM IN ORDER, I WILL GIVE A THUMB-NAIL SKETCH OF EACH TYPE SO YOU WILL BE FAMILIAR WITH THEM. THESE ARE:

1. ELECTROSTATIC PRINTERS

THE ELECTROSTATIC TYPE IS (FORTUNATELY) AN OBSOLETE TYPE ORIGINALLY OFFERED IN THE MODEL 1 ERA. THE EARLIEST ONE WAS CALLED THE SCREEN PRINTER, AND IT WAS FOR THIS PRINTER THAT THE MODEL I HAD A 40 LINE BUS CONNECTION ON THE LEFT SIDE OF THE EXPANSION INTERFACE LABELLED 'SCREEN PRINTERS". THIS IS WHERE YOU WOULD PLUG IN THE SCREEN PRINTER OR ANY OTHER BUS DEVICE YOU MIGHT OWN. WHEN YOU PUSHED A BUTTON ON THE PRINTER, THE IMAGE ON THE SCREEN OF THE COMPUTER AT THAT MOMENT WAS PRINTED. ACTUALLY, THE SCREEN PRINTER DESIGN WAS CLOSELY INTEGRATED WITH THE OPERATION OF THE COMPUTER. WHEN THE SCREEN PRINTER WAS ACTUATED, IT SEIZED CONTROL OF THE COMPUTER, HALTED THE Z-80 CPU, AND DID A DMA TRANSFER OF THE SCREEN MAGE BY SCANNING THE VIDEO ROM (AT 2200 CPS). WHILE THIS WAS A TECHNOLOGICAL TOUR DE FORCE IT ALSO MADE THE SCREEN PRINTER ESSENTIALLY INCOMPATIBLE WITH EVERYTHING THAT WAS NOT A MODEL 1, LEVEL 1 COMPUTER. THE SCREEN PRINTER USED AN ELECTRIC ARC TO BURN A THIN FILM OF ALUMINIUM OFF A PAPER SUBSTRATE, LEAVING BLACK LETTERS ON AN ALUMINIUM FOIL SURFACE OF A SHEET OF PAPER.

THE QUICK PRINTER (1) AND QUICK PRINTER 11 USED THE SAME PRINTING TECHNIQUE, BUT MUCH SLOWER BECAUSE IT DID NOT DO DMA. THE RESULTS OF ALL THREE PRINTERS WERE UNIFORMLY POOR. THE LETTER SIZE AND SHAPE WAS FIXED IN THE DESIGN OF EACH PRINTER, COULD NOT BE ALTERED, WAS VERY SMALL, AND PRODUCED A BAD SMELL IN THE PROCESS OF PRINTING. YES, THEY WORKED, BUT NOT WELL, AND WE ARE WELL RID OF THEM.

2. PEN PLOTTERS
R/S PRODUCED SEVERAL PEN PLOTTER DEVICES. MOST OF THEM USED A SMALL BALL POINT OR FELT TIP PEN TO SCRIBE LETTERS ONTO PAPER, WHICH WAS VERY LIMITED IN SIZE. THESE WERE REALLY INTENDED TO PRODUCE GRAPHIC IMAGES, BUT I AM INCLUDING THEM BECAUSE THEY COULD OPERATE AS PRINTERS WHEN DESIRED.

3. INK JET PRINTERS
R/S ALSO PRODUCED AN INK-JET PRINTER, THE CGP220 WHICH WAS REALLY A PLOTTER.

4. THERMAL PRINTERS
WERE OFFERED FOR THE PC-3 AND PC-4 PACKET COMPUTERS. THERMAL PRINTERS ARE QUIET, BUT SUFFER FROM THE NEED FOR SPECIAL PAPER, USUALLY RATHER NARROW, AND WORSE, THE IMAGE WILL FADE WITH TIME, ESPECIALLY WHEN EXPOSED TO SUNLIGHT. THE TP10 AND TRP-100 UNITS WERE INTENDED FOR USE WITH DESK TOP COMPUTERS. THE TRP-100 WAS ESPECIALLY CLEVER; IT COULD OPERATE ON BATTERIES OR AC POWER, AND COULD USE EITHER THERMAL PAPER OR A THERMAL RIBBON TO PLACE IMAGES ON PLAIN PAPER.

5. DAISY WHEEL PRINTERS

MANY DIFFERENT DAISY WHEEL PRINTERS WERE OFFERED AT VARIOUS TIMES. ALL BUT ONE OF THESE ARE IDENTIFIED WITH MODEL NAMES BEGINNING WITH DWP MEANING DAISY WHEEL PRINTER THE ODD ONE WAS NAMED DW-11, WHICH ALSO MEANT DAISY WHEEL. I AM NOT SURE IF THERE EVER WAS A DW-I PRINTER; THE 11 ON DW-II WOULD SEEM TO IMPLY THIS, BUT I HAVEN'T FOUND ANY SOLID EVIDENCE OF SUCH. POSSIBLY A DW-1 WAS PLANNED BUT NEVER GOT INTO PRODUCTION. BILL BARDEN, ANOTHER AUTHOR, INDICATES THE DW-I EXISTED BUT I CAN'T FIND IT IN THE CATALOGUES I HAVE. SEE TABLE 5A. TABLE 5A LISTS THE WHEELS FOR SOME OF THE PRINTERS. THERE ARE INCONSISTENCIES IN THE CATALOGUES, ESPECIALLY REGARDING THE WHEELS FOR THE DW-II AND THE DW-410; IF THE SAME WHEEL FITS BOTH, AS FOR EXAMPLE, THE PICA 10, THEN WHY SHOULD THERE BE DIFFERENT WHEELS FOR CERTAIN OTHER FONTS, AS FOR EXAMPLE COURIER 10? THIS DOESN'T MAKE GOOD SENSE, BUT IT IS WHAT I FOUND IN THE CATALOGUES.

6. IMPACT DOT MATRIX
R/S PRODUCED A GREAT MULTITUDE OF IMPACT DOT MATRIX MACHINES, NUMBERING AT LEAST 38 DIFFERENT MODELS. WOW! I SUSPECT THEY ORDERED EACH BATCH OF PRINTERS TO A SPECIFICATION, AND WHEN THAT BATCH WAS SOLD OUT AND MORE WERE NEEDED, A NEW SPECIFICATION WAS DEVELOPED. SEEMS LIKE A SILLY WAY TO OPERATE, BUT THE EVIDENCE IS IN THE CATALOGUES. SEE TABLE 6.

NAMING THESE PRINTERS WAS A PROBLEM. INITIALLY, THEY PRODUCED A DOT MATRIX MODEL CALLED SIMPLY THE LINE PRINTER. AFTER THIS ONE, THERE CAME A SERIES OF EIGHT MACHINES NAMED LP-I THROUGH LP-VIII. AFTER THESE, RADIO SHACK DECIDED TO USE DESCRIPTIVE LETTER PREFIXES, AND THE LETTERS DMP FOR DOT MATRIX PRINTER WERE USED FOR MOST OF THE LATER MACHINES. THERE WAS A PC-I PRINTER WHICH WAS A DOT MATRIX PRINTER FOR USE WITH THE ORIGINAL POCKET COMPUTER. THERE IS ALSO A MACHINE IDENTIFIED AS LMP-2150, WHICH IS ALSO A DOT MATRIX PRINTER. SOME PRINTERS MAY NOT HAVE EXISTED. CERTAIN R/S CATALOGUES REFER TO RIBBONS FOR PRINTERS WHICH APPARENTLY WERE NEVER IN THE CATALOGUES. EXAMPLES ARE THE DMP-133 AND DMP-300, WHICH ARE LISTED FOR REPLACEMENT RIBBONS, BUT WHICH WERE NOT THEMSELVES LISTED IN THE RSC CATALOGUES. I AM SURE THE DMP-300 WAS A TYPO, AND SHOULD HAVE BEEN DMP-302. I CAN'T ACCOUNT FOR THE DMP-133, PROBABLY ANOTHER TYPO WHEN THE CATALOGUES WERE PREPARED.

7. LASER PRINTERS
FINALLY, RS PRODUCED AT LEAST TWO LASER PRINTERS, UNDER THE DESCRIPTORS LP-950 AND LP-1000. EVIDENTLY LP IN THIS CASE MEANT LASER PRINTER, AS OPPOSED TO LP-ROMAN NUMERAL WHICH WAS THE ORIGINAL DOT MATRIX IMPACT SERIES. SEE TABLE 7. BOTH OF THESE PRINTED SIX PAGES PER MINUTE (PPM) AND HAD A RESOLUTION OF 300 DOTS PER INCH (DPI).

THERE YOU HAVE IT, SEVEN MAJOR TYPES OF MACHINES TOTALLING ABOUT 60 MACHINES ALL TOLD.

MISCELLANEOUS COMMENTS.
I HAVE INCLUDED THE RIBBON CATALOGUE NUMBERS FOR TWO REASONS; ONE IS I ALLOW YOU TO IDENTIFY SIMILAR PRINT MECHANISMS, AND THE OTHER FOR THE SAKE OF ASSISTING YOU TO ORDER THE CORRECT RIBBON FOR YOUR MACHINE. BASED UPON THE COMMONALITY OF RIBBONS AMONG THE DOT MATRIX MACHINES, IT IS REASONABLE I ASSUME THE CORRESPONDING MACHINES ARE BY THE SAME MANUFACTURER, WITH SIMILAR INTERNAL PARTS.
I HAVE ALSO INCLUDED MY AVAILABLE INFORMATION ON PRINT WHEELS FOR THE DAISY WHEEL MACHINES.

THE CATALOGUE PRICES ARE SHOWN FOR INFORMATION ONLY. KNOWING THE RELATIVE ORIGINAL PRICES OF THE VARIOUS MACHINES WILL GIVE YOU SOME INFORMATION ABOUT THE RELATIVE QUALITY OF PRINTERS YOU MAY SEE AT SWAP MEETS. USUALLY, R/S INTRODUCED ITS PRODUCTS AT SOME PRICE, AND THEN LATER REDUCED THE PRICE. IN A FEW CASES, THEY ACTUALLY RAISED PRINTER PRICES IN LATER CATALOGUES. THE TABLES ALSO SHOW WHICH CATALOGUES LISTED EACH PRINTER. TABLE 8 SHOWS THE DATE OF EACH RSC CATALOGUE. BY NOTING WHICH CATALOGUES LISTED A PARTICULAR PRINTER, AND THEN CHECKING TABLE 8, YOU CAN DETERMINE THE INITIAL AND FINAL OFFERING DATE OF EACH PRINTER, AND THUS CAN DETERMINE THE APPROXIMATE AGE OF ANY PARTICULAR MACHINE YOU MAY SEE AT A

SWAP MEET, ETC.
WHERE POSSIBLE, I HAVE LISTED THE PRINT SPEED IN CHARACTERS PER SECOND (CPS) AND THE MAXIMUM PAPER WIDTH IN INCHES WHICH CAN BE ACCOMMODATED.
NOTE THAT TRACTOR FEED PAPER IS ABOUT ONE INCH WIDER THAN THE FINAL PRODUCT BECAUSE OF THE TEAR-OFF EDGES. SEVERAL OF THE PRINTERS USE NARROWER THAN 8.5 INCH PAPER. BEWARE OF THESE. THE STANDARD PAPER SIZES YOU MIGHT EXPECT TO SEE ARE 8.5 AND 9.5 INCH WIDE, WHERE THE EXTRA INCH REPRESENTS THE TEAR-OFF EDGE CONTAINING SPROCKET HOLES. SIMILARLY, 14.875 INCH PAPER IS AVAILABLE, WHICH HAS 14 INCHES OF USEABLE WIDTH, WHETHER THE EDGES TEAR OFF OR NOT. I HAVE ROUNDED THIS PAPER TO 15 INCHES IN THE TABLE. MOST OF THE WIDE CARRIAGE PRINTERS WILL WORK JUST FINE WITH 8.5 OR 9.5 INCH PAPER, SOME MUCH LESS, DOWN TO AS SMALL AS 4 INCHES. I HAVE LISTED THE MAXIMUM QUOTED CAPACITY OF EACH MACHINE.
THE LINE PRINTER AND THE TWO VERSIONS OF LP- 1 WERE UPPER CASE ONLY; AVOID THEM.
AS YOU ALL KNOW, I AM SURE, R/S COMPUTERS ARE DESIGNED TO SEND ONLY A CARRIAGE RETURN (CR) TO THE PRINTER AT THE END OF EACH LINE. R/S PRINTERS ARE DESIGNED TO INTERPRET THIS COMMAND AS MEANING THE PRINTER SHOULD EXECUTE BOTH A CR AND A LINE FEED (LF) BUT OTHER COMPUTERS ARE DESIGNED TO SEND A SEPARATE CR AND LF UNDER THE SAME CIRCUMSTANCES.
MOST R/S PRINTERS CAN BE SET UP TO RECOGNISE BOTH THESE CIRCUMSTANCES, BUT THERE IS NO CONSISTENT STANDARD FOR HOW YOU PROGRAM THEM TO DO THIS. SOME PRINTERS HAVE A DIP SWITCH TO ENABLE THE CR+LF RESPONSE, SOME WILL ACCEPT SOFTWARE COMMANDS, AND SOME WILL ACCEPT BOTH. I DO NOT HAVE ANY OVERALL DATA FOR THIS, I CAN ONLY COMMENT UPON THE SITUATION. SINCE YOU REALLY OUGHT TO HAVE THE MANUAL FOR WHATEVER PRINTER YOU HAVE, YOU CAN FIND THIS INFO FOR YOUR PRINTER IN YOUR MANUAL. R/S IS PRETTY GOOD ABOUT MAKING MANUALS AVAILABLE FOR OLD EQUIPMENT, SO DO ORDER ONE FROM THEM.

Table 1
Electrostatic Printer

Model	Cat. No.	Speed CPS	Dots	Max Paper	Orig. Price $	Listed in Cat RSC-	Paper No.
Quick Printer (I)	26-1153	100	7	4.75	499	2 to 3	26-1405
Quick Printer II	26-1155	64	7	2.375	219	3 to 5	26-1412
Screen Printer	26-1151	2200	7	4.75	599	1	26-1405

Table 2
Pen Plotters

Model	Cat. No.	Max Paper	Orig. Price $	Listed in Cat RSC-.
CGP-115	26-1192	4.5	250 to 200	8 to 11
FP-215	26-1193	10	995	8 to 14
Multi-Pen Plotter	26-1191	8.5	1995	6 to 9
PC-2	26-3605	2.25	240	8
Plotter/Printer	26-1190	9	1460	4 to 7

Table 3
Ink Jet Printers

Model	Cat. No.	Speed CPS	Pins	Max Paper	Orig. Price $	Listed in Cat RSC-	Ribbon No.
CGP-220	26-1268	37	7	8.5	699 to 599	10 to 16	26-1281 26-1282

Table 4
Thermal Printers

Model	Cat. No.	Speed CPS	Dots	Max Paper	Orig. Price $	Listed in Cat RSC-	Paper No.
PC-3	26-3591	24	7	2.25	120	14	26-3592
PC-4	26-3652	20	7	1.5	80	9 to 14	26-????
PC-8	26-3591	24	7	2.25	120	19	26-3592
TP-10	26-1261	30	7	4	100	12 to 15	
TRP-100	26-1275	50	7	8.5	300	12 to 16	26-1297

Table 5
Daisy Wheel Printers

Model	Cat. No.	Speed Char/Sec	Max Paper	Orig. Price $	Listed in Cat RSC-	Ribbon No.
DW-II	26-1158	43	16"	1960 to 1995	4 to 11	26-1419 26-1449
DW-IIB	26-	This printer is a slight upgrade of DW-II				
DWP-210	26-1257	18	13	799 to 599	10 to 14	26-1445 26-1458
DWP-220	26-1278	20	16	599	15 to 16	26-1299
DWP-230	26-2812	20	16	400 to 460	17 to 20A	26-1445 26-1458
DWP-410	26-1250	25	16	1495 to 1295	8 to 11	26-1419 26-1449
DWP-510	26-1270	43	16	1495	12 to 16	26-1419 26-1449
DWP-520	26-2800	43	16	995	17 to 19	26-1445 26-1458

Table 5A
Wheels for Table 5

Name of Wheel	DW-II	DWP-210	DWP-220	DWP-410
Courier 10	26-1420	26-1467	26-1230	26-1430
Prestige Elite 12	26-1421	26-1468		26-1431
Madeleine PS	26-1422			26-1432
Cubic PS	26-1425			26-1433
Title Italic 12	26-1426			26-1434
OCR B	26-1484			26-1435
Letter Gothic 12	26-1485		26-1231	26-1436
Cubic 15	26-1487			26-1438
Bold PS	26-1488			26-1439
Venezia PS		26-1469	26-1232	
Scientific A/N	26-1486			26-1486
Pica 10	26-1290			26-1290
Narrator	26-1291			26-1291
OCR-A	26-1292			26-1292
Elite 12	26-1293			26-1293

Table 6
Impact Dot Matrix Printers

Model	Cat. No.	Speed CPS	Pins	Max Paper	Orig. Price $	Listed in Cat RSC-	Ribbon No.
Line Printer	26-1150	60	7	9.8	1300	2	
LP-I	26-1152	60	7	12.1	1559	2 to 3	26-1413
LP-II	26-1154	100	7	9.5	970 to 799	3 to 5	26-1413
LP-III	26-1156	120	9	15	1960	3 to 4	26-1414
LP-IV	26-1159	50	9	9.5	999	4 to 5	26-1413
LP-V	26-1165	160	9	15	1860	5 to 7	26-1414
LP-VI	26-1166	100	7	15	1160	4 to 7	26-1418
LP-VII	26-1167	30	7	9.5	399	6 to 7	26-1424
LP-VIII	26-1168	100	9	9.5	799	6 to 7	26-1418
DMP-100	26-1253	50	7	9.5	399	8 to 10	26-1424
DMP-105	26-1276	80	9	9.5	200	12 to 17B	26-1288
DMP-106	26-2802	80	9	9.5	220	19 to 20A	26-1288
DMP-107	26-2821	100	9	10	280	22 to 22A	26-1235 26-1236
DMP-110	26-1271	50	9	10	400	11 to 14	26-1283
DMP-120	26-1255	120	9	9.5	500	10 to 14	26-1483
DMP-130	26-1280	100	28	10	350	15 to 17B	26-1235 26-1236 26-1238
DMP-130A	26-1280A	120	28	10	360	19	26-1235 26-1236 26-1238
DMP-132	26-2814	120		10	380	20A	26-1235 26-1236 26-1238
DMP-134	26-2848	160	9	10	360	22 to 22A	26-1235 26-1236
DMP-200	26-1254	120	9	9.5	799 to 699	8 to 11	26-1483
DMP-240	26-2839	192	24	10	550	22 to 22A	26-2824 26-2826
DMP-302	26-2849	270	24	10	599	22 to 22A	26-2819
DMP-400	26-1251	140	9	15	1195	8 to 9	26-1418
DMP-420	26-1267	140	9	15	999	10 to 11	26-1418
DMP-430	26-1277	180	18	15	899 to 699	12 to 17B	26-1296
DMP-440	26-2808	300	9	15	699	19 to 20A	26-2809 26-2827
DMP-442	26-2822	300	9	16	699	21A to 22A	26-2823
DMP-500	26-1252	220	9	15	1795 to 999	8 to 11	26-1482
DMP-2100	26-1256	160	24	15	1995	9 to 11	26-1442
DMP-2100P	26-1274	160	24	15	1995 to 1495	12 to 16	26-1442
DMP-2102	26-2817	270	24	16	999	21A	26-2819
DMP-2103	26-2850	270	24	16	899	22 to 22A	26-2819
DMP-2110	26-2810	240	24	15	1295	17 to 17B	26-1442
DMP-2120	26-2811	240	24	16.5	1599	19 to 21A	26-2834 26-2835 26-2836
DMP-2130	26-2845	480	28	16.5	1199	22 to 22A	26-2846
LMP-2150	26-1272	290	9	16	3995	12 to 14	26-1287
DMP-2200	26-1279	380		16	1695	15 to 17B	26-2825
PC-1	26-3505	16	7	1.75	150 to 128	6 to 7	26-3507

Table 7
Laser Printers

Model	Cat. No.	Speed PPM	DPI	Max Paper	Orig. Price $	Listed in Cat RSC-
LP-950	26-2838	6	300	8.5	1599	22 to 22A
LP-1000	26-2804	6	300	8.5	2199 to 2599	19 to 21A

Table 8
RSC Catalogs

RSC-	Date	RSC-	Date	RSC-	Date	RSC-	Date
1	1978	8	1983	15	1986	19E	1988
2	1978	9	1983	16	1986	20	1989
3	1979	10	1984	17	1987	20A	1989
4	1981	11	1984	17B	1987	21	1990
5	1981	12	1985	18	1987	21A	1990
6	1982	13	1985	18E	1987	22	1991
7	1982	14	1985	19	1988	22A	1991

--== oØo ==--

AS THIS IS THE CHRISTMAS EDITION, I THOUGHT A FESTIVE PICTURE WOULD BE IN ORDER. I KNOW IT'S USUALLY SANTA SO JUST FOR A CHANGE, I THOUGHT THIS YEAR WE COULD HAVE AN 'XMAS COWBOY'!

I WANTED THE PROGRAM TO WORK WITH THE STANDARD LEVEL 2, MODEL 1 AND I WANTED TO STORE THE MINIMUM AMOUNT OF INFORMATION REQUIRED TO PRINT THE IMAGE. IT SEEMED TO ME THAT IT IS ONLY NECESSARY TO STORE THE START POSITION, LENGTH OF CHARACTERS REQUIRED AND THE 'END OF LINE' MARKER.

THE DATA STATEMENTS AS SHOWN FROM LINE 200 ONWARDS CONTAIN JUST THAT! THE FIRST 3 ITEMS REPRESENT (1) START PRINTING AT POSITION 32 (FROM THE LEFT) (2) FOR 8 CHARACTERS AND (3) Ø MEANS THAT'S IT FOR THIS LINE. THE LAST DATA STATEMENT,'99', JUST ENABLES THE PROGRAM TO END DURING THE NORMAL READ ROUTINE.

I ALSO WANTED THE PRINTED CHARACTERS TO SAY, IN THIS INSTANCE, MERRY XMAS AND NOT JUST BE A SELECTION OF RANDOM CHARACTERS.



I'D FORGOTTEN MANY OF THE RESTRICTIONS THIS WOULD PLACE ON ME REGARDING STRING HANDLING. (THE LAST SYSTEM I WORKED ON WAS A MCDONNELL DOUGLAS RUNNING 'PICK' WHICH SUPPORTS 'DATABASIC'. THIS BASIC HAS EXCEPTIONAL STRING HANDLING CAPABILITIES, WHICH I HAD BECOME ACCUSTOMED TO). ANOTHER PROBLEM ENCOUNTERED, WHICH, AGAIN, I'D FORGOTTEN, WAS THAT MANY OF THE LEWIS ROSENFELDER'S SUPERB STRING HANDLING ROUTINES FROM BF&B REQUIRE DISK BASIC!

I ALSO CONSIDERED HAVE A SMALL MACHINE CODE ROUTINE TO FILL THE PRINTING STRING (LDIR IS MADE FOR SUCH EVENTS IF I REMEMBER CORRECTLY) BUT I DECIDED TO STICK TO POKEING 'COS I CAN'T REMEMBER (I'M EMBARRASSED TO ADMIT) HOW TO PASS THE NECESSARY VARIABLES TO A M/C PROGRAM!

SO, BRIEFLY, THIS IS HOW THE PROGRAM WORKS.

PROGRAM LINE 10 CLEARS A BIT OF STRING SPACE.
20 DECLARES THE FULL STRING, PART OF WHICH WILL BE PRINTED.
30 A SUBROUTINE WHICH INITIALISES THE PRINT LINE.
40 READS THE FIRST DATA ITEM (STARTING POSITION OF PRINTING).
50 CHECK TO SEE IF IT'S THE END OF A LINE.
60 READS THE NEXT DATA ITEM (LENGTH OF STRING TO PRINT).
70 INITIALISES A TEMPORARY VARIBLE WHICH CONTAINS THE DATA TO BE POKED INTO THE PRINT LINE.
80 FIND, WHERE IN MEMORY THE PRINT LINE IS STORED.
90-110 POKE THE TEMPORARY STRING INTO THE PRINT LINE.
120 BACK TO LINE 40 TO READ THE NEXT DATA STATEMENT.
130 SUBROUTINE TO SEND THE PRINT LINE OFF TO THE PRINTER.
140 RE-INITIALISE THE PRINT LINE AFTER PRINTING.
150 READ THE NEXT DATA STATEMENT.
160 PROGRAM FINISHED IF IT'S = 99

THE XMAS COWBOY

```
10 CLEAR500
20M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMAS
MERRYXMAS"
30 GOSUB 180
40  READ S
50  IF S=0 GOSUB 130
60  READ L
70     T$=MID$(M$,S,L)
80     X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256
90        FOR Y=1TOL
100       POKE X+S+Y,ASC(MID$(T$,Y,1))
110       NEXT Y
120 GOTO 40
130 LPRINT P$
140 GOSUB180
150 READ S
160 IF S=99 THEN END
170 RETURN
180 P$=STRING$(63,32)
190 RETURN
200 DATA 32,8,0,26,18,0,25,22,0,24,3,28,20,0,23,3,27,22,0
210 DATA 22,3,27,22,0,22,2,26,24,0,21,2,25,26,0
220 DATA 11,7,20,3,25,26,57,3,0,6,45,56,5,0
230 DATA 3,48,55,3,59,3,0,2,19,24,28,54,7,0
240 DATA 1,3,5,19,27,31,59,2,0,1,2,4,23,31,30,0
250 DATA 1,2,4,27,35,24,0,1,32,51,7,0,2,2,5,34,44,11,0
260 DATA 3,2,6,47,0,4,2,7,44,0,5,3,9,39,49,1,0,7,35,49,2,0
270 DATA 10,31,49,2,0,16,25,48,2,0,18,2,21,2,24,21,47,3,0
280 DATA 18,2,24,16,43,2,47,2,0,19,4,25,18,46,3,0
290 DATA 21,2,25,19,46,2,0,19,4,26,15,45,3,0
300 DATA 17,6,26,15,45,2,0,16,2,20,4,26,11,43,5,0
310 DATA 15,2,20,4,26,11,42,6,0,15,3,21,4,26,21,0
320 DATA 16,3,27,15,43,3,0,17,23,42,3,0,18,20,41,3,0
330 DATA 30,3,34,3,41,2,0,31,3,40,2,0,32,4,38,3,0,34,5,0,99
```

--== o0o ==--

HERE'S A NEAT AND SMALL ONE-LINER FROM AN ORIGINAL IDEA FROM STEPHEN ROY HUGLI

```
1 CLS:A=15360:B=16383:FORN=0TO512:POKEA+N,RND(63)+128:POKEB-N,RN
D(63)+128:NEXTN:RUN
```

```
                              YXMASMER
                            ASMERRYXMASMERRYXM
                          MASMERRYXMASMERRYXMASM
                        XMA MERRYXMASMERRYXMASME
                      YXM SMERRYXMASMERRYXMASMER
                    RYX  SMERRYXMASMERRYXMASMER
                    RY  ASMERRYXMASMERRYXMASMERR
                  RR  MASMERRYXMASMERRYXMASMERRY
          ERRYXMA  ERR  MASMERRYXMASMERRYXMASMERRY       RRY
      XMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRY      ERRYX
   RRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRY    MER YXM
  ERRYXMASMERRYXMASME   XMASMERRYXMASMERRYXMASMERRYX  SMERRYX
MER YXMASMERRYXMASMERRY   SMERRYXMASMERRYXMASMERRYXMASMER YX
ME RYXMASMERRYXMASMERRYXMA    RYXMASMERRYXMASMERRYXMASMERRYX
ME RYXMASMERRYXMASMERRYXMASMER    ASMERRYXMASMERRYXMASMERR
MERRYXMASMERRYXMASMERRYXMASMERRY                    XMASMER
 ER YXMASMERRYXMASMERRYXMASMERRYXMASME     ASMERRYXMAS
  RR XMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXM
   RY MASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRY
    YXM SMERRYXMASMERRYXMASMERRYXMASMERRYXMASME R
      MASMERRYXMASMERRYXMASMERRYXMASMERRY       RY
        MERRYXMASMERRYXMASMERRYXMASMERR         RY
            MASMERRYXMASMERRYXMASMERR        RR
              SM RR XMASMERRYXMASMERRYXMA  ERR
              SM    XMASMERRYXMASMER   MA  ER
               MERR  MASMERRYXMASMERRYX   MER
                 RR  MASMERRYXMASMERRYXM  ME
               MERR   ASMERRYXMASMERR    SME
             ASMERR   ASMERRYXMASMERR    SM
            MA  ERRY  ASMERRYXMAS       MASME
           XM   ERRY  ASMERRYXMAS     XMASME
           XMA   RRYX ASMERRYXMASMERRYXMASM
            MAS         SMERRYXMASMERRY MAS
             ASMERRYXMASMERRYXMASMER  XMA
              SMERRYXMASMERRYXMASM   YXM
                          RRY MAS    YX
                           RYX      RY
                            YXMA  ERR
                              MASME
```



MERRY CHRISTMAS EVERYONE, AND A HAPPY NEW YEAR

WELL, THAT'S ABOUT IT FOR THIS ISSUE AND CLOSES OUR FIRST YEAR. THANKS TO ALL OF YOU WHO HAVE EMAILED ME. THE ENCOURAGEMENT WAS MUCH APPRECIATED, AND, OF COURSE, IT WAS JUST NICE TO HEAR FROM ANYONE WITH AN INTEREST IN TANDY'S. WITH ANY LUCK, THE NEXT ISSUE WILL HIT THE STREETS IN MARCH 2008, SO THERE'S PLENTY OF TIME IF YOU WISH TO SEND IN AN ARTICLE, REQUEST FOR HELP, OR ANYTHING OF INTEREST TO THE TANDY 8-BIT WORLD. PLEASE EMAIL ME AT

DUSTYM@BEEB.NET

OH! BEFORE I FORGET, I CAME ACROSS A RADIO-SHACK SHOP IN LANCASTER THE OTHER WEEK. I THOUGHT THAT THEY HAD ALL CLOSED DOWN LONG AGO. DOES ANYONE KNOW OF ANYMORE OUT THERE?

TRS8BIT



WELCOME TO THIS, THE FIRST EDITION OF 2008. THANKS FOR TAKING THE TIME TO DOWNLOAD AND READ IT! DON'T FORGET THAT THE NEWSLETTER FORMAT IS DESIGNED TO PRODUCE AN 'A4' BOOKLET WHEN PRINTED. I FIND THAT AN 'A5' SIZED BOOKLET IS MUCH EASIER TO HOLD AND READ.
I HOPE THAT, ONCE AGAIN, I'VE MANAGED TO FIND INTERESTING ITEMS TO CATER FOR A SELECTION OF TASTES AND INTERESTS.
THE MODEL 1 DOES SEEM TO DOMINATE BUT NEVER HAVING OWNED ANY OTHER TRS-80 I'M A BIT IN THE DARK WITH THE LATER MODEL 3'S AND 4'S.
IF THERE IS ANY SUBJECT YOU WISH TO BE FEATURED, PLEASE EMAIL ME WITH SUGGESTIONS AND I'LL DO MY BEST TO OBLIGE.

I'VE NOTICED, ON EBAY, OVER THE LAST 6 TO 8 WEEKS, 2 VIDEO GENIES, 3 MODEL 100/2'S AND VARIOUS MODEL 4 COME UP FOR SALE. I WAS SURPRISED AT HOW LITTLE THEY FETCHED. ONE OF THE GENIES (A MODEL 3003) WENT FOR JUST UNDER £40. ONE OF THE MODEL 100 ONLY FETCHED 99P AND A MODEL 4P ONLY MANAGED JUST OVER £15.

WHILST RUMMAGING THROUGH MY BIT AND PIECES, I CAME ACROSS ITEMS FOR BOTH THE MODEL 100 AND THE VIDEO GENIE. THIS LEAD ME TO DISCOVER ANOTHER INTERESTING WEB SITE IN NEW ZEALAND. I'VE PUT A LINK TO THIS ON TRS-80.ORG.UK

THIS EDITION INCLUDES AN ARTICLE BY KEN ROBINSON WHICH LISTS DETAILS OF THE GENIE'S 50 WAY BUS AS MATCHED TO THE MODEL 1'S 40 WAY BUS. (KNUTT HAS VARIOUS PIECES OF GENIE INFORMATION ON HIS WEB SITE. IT'S WELL WORTH A VISIT IF YOU'RE INTERESTED).

THERE'S A CLEAR AND COMPREHENSIVE WALK-THROUGH INSTRUCTION SET FOR DOWNLOADING MODELS 1 AND 100 USING ST-80 III AND DETAILS OF HOW TO BUILD A NULL MODEM. BOTH OF THESE ARTICLES WERE ORIGINALLY WRITTEN BY LEON HELLER AND PUBLISHED IN NATGUG NEWS.

TALKING OF LEON, HE WAS THE FIRST PERSON I CAME ACROSS WHO SPARKED MY INTEREST IN THE LANGUAGE 'C'. IF I REMEMBER CORRECTLY, THERE WAS A VERSION OF 'SMALL C' AVAILABLE FOR THE MODEL 1, BUT IT WAS ONLY PRODUCED FOR DISK SYSTEMS.
ON EARLY PC'S I BECAME INVOLVED WITH THE AMERICAN 'MIX' 'C' LANGUAGE. IT HAD A BIG USER BASE IN THE STATES WITH MASSIVE AMOUNTS OF FREEWARE FUNCTIONS. BUT AS THE YEARS PROGRESSED, I FOUND THE CURLY, CURVED AND STRAIGHT BRACKETS MORE AND MORE DIFFICULT TO SEE (NO PUN INTENDED!!). SO I MOVED BACK TO COBOL, PICK AND AS A CONSEQUENCE BASIC.
I STILL THINK THAT 'C' IS A NEAT, POWERFUL AND COMPACT LANGUAGE, HOWEVER, AT TIMES, IT CAN BE A BIT ON THE OBTUSE SIDE!

ON THE SUBJECT OF 'C' I'VE FOUND (ON ONE OF THE MANY SHAREWARE CD'S I HAVE) A NICE LITTLE 'C' SOURCE PROGRAM FOR USE ON COMPUTERS WITH ONLY UPPER CASE.

AND WHILE I'M MENTIONING THE LACK OF LOWER CASE, I'VE ALSO INCLUDED ONE OF THE MOST USEFUL LOWER CASE DRIVERS I'VE EVER USED. (IT WORKS GREAT WITH MATTHEW REED'S MODEL 1 EMULATOR, WHICH, BY THE WAY, HAS HAD ANOTHER UPDATE. SEE TRS-80.ORG FOR DETAILS).
ALL OF THE ONE-LINERS AND PROGRAMS FOR TRS8BIT ARE EITHER WRITTEN, DEVISED OR TESTED USING THE EMULATOR. I'M QUITE HOOKED ON THE WINDOWS VIRTUAL CASSETTE!



THAT'S ABOUT IT FOR NOW, KEEP IN TOUCH.

DUSTY

--== o0o ==--



AT THE
READY>
PROMPT

WITHIN LEVEL II BASIC THERE ISN'T AN XOR INSTRUCTION (ONLY AND, OR, NOT). HOWEVER IT IS POSSIBLE, ACCORDING TO JOHN PHELAN, TO USE THE FOLLOWING :-

A XOR B = (A OR B)-(A AND B)

AND OBTAIN THE CORRECT RESULT.

IN THE LAST EDITION I MENTIONED THE PRINT USING "!" SIGN SO IT SEEMS ONLY FAIR TO 'FEATURE' THE OTHER STRING OUTPUT CODE WHICH IS THE PERCENT "%" SIGN.
THE PERCENT SIGN MARKS THE BEGINNING AND THE END OF A PRINT LOCATION TO BE FILLED WITH STRING DATA. BETWEEN THE 2 PERCENT SIGNS IS A GIVEN NUMBER OF SPACES. THESE SPACES DEFINE THE WIDTH OF THE STRING FIELD. FOR EXAMPLE, THE EXPRESSION
"% %" DEFINES A STRING FIELD OF 4 POSITIONS. (THE BEGINNING AND END PERCENT SIGNS COUNT AS ONE SPACE EACH.) WHEN YOU USE THIS PRINT MODE, THE COMPUTER TAKES THE OPERAND STRING SUPPLIED BY YOU AND TRIES TO PUT IT INTO THE 'PRINT USING' FIELD.
FOR EXAMPLE -

```
10 PRINT USING "%    %";
"DUSTY"
```

WILL PRINT - DUST.
ALL CHARACTERS THAT DON'T FIT INTO THE 'PRINT USING' STATEMENT ARE TRUNCATED.
ANY PERCENT SIGN THAT IS NOT BEGINNING OR ENDING A STRING FIELD IS REGARDED AS JUST ANOTHER CHARACTER.

YOU CAN, OF COURSE, PRE-DEFINE YOUR PRINT USING STRING TO SAVE MEMORY (AND EFFORT), FOR EXAMPLE -

```
10 A$="%          %"
20 B$="27 LETSBIE AVENUE"
20 PRINT USING A$;B$
```

WILL PRINT - 27 LETSBIE.
THIS IS A NEAT LITTLE DEVICE FOR LIMITING THE WIDTH USED BY FIELDS WITHIN PRINT OUTS OR SCREEN DISPLAYS.

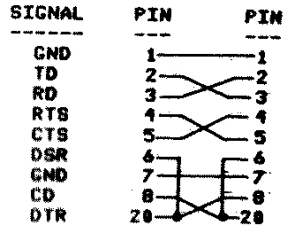
--== o0o ==--

HERE'S A GREAT ONE-LINER FOR DISPLAYING CHARACTERS RAPIDLY TO THE SCREEN BY PATRICK BOYLE

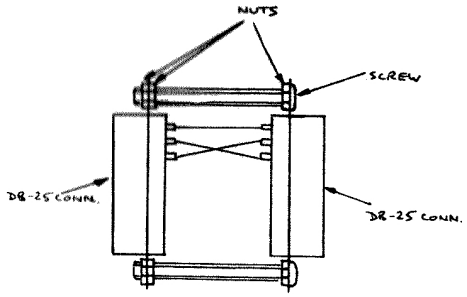
```
0 CLEAR22:A$=STRING$(22,32):J=VARPTR(A$):I=PEEK(J+1)+256*PEEK(J+
2):FORK=ITOI+21:READZ:POKEK,Z:NEXT:POKE16526,PEEK(J+1):POKE16527
,PEEK(J+2):FORX=1TO2:POKEI+10,RND(255):L=USR(0):X=1:NEXT:DATA33,
0,60,17,1,60,1,255,3,54,0,237,176,6,5,33,0,0,43,124,181,201
```

IF YOU ARE CONNECTING A COMPUTER TO ANOTHER VIA THEIR RS-232 SERIAL PORTS, YOU NEED A DEVICE CALL A 'NULL MODEM'. THIS IS JUST A CABLE WITH APPROPRIATE CONNECTORS, WIRED IN SUCH A WAY THAT EACH COMPUTER APPEARS AS A MODEM TO THE OTHER. WITH A COUPLE OF DB-25 CONNECTORS AND A FEW PIECES OF WIRE, TWO SCREWS AND A FEW NUTS, YOU CAN EASILY BUILD A NULL MODEM FOR A FEW POUNDS IN ABOUT 20 MINUTES.

ALL YOU DO IS PLACE THE CONNECTORS BACK TO BACK, AND LINK THE PINS BETWEEN THE CONNECTORS ACCORDING TO THE TABLE BELOW.



THEN YOU FIX THE TWO CONNECTORS TOGETHER USING THE SCREWS AND NUTS AS SHOWN BELOW.



IF YOU WANT TO BE REALLY POSH, YOU COULD USE STUDDING, SPACERS AND A SMALL BOX OF COURSE.

I USED A MALE AND FEMALE CONNECTOR, AS THIS ENABLES ME TO CONNECT MY MODEL 1 OR MODEL II TO MOST OTHER SYSTEMS.

BY WIRING THE PINS ON A ONE TO ONE BASIS, WITH CONNECTORS OF THE APPROPRIATE GENDER, YOU CAN EASILY MAKE WHAT IS TERMED AS A GENDER-CHANGER.

--== o0o ==--

TRS-80 MODEL 1/GENIE EDGE CONNECTOR CROSS-REFERENCE FROM AN ORIGINAL ARTICLE BY KEN ROBINSON

THE FOLLOWING TABLE SHOWS THE PIN INTER-CONNECTIONS TO OBTAIN THE TRS-80 BUS FROM THE VIDEO GENIE. IN THE TABLE THE SYMBOLS USED EITHER SIDE OF THE GENIE PIN NUMBER HAVE THE FOLLOWING MEANING.
NN INDICATES AN ADDITION TO THE GENIE BUS
#NN# INDICATES A SIMILARITY BETWEEN THE GENIE AND THE TRS-80 BUT NOT AN IDENTITY.

AN ACULAB FLOPPY TAPE AND AN LNW EXPANSION BOARD CONTROLLING DISKS CAN BE RUN OFF AN EXPANSION BUS USING THE FOLLOWING SUBSTITUTIONS-
PIN 12 OUT* CAN BE OBTAINED BY OR'ING GENIE PINS 40 AND 38
PIN 13 VR* CAN BE OBTAINED BY OR'ING GENIE PINS 40 AND 43
PIN 14 INTAK* CAN BE OBTAINED BY OR'ING GENIE PINS 38 AND 45
PIN 15 RD* CAN BE OBTAINED BY OR'ING GENIE PINS 41 AND 43
PIN 19 IN* CAN BE OBTAINED BY OR'ING GENIE PINS 38 AND 41

SUITABLE OR GATES CAN BE FOUND ON THE 74LS32, WHICH IS A QUAD ARRAY. UNLESS INTAK* IS DEFINITELY REQUIRED IR CAN BE LEFT CUT FOR MOST APPLICATIONS. IT IS WORTH NOTING THAT RD* AND WR* ON THE TRS-80 ARE IN FACT □NENRD□ AND □MEMWR□ AND

TRS-80			VIDEO GENIE		
1	RAS*	43	1	GND	8,37,29
2	SYSRES*	#48#	2	GND	8,37,29
3	CAS*	*13*	3	A7	36
4	A10	26	4	A6	38
5	A12	29	5	A5	35
6	A13	27	6	A4	31
7	A15	21	7	A1	27
8	GND	1,2,49,50	8	A3	34
9	A11	28	9	A2	40
10	A14	23	10	A0	25
11	A8	22	11	D5	28
12	OUT*	*32*	12	D2	32
13	WR*	#40#	*13*	□CAS□	3
14	INTAK*	--	14	D1	22
15	RD*	#41#	15	D0	30
16	MUX	*25*	16	D3	26
17	A9	24	17	D7	20
18	D4	20	18	D6	24
19	IN*	*33*	19	VCC	--
20	D7	17	20	D4	18
21	INT*	31	21	A15	7
22	D1	14	22	A8	11
23	TEST*	#39#	23	A14	10
24	D6	16	24	A9	17
25	A0	10	*25*	MUX	16
26	D3	16	26	A10	4
27	A1	7	27	A13	6
28	D5	11	28	A11	9
29	GND	1,2,49,50	29	A12	5
30	D0	15	30	PHI	--
31	A4	6	31	PINT	21
32	D2	12	*32*	□OUT□	12
33	WAIT*	37	*33*	□IN□	15
34	A3	6	34	□PHLDA□	--
35	A5	5	35	□PHAN□	--
36	A7	3	36	□HALT□	--
37	GND	1,2,49,50	37	□PWAIT□	--
38	A6	4	38	□ILRQ□	--
39	GND (LEV 2 ONLY)		#39#	□PHOLD□	23
40	A2	9	#40#	□WR□	13
			#41#	□RD□	15
			42	□CCDBS□	--
			43	□MREQ□	1
			44	□D0DBS□	--
			45	□MI□	--
			46	□RESET□	--
			47	RFSH	--
			#48#	□NNI□	2
			49	GND	8,37,29
			50	GND	8,37,29

NOT □RD□ AND □WR□, HENCE THE * AS OPPOSED TO THE BAR □CAS□ CAN BE OBTAINED FROM THE CPU BOARD INSIDE THE GENIE ON PIN 6 OF Z39. MUX CAN BE OBTAINED ON PIN 5 OF Z37. ALSO IF YOU WISH □OUT□ IS AVAILABLE ON PIN 8 OF Z15 WHILST □IN□ IS AVAILABLE ON PIN 6 OF Z15.

(THE ORIGINAL ARTICLE WAS IN VERY POOR CONDITION AND CONSEQUENTIALLY VERY DIFFICULT TO READ. I HOPE I'VE MANAGED TO COPY ALL THE DETAILS CORRECTLY).

HERE'S A NICE LITTLE MOD TO THE CASSETTE VERSION OF ELECTRIC PENCIL FROM AN ORIGINAL ARTICLE WRITTEN BY RODNEY SCHREINER.

FOR SEVERAL MONTHS I'VE BEEN USING THE CASSETTE VERSION OF PENCIL TO PREPARE MANUSCRIPTS AND LETTERS USING SINGLE SHEETS OF 8.5 X 11 INCH PAPER. SINCE PENCIL DOES NOT AUTOMATICALLY PAUSE AT THE END OF A PAGE, I HAVE TO WATCH THE PRINTER AND STOP THE PROGRAM MANUALLY SO I CAN INSERT A NEW SHEET OF PAPER.
THE TEDIOUSNESS OF THIS TASK FINALLY GOT TO ME, AND I DECIDED TO INVESTIGATE THE PROGRAM TO SEE IF I COULD MODIFY IT SO THAT IT WOULD STOP AT THE END OF EACH PAGE. AFTER A LENGTHY SEARCH THROUGH THE CODE, I LOCATED THE SPOT AT WHICH THE PROGRAM TESTS FOR THE FINAL LINE OF A PAGE. AT THIS POINT THE PROGRAM JUMPS TO THE PAGE SPACING ROUTINE. BY CHANGING ONLY 2 BYTES I ALTERED THIS JUMP TO A RETURN TO THE MAIN SYSTEM. NOW, WHEN THE PRINTER REACHES THE END OF PAGE, IT STOPS AND I CAN LOAD A FRESH SHEET OF PAPER AT MY LEISURE. ALL THAT IS NECESSARY TO CONTINUE PRINTING IS TO MOVE THE CURSOR TO THE START OF THE NEXT PAGE I REQUIRE PRINTING AND OFF I GO.

THE CHANGES I'VE MADE ARE DETAILED BELOW:

CHANGE

4F3FH FROM 28 TO C8
4F40H FROM 06 TO 00

ONLY BYTES 4F3F AND 4F40 ARE CHANGED. THIS CAN BE ACCOMPLISHED BY USING T-BUG, RELOCATED TO HIGH MEMORY. AFTER THE 2 CHANGES ARE COMPLETED, A NEW SYSTEM TAPE CAN BE WRITTEN FROM T-BUG. THE STARTING ADDRESS IS 4350H, THE ENDING ADDRESS IS 5374H AND THE ENTRY POINT IS 4350H.
THIS NEW VERSION OF PENCIL CANNOT PRINT CONTINUOUSLY, SINCE THE ALTERATIONS ABOVE FORCE IT TO STOP AT THE END OF EACH PAGE. HOWEVER, IF I NEED CONTINUOUS PRINTING I CAN USE MY ORIGINAL PENCIL CASSETTE.

--== o0o ==--

FILE TRANSFERS ON THE MODELS 1 AND 100 BY LEON HELLER

IF YOU ARE USING THE TANDY RS-232 BOARD, IT IS NOT ADVISABLE TO JUST PLUG THE CABLE FROM THE MODEL 1 INTO THE BACK OF THE MODEL 100. BOTH MACHINES ARE CONFIGURED AS 'DATA TERMINAL EQUIPMENTS' (DTE) AND ALTHOUGH YOU CAN REVERSE PINS 2 AND 3 ON THE RS-232 BOARD IN THE MODEL 1, THE OTHER SIGNALS WILL BE TRYING TO OUTPUT TO ONE ANOTHER. THIS SHOULD NOT RESULT IN ANY DAMAGE, BUT WILL PREVENT PROPER HAND-SHAKING BETWEEN THE SYSTEMS. YOU WILL NEED WHAT IS TERMED A NULL MODEM CONNECTOR, WITH ALL THE SIGNAL CROSSED OVER. THESE CAN BE OBTAINED AT MANY LOCAL COMPUTER SUPPLY SHOPS OR YOU CAN EASILY MAKE ONE FOR YOURSELF. (SEE THE ARTICLE ON PAGE 3 ED.)

ASSUMING YOU HAVE A SMART TERMINAL PROGRAM LIKE ST80-III ON THE MODEL 1, YOU SET THE SYSTEM UP FOR 300 BAUD, 7 DATA BITS, EVEN PARITY, 1 STOP BIT, TRANSMIT AUTO LINE FEED ON, AND RECEIVE

AUTO LINE FEED ON.
ON THE MODEL 100, SELECT 'TELECOM' FROM THE MAIN MENU, THEN SET THE RS-232 PARAMETERS USING F3 (STAT: 37E1E). NOW USE F4 TO GET INTO TERMINAL MODE. KEYS PRESSED ON THE MODEL 1 SHOULD BE ECHOED ON THE MODEL 100, AND VICE-VERSA, SHOWING THAT ALL CONNECTIONS ON BOTH SYSTEMS ARE FUNCTIONING.

IF YOU ARE TRANSFERRING A FILE FROM THE MODEL 1 TO THE MODEL 100, YOU NOW GET THE FILE INTO THE MODEL 1 BUFFER, USING (SHIFT) G. RESPOND TO THE TRANSMISSION SPEED QUESTION WITH 1, TO BE ON THE SAFE SIDE. ON THE MODEL 100, USE F2 TO SELECT THE DOWNLOAD MODE. THE SYSTEM WILL REQUEST A FILENAME. AFTER INPUTTING THE FILENAME, F2 WILL BE DISPLAYED IN REVERSE VIDEO. NOW PRESS (SHIFT) O ON THE MODEL 1 KEYBOARD, AND THE FILE WILL BE TRANSFERRED FROM THE MODEL 1 TO THE MODEL 100, BEING SIMULTANEOUSLY DISPLAYED ON THE MODEL 1 AND MODEL 100 SCREENS. WHEN THE FILE HAS BEEN TRANSFERRED ST80-III 'READY' WILL BE DISPLAYED ON THE MODEL 1. IF YOU PRESS F2 ON THE MODEL 100, THE FILE WILL BE CLOSED, AND YOU CAN EXIT FROM 'TELECOM' BY PRESSING F8 TWICE.

TO UPLOAD FROM THE MODEL 100 TO THE MODEL 1, FIRST SET THE MODEL1 UP FOR RECEIVE AUTO LINE FEED ON (SHIFT B), AND OPEN THE MEMORY BUFFER (SHIFT) C. KEYING F3 ON THE MODEL 100 WILL RESULT IN THE REQUEST FOR THE FILE NAME. INPUT THE FILE NAME, AND RESPOND TO THE WIDTH QUESTION WITH (ENTER). 'UP' WILL BE DISPLAYED IN REVERSE VIDEO, AND THE FILE WILL BE TRANSFERRED TO THE MODEL 1, BEING DISPLAYED ON THE MODEL 1 SCREEN AS TRANSFER OCCURS.

WHEN THE FILE HAS BEEN TRANSFERRED, 'UP' ON THE MODEL 100 SCREEN WILL CHANGE FROM REVERSE VIDEO TO NORMAL. KEYING (SHIFT) X ON THE MODEL 1 WILL CLOSE THE MEMORY BUFFER, AND THE FILE CAN BE TRANSFERRED TO DISK USING (SHIFT) F.

--== o0o ==--

THIS PROGRAM IS ONE OF THE MOST USEFUL ONES I'VE EVER HAD FOR A 16K MODEL 1! (IT WILL ACTUALLY FIT DOWN TO A ONE-LINER). IT'S A SUPER LITTLE LOWER-CASE DRIVER. IT WORKED WELL (SUBJECT TO ANY NECESSARY HARDWARE MOD BEING COMPLETED) WITH ORDINARY LEVEL II BASIC, ACULAB XBAS, DISK BASIC AND MORE RECENTLY, MATTHEW REEDS EMULATOR.
THE REALLY NICE THING ABOUT IT WAS THAT IT WORKS IN REVERSE! I.E. WITH THE SHIFT HELD DOWN, LOWER-CASE LETTERS ARE PRODUCED. THIS AVOIDED ACCIDENTAL LOWER-CASE ENTRIES. A REAL BOON FOR PROGRAM DEVELOPMENT AND DE-BUGGING!

```
10 REM FOR 16K MACHINES SET MEM SIZE TO 32737
20 POKE 16553,255:FORI=32738TO32767:READJ:POKEI,J:NEXT
30 POKE 16414,226:POKE16415,127
40 DATA 221,110,3,221,102,4,218,154,4,221,126,5,183,40,1
50 DATA 119,121,254,32,218,6,5,254,128,210,166,4,195,125,4
```

"CONVERT"
WRITTEN BY LEOR ZOLMAN

THIS PROGRAM CONVERTS REGULAR C SOURCE FILES INTO A FORMAT SUITABLE FOR EDITING ON THE TRS-80 (OR ANY UPPER-CASE-ONLY SYSTEM.) SINCE THERE ARE QUITE A FEW ASCII CHARACTERS THAT NEED TO BE REPRESENTED EVEN THOUGH THEY DON'T SHOW UP ON UPPER-CASE-ONLY SYSTEMS, A SPECIAL NOTATION HAS BEEN CREATED FOR REPRESENTING THESE CHARACTERS. THE POUND SIGN IS USED AS A SORT OF 'SHIFT' KEY, WITH THE LETTER FOLLOWING THE POUND SIGN DENOTING THE SPECIAL CHARACTER NEEDED. NOTE THAT THE C COMPILER DOES NOT RECOGNIZE THIS SPECIAL SCHEME, AND BEFORE YOU CAN COMPILE A SOURCE FILE CONTAINING THE SPECIAL CODES YOU MUST PREPROCESS THE FILE USING THE "CCØT" COMMAND.

THE SPECIAL CODES AND THE CHARACTERS THEY REPRESENT ARE:

```
#L   LEFT BRACKET (FOR SUBSCRIPTING) (5B HEX)
#R   RIGHT BRACKET           (5D HEX)
#C   CIRCUMFLEX (BITWISE "NOT")  (7E HEX)
#H   UP-ARROW (EXCLUSIVE "OR" OPERATOR)
                             (5E HEX)
#V   VERTICAL VAR (LOGICAL AND BITWISE "OR")
                             (7C HEX)
#B   BACKSLASH (FOR ESCAPE SEQUENCES)(5C HEX)
#U   UNDERSCORE                   (5F HEX)
```

FOR EXAMPLE, THE COMMAND

```
A>CONVERT FOO.C BAR.CT
```

WILL EXPECT FOO.C TO BE A NORMAL C SOURCE FILE ON DISK, AND WILL CONVERT IT INTO A FILE NAMED BAR.CT. THE FILE BAR.CT MAY THEN BE EDITED TO YOUR TASTE, BUT REMEMBER TO PREPROCESS IT WITH "CCØT" BEFORE APPLYING THE C COMPILER.

AS YOU MAY HAVE GATHERED FROM ALL THIS, THE LANGUAGE "C" WAS NEVER INTENDED TO BE IMPLEMENTED ON A SYSTEM HAVING UPPER-CASE ONLY; NEVERTHELESS, HERE IS A WAY FOR IT TO BE DONE.

THIS PROGRAM IS RATHER SIMPLE, AND THUS IT WILL NOT RECOGNIZE THAT SPECIAL CHARACTERS IN QUOTES ARE NOT SUPPOSED TO BE CONVERTED.

```
#DEFINE LEFTCURLY ØX7B
#DEFINE RIGHTCURLY ØX7D
#DEFINE LEFTBRACK ØX5B
#DEFINE RIGHTBRACK ØX5D
#DEFINE CIRCUM ØX7E
#DEFINE UPARROW ØX5E
#DEFINE VERTIBAR ØX7C
#DEFINE BACKSLASH ØX5C
#DEFINE UNDERSCORE ØX5F

CHAR IBUF[134], OBUF[134];

MAIN(ARGC,ARGV)
INT ARGC;
CHAR *ARGV[];
BEGIN
     INT FD1, FD2;
     CHAR C;
     IF (ARGC != 3) BEGIN
          PRINTF("USAGE: CONVERT OLD NEW <CR>\N");
          EXIT();
     END
```

```
     FD1 = FOPEN(ARGV[1],IBUF);
     IF (FD1 == -1) BEGIN
          PRINTF("NO SOURCE FILE.\N");
          EXIT();
     END
     FD2 = FCREAT(ARGV[2],OBUF);
     IF (FD2 == -1) BEGIN
          PRINTF("CAN'T OPEN OUTPUT FILE.\N");
          EXIT();
     END

     WHILE ((( C = GETC(IBUF)) != ØX1A) && C != 255) BEGIN
          SWITCH (C) BEGIN
           CASE LEFTCURLY: PUTST(" BEGIN ");
                BREAK;
           CASE RIGHTCURLY: PUTST(" END ");
                 BREAK;
           CASE LEFTBRACK: PUTSPEC('L');
                BREAK;
           CASE RIGHTBRACK: PUTSPEC('R');
                BREAK;
           CASE CIRCUM: PUTSPEC('C');
                BREAK;
           CASE UPARROW: PUTSPEC('U');
                BREAK;
           CASE VERTIBAR: PUTSPEC('V');
                BREAK;
           CASE BACKSLASH: PUTSPEC('B');
                BREAK;
           CASE UNDERSCORE: PUTSPEC('U');
                BREAK;
           DEFAULT: PUTC(TOUPPER(C),OBUF);
          END
     END

     IF (C==255) C = ØX1A;  /* DIGITAL RESEARCH....WOW. */
     PUTC(C,OBUF);
     FFLUSH(OBUF);
END

PUTST(STRING)
CHAR *STRING;
BEGIN
     WHILE (*STRING) PUTC(*STRING++,OBUF);
END

PUTSPEC(C)
CHAR C;
BEGIN
     PUTC('#',OBUF);
     PUTC(C,OBUF);
END
```



--=== o0o ===--

I HOPE YOU'VE ENJOYED THIS ISSUE. IF YOU HAVE, WHY NOT DROP ME A LINE AND TELL EVERYONE WHAT YOU'RE UP TO WITH YOUR TANDY? IF NOT. LET ME KNOW AND I'LL TRY AND IMPROVE THE NEXT ISSUE, WHICH BY THE WAY, SHOULD BE OUT EARLY JUNE Ø8.

AND FINALLY, HAS ANYONE A PDF COPY OF 'PATHWAYS THROUGH THE ROM' FROM IRA'S SITE? IF YOU HAVE, WOULD YOU EMAIL ME A COPY? IT'S THE SAME ADDRESS FOR ALL CORRESPONDENCE - TIA

DUSTYM@BEEB.NET

TRS8BIT



I CAN'T BELIEVE THAT THIS IS OUR 6TH EDITION OF TRS8BIT! TIME DEFINITELY GOES FASTER AS MY AGE INCREASES.

I HOPE YOU ENJOY THIS EDITION AS I FEEL THERE ARE SOME FASCINATING ARTICLE TO KEEP YOU AMUSED.

STAR BILLING IN THIS ISSUE MUST GO TO AN ARTICLE ROY T BECK HAS UPDATED WHICH WAS ORIGINALLY WRITTEN FOR TRSTIMES. IT'S FOR ANY BRAVE SOUL WHO WOULD LIKE TO ATTACH A HARD DRIVE TO A MODEL 1, 3 OR 4. IF ANYONE HAS (OR HAS HAD!) A GO I'D LOVE TO HEAR HOW YOU GOT ON. THIS MUST BE ONE OF THE MOST EXCITING AND WELL RESEARCHED ARTICLE I'VE EVER COME ACROSS FOR THE TRS-80 COMMUNITY.

THERE'S A SMASHING ONE-LINER TO CALCULATE PRIME NUMBERS.
JUST AS A MATTER OF INTEREST, I CAN LOAD-UP MATTHEW REED'S EMULATOR, RUN THE PROGRAM, PRINT OUT THE ANSWERS AND CLOSE DOWN, IN LESS TIME THAT DEE'S LAPTOP COMPUTER (UNDER WINDOWS XP) TAKES TO POWER ON!

THERE'S THE USUAL 'AT THE READY PROMPT' AND VARIOUS OTHER LITTLE SNIPPETS I'VE BEEN PLAYING AROUND WITH OVER THE LAST FEW WEEKS.

AT LONG LAST, I FELT I HAD TO BUCK-UP COURAGE AND HAVE A GO AT RUNNING MATHEW REED'S EMULATOR WITH DISKS! (A GIANT LEAP FOR MANKIND YOU MIGHT THINK). THE ONLY DOS I OWN IS NEWDOS80 V2 AND I'M ON A STEEP RE-LEARNING CURVE! (SEE, AT THE READY PROMPT, FOR WHAT I MEAN).
DURING THE 80'S I NEVER MANAGED TO AFFORD AN E.I. AND DISK DRIVES OF MY OWN, BUT I WAS LUCKY ENOUGH TO BE ABLE TO USE ONES OWNED BY FRIENDS. (I HAS HALF A DOZEN DISKS OF MY OWN AND A GENUINE COPY OF NEWDOS80).
AS YOU CAN IMAGINE, I'M REALLY ENJOYING MYSELF!
AS AN ASIDE, I'VE JUST BEEN 'ELECTED' TREASURER OF THE LOCAL BRANCH OF THE BRITISH PRINTING SOCIETY AND HAVE DECIDED TO DO ALL THE ACCOUNTANCY WORK ON MY M1 EMULATOR BY USING ELECTRIC PENCIL AND VISICALC. I'LL LET YOU KNOW HOW I GET ON!

THANKS TO EVERYONE WHO OFFERED ME COPIES OF PATHWAYS THROUGH THE ROM. IT WAS GREATLY APPRECIATED. THANKS MUST GO TO DAVID COOPER FOR BEING THE FIRST TO OFFER AND LETTING ME HAVE MANY OTHER BITS AND PIECES, SOME OF WHICH I HAVEN'T SEEN FOR YEARS.

KNUTT HAS RELEASED HIS LATEST PROGRAM 'PLAYCASS'. IT'S AVAILABLE AS A DOWNLOAD FROM HIS WEBSITE. THERE IS A LINK TO HIS SITE ON TRS-80.ORG.UK
HAVE A LOOK, IT'S, YET AGAIN, ANOTHER REMARKABLE PIECE OF SOFTWARE, JUST RELEASED FOR A 30 YEAR-OLD PIECE OF HARDWARE!!

THERE HAS BEEN A REDUCTION IN THE AMOUNT OF TRAFFIC ON THE UK'S EBAY FOR TRS-80 RELATED ITEMS FOR QUITE A WHILE. VARIOUS BOOKS SEEM TO FETCH A FEW POUNDS EACH

AND A COUPLE OF MODEL 4P'S RANGING FROM £12-£40 EACH. SOMEONE HAD BEEN ADVERTISING 270 ISSUES OF BYTE FOR SALE. WITH A STARTING BID OF £200 IT WAS TOO MUCH MONEY FOR ME, BUT THEY WOULD MAKE VERY INTERESTING READING.
EBAY.COM STILL MANAGES TO AMAZE ME WITH THE QUANTITY AND DIVERSITY OF ITEMS STILL BEING OFFERED FOR SALE. IF ONLY THE POSTAGE COSTS FROM THE STATES WERE NOT SO HIGH.

JUST BY WAY OF ADVANCE NOTICE, MY ISP, BEEB.NET, IS CLOSING DOWN AT THE END OF JUNE08. THIS MEANS A NEW HOME FOR TRS-80.ORG.UK HAS HAD TO BE FOUND PRETTY SHARPISH. I HOPE TO ACCOMPLISH THE CHANGE-OVER AS SEAMLESSLY AS POSSIBLE, BUT JUST IN CASE, I'LL ASK FOR YOU FORBEARANCE! IT ALSO MEANS, OF COURSE, MY EMAIL ADDRESS WILL CHANGE TO:-

DUSTYM@FABSITESUK.COM

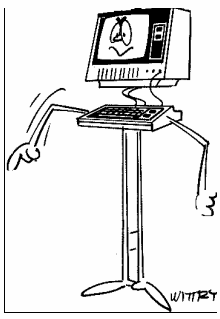
PLEASE USE THIS ADDRESS WITH IMMEDIATE EFFECT. I SHALL BE SENDING AN EMAIL TO EVERYONE WHO HAS CONTACTED ME OVER THE LAST TWO YEARS, SO HOPEFULLY NO CONTRIBUTION TO TRS8BIT WILL GO AMISS.

(CONTINUED ON PAGE 10)

AT THE
READY>
PROMPT



HAVING BITTEN THE BULLET AND STARTED USING MATTHEW REED'S EMULATOR RUNNING NEWDOS80 V2, HERE'S A TIMELY REMINDER OF SOMETHING (ELSE) I'D COMPLETELY FORGOTTEN. IN ORDER FOR THE CASSETTE PROGRAMS TO LOAD AND SAVE, A CMD"T" MUST BE ISSUED IMMEDIATELY BEFORE ANY BASIC TAPE INPUT/OUTPUT OPERATION. THESE OPERATIONS ARE TIMING SENSITIVE AND ARE AFFECTED BY INTERRUPT-DRIVEN TASKS SUCH AS TRACE & CLOCK.
JUST TO REMIND YOU, THE COMMANDS AFFECTED ARE :- CLOAD, CLOAD?, CSAVE, INPUT#-1,INPUT#-2, SYSTEM, PRINT#-1, AND PRINT#-2,. DON'T FORGET TO ENABLE INTERRUPTS WHEN YOU HAVE FINISHED, WITH THE CMD"R" COMMAND. (AND YES, BEFORE YOU ASK, I DID!)



AND FINALLY, JUST A LITTLE BIT OF HISTORY TURNED UP THE OTHER DAY. TUCKED AWAY INSIDE A BOOK WAS A LITTLE CLOTH BADGE THAT SOMEONE HAD MADE FOR NATUG MEMBERS WHEN THE GROUP FIRST STARTED. (BEFORE THE NAME WAS CHANGED TO NATGUG!).
IT'S STRANGE, I CAN QUITE CLEARLY REMEMBER BRIAN ISSUING THEM BUT I CAN'T REMEMBER WHO IT WAS WHO HAD THEM MADE. IS THERE'S ANYONE OUT THERE WITH A BETTER MEMORY THAN ME WHO CAN REMEMBER WHO IT WAS?



--== O0O ==--

LOUIS PELLETIER HAS WRITTEN A CRACKING ONE LINER TO CALCULATE PRIME NUMBERS. IT'S SURPRISINGLY FAST FOR JUST LEVEL 2 BASIC. IT CALCULATES ALL THE PRIMES UP TO 1000 IN UNDER 10 SECONDS ON A 16K LEVEL II MACHINE. DON'T BELIEVE ME? JUST TRY IT!

```
1 DEFINTA-Z:CLS:L=1000:DIMA(L):V=INT(SQR(L)):FORB=3TOLSTEP2:IFA(
B)NEXT:ELSEIFB>VPRINTB,:NEXT:ELSEFORC=B*BTOLSTEP2*B:A(C)=1:NEXT:
PRINTB,:NEXT
```

SUMMARY OF RADIO SHACK HARD DRIVE PARAMETERS

AN UPDATED ARTICLE BY
ROY T BECK
WHICH FIRST APPEARED IN
TRSTIMES

INTRODUCTION

RECENTLY I RECEIVED A CALL FROM A MAN WHO NEEDED TO KNOW HOW MANY TRACKS AND HEADS WERE ON A RADIO SHACK 70 MEG HARD DRIVE. SUCH A SIMPLE REQUEST, AND COMPLETELY NECESSARY IN ORDER TO PARTITION AND FORMAT A HARD DRIVE. BUT WHERE DO YOU FIND THE INFORMATION WHEN YOU NEED IT AT 1 AM? THE ANSWER SHOULD BE IN THE RADIO SHACK HARD DRIVE MANUALS, BUT IT IS ONLY PRESENTED IN A FRAGMENTARY AND INCOMPLETE FASHION, UNLESS YOU HAVE THE SERVICE MANUAL FOR THAT PARTICULAR DRIVE. IF YOU ARE DOING A DRIVE SWAP, LIKE HOT-RODDERS USED TO SWAP ENGINES, FORGET IT. RADIO SHACK DOESN'T EVEN WANT TO TALK TO YOU!
THIS ARTICLE WILL PROVIDE A QUICK SUMMARY OF THE PERTINENT FACTORS FOR THE VARIOUS DRIVES RADIO SHACK HAS USED IN OUR TRS HARD DRIVE PACKAGES. I WILL EXCLUDE THE 8.4 MEG DRIVE, AS THAT DRIVE IS NOT SUITABLE FOR THE MODEL 1, 111 AND 4 FAMILY.

MODEL 1, 3 AND 4 DRIVES

I WILL INCLUDE A FEW OTHER DRIVES WHICH YOU MAY RUN INTO. TO BEGIN WITH, ALL DRIVES SUITABLE FOR USE WITH RADIO SHACK CONTROLLERS ARE CATEGORISED AS MFM, (MEANING MODIFIED FREQUENCY MODULATION), AND HAVE THE SAME INTERFACE AS THE SEAGATE ST-412 OR SEAGATE ST-506 DRIVES. THIS INTERFACE PHYSICALLY CONSISTS OF TWO CARD-EDGE CONNECTORS, ONE WITH 20 CONDUCTORS, THE OTHER WITH 34, PLUS A 4 WIRE MALE POWER CABLE CONNECTOR. USABLE DRIVES IN THIS CATEGORY RANGE FROM 5 TO 70 MEGS. THE MAXIMUM HEAD COUNT AND CYLINDER COUNT WHICH CAN BE USE BY THE RADIO SHACK HARD DRIVE CONTROLLERS (HDC) IS 8 AND 1024, RESPECTIVELY. ALL OF THESE DRIVES ARE FULL HEIGHT UNLESS NOTED TO THE CONTRARY. INCIDENTALLY, THE BARE HARD DRIVE IS ALSO COMMONLY REFERRED TO AS THE 'BUBBLE', AND OCCASIONALLY I USE THAT TERM TO DISTINGUISH THE HARD DRIVE FROM THE OVERALL PACKAGE, WHICH INCLUDES THE HDC AND THE POWER SUPPLY, ALL IN A CASE. THE BUBBLE IS CALLED THAT BECAUSE THE HEADS AND PLATTERS ARE LOCATED INSIDE A SEALED DUST-TIGHT CHAMBER IN THE DRIVE. DON'T EVER OPEN THE SEALED CHAMBER, THAT SHOULD ONLY BE DONE IN A CLEAN ROOM.

Table 1

Hard Drive (Bubble) Characteristics

Mfg.	Model	Size	Hds	Cyls	Step Rate	Notes	Access
Tandon	TM-602S	5	4	153	10 usec	1	99 ms
Tandon	TM-501	5	2	306	10 usec	2	85
Tandon	TM-502	10	4	306	10 usec	3	85
Tandon	TM-503	15	6	306	10 usec	4	85
Tandon	TM-603S	12	6	230	10 usec	5	99
Seagate	ST-225	20	4	615	10 usec	6	65
Seagate	ST-412	10	4	306	10 usec	7	99
Seagate	ST-506	5	4	153	10 usec	8	
Quantum	Q-540	35	8	512	10 usec	9	45
Micropolis	1325T	70	8	1024	10 usec	10	28

NOTES FOR TABLE 1

1. THIS DRIVE WAS USED IN MOST RADIO SHACK 5 MEG BOXES, CAT NO 26-1130. SOME OF THESE DRIVES REQUIRE THE STEP RATE TO BE SET AT 6 OR EVEN MORE. THIS CORRESPONDS TO 3 MILLISECONDS. TRY THE DEFAULT VALUE OF 10 MICROSECONDS. IF THE STEP RATE IS TOO FAST, VERIFICATION WILL REPORT NUMEROUS BAD TRACKS. LF THIS OCCURS, REPEAT THE PARTITIONING WITH 6, WHICH SHOULD WORK.
2. APPARENTLY SOME RADIO SHACK 5 MEG BOXES HAD THIS DRIVE.
3. THIS DRIVE WAS NOT USED BY RADIO SHACK, BUT IS MENTIONED IN SOME OF THEIR SERVICE MANUALS.
4. THIS DRIVE WAS USED IN ALL THE RADIO SHACK 15 MEG BOXES, CAT NO 26-4155.
5. THIS DRIVE WAS USED IN ALL THE RADIO SHACK 12 MEG BOXES, CAT NO 26-4152.
6. THIS DRIVE IS WIDELY AVAILABLE AND WORKS WELL IN RADIO SHACK BOXES. IT IS A HALF-HEIGHT DRIVE, BUT IS A DROP-IN FIT IN PLACE OF A FULL HEIGHT DRIVE.
7. THIS DRIVE WAS NOT USED BY RADIO SHACK, BUT IS LISTED HERE BECAUSE IT IS ONE OF THE ''GENERIC'' MFM DRIVES FREQUENTLY MENTIONED.
8. THIS DRIVE WAS NOT USED BY RADIO SHACK, BUT IS LISTED HERE BECAUSE IT IS THE OTHER ''GENERIC'' MFM DRIVE FREQUENTLY MENTIONED.
9. THIS DRIVE WAS USED IN THE RADIO SHACK 35 MEG BOXES, CAT NO 26-4171. IT HAS AUTO-PARKING BUILT IN; IT PARKS ITSELF AT THE HIGHEST CYLINDER ON LOSS OF POWER.
10. THIS DRIVE WAS USED IN THE RADIO SHACK 70 MEG BOXES, CAT NO 26-4173.

THE INFAMOUS 'THREE WIRES'

THE 'THREE WIRES' L AM REFERRING TO ARE THREE WIRES WHICH RUN FROM A HARNESS IN THE HARD DRIVE CASE TO SPECIFIC POINTS ON THE BUBBLE, VARYING WITH THE ACTUAL BUBBLE USED IN THE SYSTEM. RADIO SHACK SOLDERED THESE THREE WIRES DIRECTLY TO THE PC BOARD OF THE BUBBLE, WHICH COULD EASILY BE DONE ON THE PRODUCTION LINE. IT DOES POSE A PROBLEM TO US USERS, ESPECIALLY WHEN WE WISH TO SWAP BUBBLES.

IT MAY BE USEFUL TO EXPLAIN THE PURPOSE OF THE THREE WIRES. THE WIRE COLOURS ARE AS FOUND IN A MASTER DRIVE; THE COLOURS ARE DIFFERENT IN A SLAVE DRIVE.

THE YELLOW WIRE RESPONDS TO THE DRIVE SELECT SIGNAL. THE SYSTEM CAN ELECTRICALLY ACCOMMODATE UP TO 4 DRIVES, ONE MASTER AND THREE SLAVES. THERE IS A DRIVE SELECT LINE FOR EACH OF THEM. THE SELECTED DRIVE HAS 0 VOLTS ON ITS SELECT LINE, AND THE OTHER THREE STAND AT + 5 VOLTS. WHEN THE MASTER DRIVE IS SELECTED, THE YELLOW WIRE WILL BE AT 0 VOLTS, AND THE GREEN LIGHT MAY BE ON. THE REASON FOR SAYING ''MAY'' IS THAT THE WHITE WIRE ALSO ENTERS INTO THE PICTURE.

THE WHITE WIRE SENSES 'SEEK COMPLETED', WHICH IS A SIGNAL RETURNED BY THE DRIVE TO THE HDC. WHILE THE DRIVE IS ACTIVELY STEPPING, THIS LINE WILL BE AT + 5 VOLTS, SIGNIFYING SEEK IS NOT COMPLETE. WHEN SEEK IS COMPLETE, THE LINE GOES LOW, AND THE WHITE WIRE WILL BE AT 0 VOLTS. THE WHITE AND YELLOW WIRES ARE NORED TO TURN ON THE GREEN LIGHT WHEN

BOTH ARE LOW. IF THE DRIVE IS STEPPING OR NOT SELECTED, THE GREEN LIGHT GOES DARK. THUS THE STEADY GREEN LIGHT MEANS THE DRIVE IS SELECTED AND IS NOT STEPPING, AND A FLICKERING GREEN LIGHT MEANS EITHER THE DRIVE IS STEPPING OR IS MOMENTARILY NOT SELECTED AS THE DOS CHECKS ON A FLOPPY DRIVE, OR DOES SOMETHING ELSE. NORMALLY THE GREEN LIGHT IS LIT ON THE MASTER DRIVE. IF YOU HAVE A SLAVE CONNECTED, ITS GREEN LIGHT WILL NORMALLY BE DARK.

THE ORANGE WIRE IS PART OF THE WRITE PROTECT CIRCUIT, AND SENDS + 5 OR Ø VOLTS TO THE HDC, THEREBY INFORMING THE LOGIC WHETHER THE WP SWITCH ON THE FRONT OF THE CASE IS DEPRESSED. WHEN THE SWITCH IS DEPRESSED AND THE RED LIGHT IS ON, THE ORANGE WIRE IS AT Ø VOLTS. WHEN THE WP LIGHT IS OFF, THE ORANGE WIRE IS AT +5 VOLTS.

A CAUTIONARY NOTE ON A QUIRK OF THE WRITE PROTECT CIRCUIT IS APPROPRIATE, THE RED LAMP IS ACTIVE, IN A SENSE, EVEN WHEN IT IS DARK. WHEN THE LAMP IS DARK, +5 VOLTS IS PASSED THROUGH IT TO THE REMAINING LOGIC IN THE HDC. THE QUIRK IS THAT IF THE LAMP BURNS OUT, OR FAILS TO MAKE GOOD CONTACT IN ITS SOCKET, THEN THE +5 VOLTS DOES NOT GET TO THE LOGIC, AND THE HDC SEES Ø VOLTS, WHICH IT INTERPRETS AS A WRITE PROTECTED CONDITION. THE RESULT IS THE DOS CANNOT WRITE ON A DRIVE EVEN THOUGH THE WRITE PROTECT WAS NOT DELIBERATELY ENGAGED. WITH A BURNED-OUT LAMP, THE DRIVE IS CONTINUOUSLY WRITE PROTECTED! THE LAMP, BY THE WAY, IS RATED 5 VOLTS, 55 MILLIAMP. THE CURRENT DRAW IS NOT CRITICAL, ANYTHING FROM ONE MILLIAMP TO 1ØØ MILLIAMP WILL WORK. THE PROBLEM IS AVAILABILITY OF THIS SPECIAL LAMP. AN EMERGENCY SOLUTION IS TO REPLACE THE BURNED OUT LAMP WITH THE ONE FROM THE GREEN ACTIVE SOCKET. THE LAMPS ARE THE SAME, AND THE SYSTEM DOESN'T CARE IF THE GREEN LAMP WORKS OR NOT.

I HAVE CHASED OUT THE ORIGINAL WIRING OF ALL THE RADIO SHACK DRIVES, EITHER PERSONALLY OR THROUGH OTHER HELPFUL PERSONS. I ESPECIALLY WANT TO THANK ART MCANINCH OF BORGER, TX AND FRED OBERDING OF SAUSALITO, CA FOR THEIR KIND EFFORTS AND COMMUNICATIONS.

Table 2

"Three Wires" Connections

Early Masters with large HDC's, 8X300 chip type

Wire	Orange	Yellow	White
Function	Write Protect	Drive Select	Seek Complete
ORIGINAL CONNECTIONS			
5/12/15 Meg	TP-18	U21-8	U7-5
GENERIC (SIMPLIFIED) CONNECTIONS			
All MFM Drives	HDC resistor pack R27 pin 6 Note 1	HDC J6 pin 26 Note 3	HDC J6 pin 6 Note 3

Late Masters with small HDC's, WD1010 chip type

Wire	Orange	Yellow	White
Function	Write Protect	Drive Select	Seek Complete
ORIGINAL CONNECTIONS			
12/15 Meg	TP-18	U21-8	TP-8
35 Meg	J2-5	U1-5	U15-16
70 Meg	Pad R97 Note 4	Outboard pin, J11	TP-1
GENERIC (SIMPLIFIED) CONNECTIONS			
All MFM Drives	HDC R54, end nearest J10 Note 2	HDC R23, end nearest U23	HDC U31 pin 8

TABLE 2 SHOWS THE ORIGINAL HOOK-UP PLUS AN ALTERNATE (AND SIMPLER) GENERIC ARRANGEMENT FOR MASTER DRIVES, ESPECIALLY WHERE SOME OTHER BUBBLE IS BEING INSTALLED. SEE THE SECTIONS HEADED 'GENERIC' FOR THE SIMPLER CONNECTIONS.
THE ADVANTAGE OF THE GENERIC CONNECTION IS THAT YOU CAN THEREAFTER REMOVE OR EXCHANGE DRIVES WITHOUT HAVING TO UNSOLDER THE THREE WIRES AS THEY ARE NOW ATTACHED TO THE HDC INSTEAD OF THE BUBBLE.

NOTES FOR TABLE 2

1. SOLDER WIRE TO FEED THROUGH NEAR PIN 6. CUT TRACE TO PIN 5 OF J1 TO PREVENT FEEDING 5V INTO THE 20/C CABLE.
2. CUT TRACE TO PIN 5 OF J6 TO PREVENT FEEDING 5V INTO THE 20/C CABLE.
3. USE EMPTY PIN HOLE OF J6 INSTEAD OF J5.
4. THE PAD IS MARKED R97, BUT NO RESISTOR IS ACTUALLY INSTALLED THERE,

ALL OF THE DRIVES I DELIVER TO CUSTOMERS ARE CONNECTED IN THE GENERIC FASHION DESCRIBED ABOVE. THIS SIMPLIFIES MATTERS FOR BOTH ME AND THE CUSTOMER, AND OF COURSE THE LOGIC WORKS THE SAME AS ALWAYS, AS THE GENERIC CONNECTION CONNECTS THE THREE WIRES AT THE DESTINATION (THE HDC) INSTEAD OF AT THE SOURCE (THE BUBBLE).

OVER THE LAST COUPLE OF YEARS, I HAVE CONNECTED MANY DIFFERENT DRIVES INTO OUR RADIO SHACK BOXES, UP TO AND INCLUDING THE 70 MEG DRIVE, AND ALL WORK WELL. BE AWARE THAT THE 12 MEG, 35 MEG AND 70 MEG DRIVES ALL SUFFER FROM A BIT OF PUFFERY. THEY WERE ADVERTISED AND SOLD BY RADIO SHACK WITH THOSE LABELS, BUT THE LABELS ARE ONLY STRICTLY TRUE WHEN INSTALLED ON MACHINES OF THE MODEL 11 FAMILY BECAUSE THOSE MACHINES FORMATTED MORE BYTES PER TRACK THAN LS-DOS AND LDOS CAN DO. THE REAL LIMITS ON THOSE DRIVES WHEN USED IN A MODEL 1, 111 OR 4 SYSTEM ARE 11,304,960 BYTES, 33,554,432 BYTES AND 67,108,864 BYTES RESPECTIVELY, BASED UPON 8 HEADS AND 230, 512 OR 1024 CYLINDERS.
FOLLOWING THE USUAL ADVERTISING PRACTICE, YOU COULD MORE PROPERLY CALL THESE 11, 34 AND 67 MEG DRIVES. IT IS NOT POSSIBLE TO USE ANY DRIVE LARGER THAN 67/70 MEGS BECAUSE THE HDC CANNOT DEAL WITH MORE THAN 8 HEADS AND 1024 CYLINDERS, AND THE DOS CANNOT HANDLE OTHER THAN 32 SECTORS OF 256 BYTES EACH. NO MATTER, THE 67 MEG IS A LARGE DRIVE, AND YOU ARE UNLIKELY TO FILL IT.

SLAVE DRIVES

I HAVE OMITTED DISCUSSING SLAVE DRIVES IN THIS ARTICLE; THE WHOLE POINT OF SWAPPING DRIVES IS TO GET MORE CAPACITY, AND MY FEELING IS THAT YOU ARE BETTER OFF TO PUT A BIG DRIVE IN THE MASTER AND FILE THE SLAVE IN THE CLOSET. YOU CAN GO AS LARGE AS 70 MEG IN YOUR MASTER, SO WHY FOOL AROUND WITH THE SLAVE DRIVE?

JUST IN PASSING, I WILL REMARK THAT THE ''THREE WIRES'' ARE ACTUALLY FOUR IN NUMBER IN A SLAVE BOX,

THE FOURTH WIRE BRINGING 12 VOLTS TO A POWER RELAY (WHICH OBVIATES THE NEED FOR A POWER SWITCH IN A SLAVE), AND THE WIRE COLOURS DIFFER FROM THOSE IN A MASTER BOX.

MISCELLANEOUS COMMENTS

A FEW OTHER FACTORS OF INTEREST ARE THE SECTOR INTERLEAVE, THE CYLINDER TO REDUCE WRITE CURRENT, THE CYLINDER TO BEGIN PRE-COMPENSATION, AND THE AVERAGE ACCESS TIME. THE SECTOR INTERLEAVE IS PREDETERMINED WITHIN THE FORMATTER, AND UNLESS YOU ARE GOOD AT MACHINE CODE, IS NOT ADJUSTABLE. THE GURUS WHO DESIGNED THE RADIO SHACK HARD DRIVE SYSTEM SET THIS FOR US. I HAVE NOT MADE ANY ATTEMPT TO 'TUNE' THE INTERLEAVE, HAVING NOT HAD THE TIME (NOR THE INTEREST) TO TACKLE THIS AREA.

THE CYLINDER AT WHICH WRITE CURRENT SHOULD BE REDUCED IS SPECIFICALLY STATED BY THE DRIVE MANUFACTURERS, BUT OUR FORMATTERS SIMPLY ASSUME A VALUE AND GO AHEAD WITHOUT ASKING US. IF YOU REVIEW THE DRIVE SPECS, THE RECOMMENDED VALUE IS TYPICALLY ABOUT HALF THE TOTAL NUMBER OF CYLINDERS, AND I BELIEVE THIS IS WHAT THE FORMATTER PROGRAMS ASSUME. FURTHERMORE, THE NEWER DRIVES TAKE CARE OF THIS FUNCTION IN HARDWARE, AND SO REGARDLESS OF WHAT THE SOFTWARE AND CONTROLLER SAY, THE LATER DRIVES DO THEIR OWN THING. DON'T WORRY ABOUT IT, IT IS NOT CRITICAL,

THE CYLINDER TO BEGIN WRITE PRE-COMPENSATION VARIES CONSIDERABLY. MOST OLDER DRIVES SET IT AT ABOUT 1/2 OF THE TOTAL CYLINDERS, BUT SOME OF THE LATER ONES DON'T WANT ANY PRE-COMPENSATION. AGAIN, THIS VALUE APPEARS NOT TO BE CRITICAL, AND IS TAKEN CARE OF IN THE DRIVER SOFTWARE. I HAVE RECEIVED ONE (ONLY) REPORT FROM A USER WHO SAID HE HAD TO PATCH HIS CODE TO SUPPRESS PRE-COMPENSATION IN ORDER TO MAKE HIS DRIVE WORK, BUT I LACK ANY FURTHER DETAILS. IT SEEMS TO BE A FACTOR OF CONCERN, GENERALLY.

FINALLY, THE AVERAGE ACCESS TIME IS A BRAGGING POINT AMONG IBM USERS, BUT FOR US TRS TYPES, HARD DRIVES ACCESS SO MUCH QUICKER THAN FLOPPIES THAT THE WHOLE PROCESS SEEMS LIKE MAGIC ANYWAY. FOR THE DRIVES WE ARE USING, THE ACCESS TIMES ARE AS SHOWN IN TABLE 1 , AND (EXCEPT FOR THE 70 MEG DRIVE), ARE SLOW BY IBM STANDARDS. EVEN SO, THE CHANGE FROM FLOPPIES IS DRAMATIC, AND YOU WILL ENJOY RADIO SHACK HARD DRIVES ANYWAY.

MOST DRIVES HAVE BUFFERED SEEK, WHICH ALLOWS THE CONTROLLER TO SEND STEPPING INSTRUCTIONS AT THE 10 MICROSECOND RATE. THE DRIVE ACTUALLY JUST HOLDS THE COUNT IN AN INTERNAL REGISTER UNTIL THE CONTROLLER STOPS SENDING. THE DRIVE THEN MOVES ITS HEADS AT ITS OWN BEST RATE, AND STOPS ON THE DESIRED CYLINDER. A FEW VERY OLD (MOSTLY 5 MEG) DRIVES LACKED THIS FEATURE, AND WITH THESE YOU MUST TELL THE CONTROLLER TO SEND THE STEPPING PULSES AT 3 MILLISECOND OR GREATER INTERVALS. ALWAYS TRY THE 10 MICROSECOND STEP RATE FIRST, AND IF THE FORMATTER CHOKES, (REPORTS MOST

CYLINDERS BAD) THEN TRY SLOWER STEPPING RATES UNTIL YOU FIND ONE THAT WORKS.

HERE IS A WARNING TO NEWDOS-80 V2.5 USERS, THE HARD DRIVE VERSION OF NEWDOS-80 IS V2.5, AND IT WORKS FINE ON THE OLD, LARGE HDC BOARDS. HOWEVER, THERE IS A BUG IN NEWDOS' FORMATTER WHICH PREVENTS OPERATION ON THE NEW, SMALLER HDC'S. FORTUNATELY, THIS BUG WAS SQUASHED BY AN AUSSIE, AND A FRIEND OF HIS SENT THE CURE ALONG TO ME.

THE PATCH TO HDFMTAPP, THE NEWDOS FORMATTER IS AS FOLLOWS. USING SUPERZAP:

```
DFS OF FILE HDFMTAPP/CMD
FRS1
MOD D1
FIND:       AF 32 CB 00
CHANGE TO: 3E 0F D3 CB
```

ALL OF THE DRIVES IN TABLE 1 ARE PHYSICALLY AND ELECTRICALLY INTERCHANGEABLE. THERE ARE ACTUALLY MANY MORE FLOATING AROUND; THEY JUST HAVE TO BE MFM TYPES. THE MISOSYS RSHARD5/6 DRIVERS AND POWERSOFT SUPREME HD DRIVERS (SERIES RS) BOTH WILL WORK WITH ALL OF THEM. THE MISOSYS DRIVERS ARE EASIER TO INSTALL, BUT THE POWERBOAT SUPREME DRIVERS ALLOW GREATER FLEXIBILITY WHEN YOU WANT MAXIMUM CONTROL OVER PLACEMENT OF PARTITIONS.

(NOTE: THE DRIVER, RSHARD, THAT ROY REFERS TO, IS CURRENTLY AVAILABLE FOR DOWNLOADING AT WWW.TIM-MANN.ORG. A BIG THANKS GO TO TIM MANN & ROY SOLTOFF FOR THIS FACILITY.
IT IS STILL POSSIBLE TO BUY MFM DRIVES. I'VE SEEN THEM ADVERTISED ON EBAY ON A NUMBER OF OCCASIONS.
ED.)

--== o0o ==--

PACKING AND SUPER-GRAPHICS

BY PHILLIP CASE

BY NOW, MOST OF YOU HAVE SEEN THOSE FUNNY LISTINGS WHICH APPEARS TO BE GARBAGE. PROGRAMS LIKE 'ANDROID NIM' 'BEE WARY' AND 'STAR SCOUT' ALL LOOK LIKE BAD LOADS WHEN LISTED. THIS IS DUE THE USE OF A PROCESS KNOWN AS 'PACKING' THE GRAPHICS.

THE REASON THESE PROGRAMS HAVE THEIR GRAPHICS STRINGS PACKED THIS WAY IS TO SAVE MEMORY. BY PACKING YOUR STRINGS IN THIS MEMORY, YOU REDUCE THE AMOUNT OF OVERHEADS MEMORY NEEDED BY ABOUT TWO-THIRDS.

TO CREATE SUPER-GRAPHICS, ONE SIMPLY CHANGES THE VALUE OF THE CHARACTERS BETWEEN THE QUOTES IN A LINE TO READ AS GRAPHICS CODES. FOR EXAMPLE: 10 A$ = "*". TO CHANGE THIS LINE, ONE WOULD SIMPLY POKE THE MEMORY LOCATION THAT CONTAINS THE "*" WITH THE VALUE OF THE GRAPHICS CHARACTER NEEDED. IN THIS CASE LET'S USE A FULL GRAPHICS BLOCK OR CHR$(191). TO CHANGE LINE 10 TO PRINT A CHR$(191), WE FIND THE ADDRESS WHICH CONTAINS THE "*" AND POKE THE ADDRESS WITH 191.

MOST OF YOU ARE PROBABLY ALREADY FAMILIAR WITH THIS PROCESS, SO I WON'T SAY ANYTHING MORE ABOUT IT EXCEPT THAT DISK USERS HAVE

A REAL ADVANTAGE IN THE USE OF DEBUG.

THE INTERESTING THING ABOUT SUPER-GRAPHICS IS THE METHOD IN WHICH THEY WORK. THE S-80 CONVERTS ALL COMMANDS INTO ONE BYTE TOKENS TO SAVE MEMORY. IT JUST HAPPENS THAT THE COMMAND TOKENS ARE THE SAME ASCII VALUES AS THE GRAPHICS CHARACTERS. THAT'S WHY A SUPER-GRAPHICS LINE CONTAINS ONLY COMMAND WORDS.

NOW FOR THE HEAVY STUFF, STAY CLOSE. WE'VE ALL BEEN TOLD THAT SUPER-GRAPHICS LINES CANNOT BE EDITED. THIS IS BECAUSE THE COMPUTER READS ALL THE CONTENTS BETWEEN QUOTES AS REGULAR CHARACTER INFORMATION RATHER THAN TOKENS. IF YOU'RE LIKE ME, YOU FIND YOU NEED TO EDIT THE LINE THAT YOU'VE SLAVED SO HARD OVER TO CONVERT TO TOKENS.

THERE IS A LITTLE TRICK THAT WILL PERMIT YOU TO EDIT YOUR SUPER-GRAPHICS WITHOUT LOSING YOUR TOKENS. IF, WHILE YOU'RE EDITING THE LINE, YOU (C)HANGE THE FIRST QUOTE IN THE LINE TO AN ASTERISK, THE LINE WILL BE RETOKENIZED. THEN, WHEN DONE EDITING, POKE THE ASTERISK BACK TO A QUOTE AND VOILA! THE LINE IS CONVERTED BACK TO SUPER-GRAPHICS AFTER EDITING.

THIS LITTLE TRICK SHOULD PROVE USEFUL WHENEVER YOU FIND YOU NEED MORE CHARACTERS IN A LINE WHICH IS ALREADY PACKED. FOR THOSE OF YOU WITH A DISK SYSTEM, I SUGGEST DOING ALL THE AFOREMENTIONED WITH DEBUG, THE MONITOR WHICH IS A PART OF DOS.

--== oØo ==--

MORE MILEAGE FROM EPSON RIBBONS

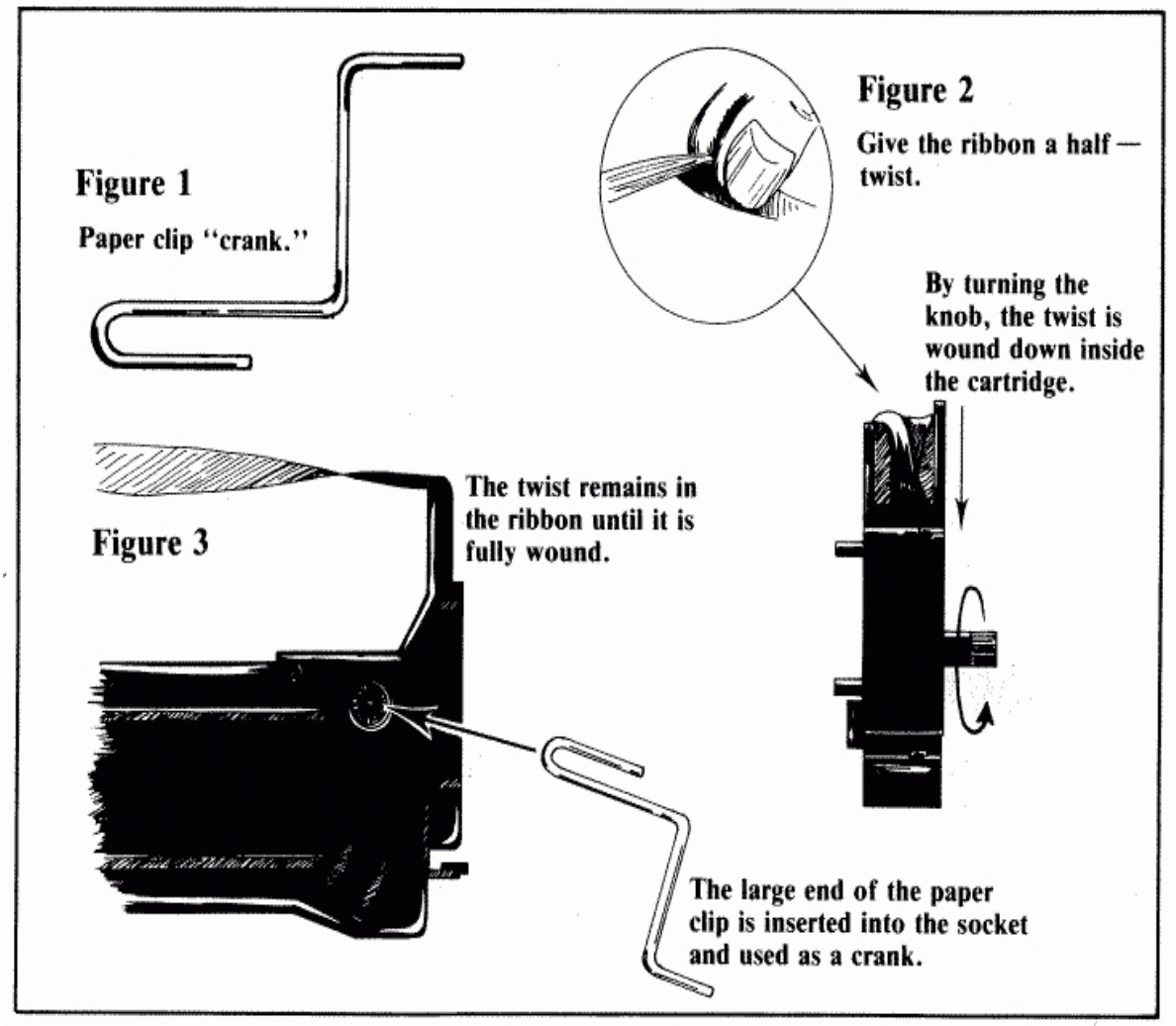
IT IS POSSIBLE TO EXTEND THE LIFE OF EPSON MX-70 AND MX-80 RIBBON CARTRIDGES BY FLIPPING THE RIBBON OVER IN THE CARTRIDGE. SINCE THE PRINT HEAD STRIKES OFF-CENTRE ON THE RIBBON, FLIPPING THE RIBBON EXPOSES A FRESH SURFACE. HERE'S A SIMPLE TECHNIQUE FOR DOING JUST THAT.

TAKE AN ORDINARY PAPER CLIP AND BEND IT AS SHOWN IN FIGURE 1. LAY THIS ASIDE, WITHIN EASY REACH. NOW TAKE THE RIBBON CARTRIDGE AND REFER TO FIGURE 2. LIFT THE RIBBON AT THE END NEAR THE RIBBON ADVANCE KNOB AND GIVE THE RIBBON HALF A TWIST. TURN THE ADVANCE KNOB A FEW TURNS TO FEED THE TWISTED RIBBON INTO THE CARTRIDGE THROUGH THE ADVANCE MECHANISM. AT THIS POINT, THE RIBBON CARTRIDGE SHOULD LOOK AS IT DID BEFORE, EXCEPT FOR A HALF-TWIST IN THE EXPOSED PORTION.

TAKE THE PAPER CLIP CRANK YOU MADE EARLIER AND INSERT THE LOOP END INTO THE RIBBON ADVANCE SOCKET ON THE CARTRIDGE. THIS IS ON THE BOTTOM OF THE CARTRIDGE, OPPOSITE THE ADVANCE KNOB AS IN FIGURE 3. NOW USE THE CRANK TO ADVANCE THE RIBBON UNTIL THE HALF-TWIST INSIDE THE CARTRIDGE COMES OUT THE OTHER END. THAT'S ALL THERE IS TO IT. THE RIBBON IS NOW FLIPPED OVER.

IT'S THE PAPER CLIP CRANK THAT MAKES THIS IDEA PRACTICAL. WITHOUT THIS, YOU WOULD HAVE TO ADVANCE THE RIBBON WITH THE KNOB. SINCE THERE ARE 20 YARDS OF RIBBON

IN THE CARTRIDGE, THIS COULD TAKE QUITE SOME TIME AND THE SHARP/HARD LITTLE KNOB WOULD PLAY HAVOC WITH YOUR FINGER(S)! WITH THE CRANK HOWEVER, IT TAKES NO MORE THAN 2 OR 3 MINUTES TO WIND THE ENTIRE RIBBON.



---== o0o

(CONTINUED FROM PAGE 2.)

I'VE MANAGED TO DO A DEAL WITH 'FABSITESUK.COM' WHO ALLOW WEB HOSTING FOR VERY MODERATE FEES IF YOU'RE A NON-PROFIT MAKING ORGANIZATION.

TALKING OF NEWS-LETTER CONTRIBUTION, THEY ARE A BIT THIN ON THE GROUND. IF YOU'RE HAVING FUN WITH YOUR TRS-80, I'D LOVE TO KNOW WHAT YOU'RE UP TO. JUST AN UPDATE OF A FEW LINES WOULD BE APPRECIATED.

MY EMAIL ADDRESS ON THE WEB-SITE HAS BEEN CHANGED FROM THE DATE OF THIS ISSUE, SO, IN CASE YOU LOOSE THIS NOTE OF IT, JUST CLICK ON THE EMAIL BUTTON.

WELL, THAT ABOUT WRAPS UP THIS EDITION. I HOPE YOU FOUND SOMETHING OF INTEREST; SO UNTIL SEPTEMBER

BYE FOR NOW

DUSTY.



TRS8BIT



WELCOME TO THE SEPTEMBER EDITION OF TRS8BIT. JUST A QUICK THANK YOU FOR THE FEEDBACK, KIND COMMENTS IN GENERAL AND IN PARTICULAR ABOUT ROY'S HARD DISK ARTICLE IN THE LAST ISSUE. IT SEEMS AT LEAST TWO PEOPLE HAVE BEEN ASSISTED BY IT.
HOW ABOUT LETTING US ALL KNOW HOW YOU'RE GETTING ON?

CONTRIBUTIONS HAVE BEEN A BIT THIN ON THE GROUND, MOST PROBABLY DUE TO THE AUGUST HOLIDAY TIME, BUT I HOPE THAT I'VE MANAGED TO FIND ODD LITTLE BITS OF INTEREST FOR YOU!

THE USUAL ONE-LINER MAKES AN APPEARANCE AND A NUMBER OF SNIPPETS IN 'AT THE READY PROMPT' WHICH I'VE BEEN PLAYING AROUND WITH.
I'VE PRODUCED A CHART OF 'LOOK AT A GLANCE' PRINTER INFORMATION TO ASSIST IN CHECKING FOR COMPARABLE OPTIONS.
THERE'S AN ARTICLE BY ANON. WHO DOESN'T LIKE 'MOD'. PERHAPS, IN HIS YOUNGER LIFE, HE WAS A ROCKER EH?

STAR-BILLING GOES TO A CRACKING ARTICLE BY EC KILPATRICK FOR ALL YOU HARDWARE BUFFS OR ANYONE WANTING TO FIT 48K INSIDE THEIR KEYBOARD.

THIS ALL CAME ABOUT BECAUSE I 'WON' SOME 8 BIT, 64K RAM CHIPS WHICH WERE FOR SALE ON EBAY AND I WAS SURE THAT, IN THE BACK OF MY MIND, THERE WAS AN ARTICLE WHICH HAD A USE THEM. SURE ENOUGH, I FOUND IT.
IT WAS ORIGINALLY PUBLISHED IN NATGUG NEWS IN SEPTEMBER 1984 AND I'VE MANAGED TO INCORPORATE ALL THE LATER, ADDITIONAL AMENDMENTS THAT WENT WITH IT.
THE CHIPS ARE MARKED-
SHARP LH2164-15
** NOW IF ANYONE WOULD LIKE TO TRY IT OUT I'D BE MORE THAN HAPPY TO SEND THE RAM CHIPS FOR YOU TO USE **

IF YOU'RE INTERESTED, PLEASE EMAIL ME WITH YOUR DETAILS AND I'LL SEND THE CHIPS OFF TO YOU.
THIS LOOKS TO ME LIKE MAJOR SURGERY AND IS WAY PAST MY MEAGRE DIY ABILITIES.
I CAN'T EVEN GUARANTEE THAT THE CHIPS ARE SUITABLE OR EVEN STILL USABLE BUT IT WOULD BE REAL FUN TO TRY IT OUT. JUST THINK OF IT, HARDWARE MODS ON A 30 YEAR OLD MICRO-COMPUTER!
I DON'T RECOMMEND ATTACKING YOUR ONE AND ONLY REMAINING MODEL 1 THOUGH!

N.B. THIS OFFER IS, BY ITS VERY NATURE, ON A FIRST-COME FIRST-SERVED BASIS! THERE WOULD BE NO CHARGE, BUT A SHORT ARTICLE FOR TRS8BIT TELLING US HOW WELL (OR OTHERWISE) THE MODS WENT, WOULD BE MOST APPRECIATED.

ALSO, WHILE I'M THINKING ABOUT FREEBIES, I'VE, ONCE AGAIN THANKS TO EBAY, ACQUIRED A NUMBER OF 'NEW-OLD-STOCK' RIBBONS FOR AN EPSON LX80. IF YOU'RE HAVING TROUBLE SOURCING A REPLACEMENT JUST LET ME KNOW. THE REFERENCE NUMBERS QUOTED ARE 8MM X 6M F3570ZA.
AND...
WHILE HELPING ON A CLEARANCE, I CAME ACROSS A

PART-BOX OF NEW, 8" FLOPPY DISKS. SO IT YOU RUN A MODEL 2 OR ANY MACHINES WITH 8" DRIVES AND YOU'RE HAVING DIFFICULTIES GETTING MEDIA, AGAIN, PLEASE LET ME KNOW. THEY ARE MEMOREX, SINGLE SIDED, DOUBLE DENSITY, SOFT SECTORED WITH 77 TRACKS.

TALKING OF MODEL 2'S, ONE SOLD ON THE UK'S EBAY FOR £75. THERE HAS BEEN QUITE A FEW TANDYS SOLD OVER THE LAST FEW WEEKS, WITH MODEL 1'S FETCHING BETWEEN £18 AND £35 AND MODEL 4'S FETCHING BETWEEN £50 AND £90. THERE'S BEEN CONSIDERABLY MORE SALES OF ACCESSORIES, BOOKS AND SOFTWARE TOO, FOR ALL MODELS.
AS AT THE TIME OF WRITING, THERE IS A VIDEO GENIE FOR SALE IN AN EBAY 'SHOP'. THE PRICE IS £80
IT'S NICE TO KNOW THERE'S STILL A BIT OF INTEREST OUT THERE.

CONTINUED ON PAGE 7

AT THE
READY>
PROMPT



ONE REALLY NEAT FEATURE OF DISK BASIC IS THAT IT ALLOWS THE USE OF HEX AND OCTAL CONSTANTS.
I.E. IF YOU ISSUE THE COMMAND PRINT &H5BBB, IT RETURNS THE ANSWER 23483, THE DECIMAL EQUIVALENT. THIS SAVES ME HAVING TO GET OUT MY TANDY PC-6 COMPUTER ANY TIME I NEED A CONVERSION FIGURE!

HERE'S A FEW OLD CHESTNUTS, BUT JUST IN CASE YOU DON'T HAVE THEM TO HAND;
TO DISABLE THE MODEL 1'S BREAK KEY POKE 16396,23
AND, JUST AS HANDY, TO ENABLE THE BREAK KEY POKE 16396,201.

IF YOU CAN'T REMEMBER WHAT YOU'VE SET MEMORY SIZE TO WHEN STARTING LEVEL II BASIC, HERE'S AN EASY WAY TO HELP.
PRINTPEEK(16561)+PEEK (16562)*256+2.

IF PEEK(293) = 73, THE MACHINE YOU'RE ON IS A MODEL 3. ANY OTHER VALUE AND IT'S A MODEL 1.

THESE WORK FINE WITH MATTHEW REED'S M1 EMULATOR

IF YOU'RE ON A MODEL 1 -
POKE 15360,1
PRINT PEEK(15360) THIS WILL PRINT A 1 IF THE R/S LOWER-CASE MOD HAD BEEN FITTED. IF NOT, THE RESULT WILL BE 65

THERE'S A LITTLE 'BUGLETT' IN SOME EDITIONS OF TRS-DOS 2.3 WHICH I HAD FORGOTTEN ABOUT. THE PASSWORD PROTECTION GIVES UP AFTER ONE TRY. THE FIRST TIME YOU TRY TO COPY AN UNCOPYABLE FILE IT WILL GIVE THE ERROR MESSAGE "FILE ACCESS DENIED". TRY TO COPY IT AGAIN - THIS TIME YOU SHOULD SUCCEED!

--== o0o ==--

HERE'S A ONE-LINER FROM AN ORIGINAL IDEA BY MICHAEL LYON

```
0 IFW>0GOTOELSECLS:D=400:B=20:L=B:S=1:E=.4:C=.05:F=.5:FORW=1TO52
:FORX=-LTOLSTEPS:Y=SQR(ABS(D-X*X)):IFD<X↑2 Y=F:SET(Y*S+B,X+20):B
=B+C:D=D-E:NEXT:L=-L:S=-S:NEXT:ELSESET(Y*S+B,X+20):B=B+C:D=D-E:N
EXT:L=-L:S=-S:NEXT:GOTO
```

48K IN THE MODEL 1 KEYBOARD

EC KILPATRICK

NOW THAT GENERAL NORTHERN HAVE GONE OUT OF BUSINESS, WE THOUGHT IT WOULD BE A GOOD IDEA TO SEE IF IT WAS POSSIBLE TO MODIFY THE MODEL 1 KEYBOARD TO READ 48K WITHOUT HAVING TO ADD A DECODER BOARD.

WITH CONSIDERABLE HELP FROM MY GOOD FRIEND GUY GRANTHAM, WHO HAD ALREADY MODIFIED HIS VIDEO GENIE, WE HAVE SUCCEEDED. HERE IS HOW WE DID IT.

THE MEMORY SIGNALS COMBINED WITH RAS* AND CAS* TO READ:-
UP TO 16K RAM REQUIRE A15 LOW AND A14 HIGH
FROM 16K TO 32K REQUIRE A15 HIGH AND A14 LOW
FROM 32K TO 48K REQUIRE A15 HIGH AND A14 HIGH
THEREFORE, TO READ ALL 48K EITHER A14 OR A15 OR BOTH MUST BE HIGH AT THE RELEVANT TIME.

ORIGINALLY THE COMPUTER COULD ONLY RECOGNISE UP TO 16K OF RAM BECAUSE IT WAS ONLY DECODED TO READ MEMORY WHILE A15 WAS LOW, DUE TO Z73 (PINS 4,5,6 OR GATE) CONTROLLING Z21 (2 LINE TO 4 LINE DECODER) AT PINS 14 AND 2.

THESE SIGNALS WILL BE RETAINED FOR VIDEO, KEYBOARD AND ROM, BUT NO LONGER USED TO GENERATE THE MEM* AND RAM* SIGNALS BY OPENING LINKS 2,3,4,5 ON X3.

NEW CONNECTIONS ARE REQUIRED TO GENERATE THE MEM* SIGNAL WHICH COMES FROM PIN 6, Z74.

THIS SIGNAL NOW HAS TO BE ACTIVE LOW WHENEVER THERE IS A RD* ACTIVE, EXCLUDING CALLS TO RESERVED MEMORY 3000H TO 37FFH WHICH IS REQUIRED FOR DISK AND PRINTER I/O, OR TO THE KEYBOARD AT 3800H TO 3BFFH, OR AGAIN TO THE SEPARATE VIDEO RAM AT 3C00H TO 3FFFH. ALL THESE INVALIDATING CONDITIONS ARE MET WHEN A15 AND A14 ARE LOW AND A13 AND A12 ARE HIGH, AS PROVIDED BY THE OUTPUT FROM Z21 PIN 12 TO INPUT PIN 4 OF Z36. SO BY CUTTING THE TRACE TO PIN 5 OF Z74 AND LINKING PIN 4 OF Z36 TO THIS PIN, THE REQUIRED CONDITIONS OF MEM* ARE MET FOR 48K RAM.

Z73 PINS 8,9,10 ARE NOW SPARE BY VIRTUE OF CUTTING THE TRACE TO Z74 PIN 5. THIS GATE, TOGETHER WITH 2 OTHER ORIGINALLY 'SPARE' GATES, Z73 PINS 11,12,13 AND Z37 PINS 8,9,10 WILL BE USED TO DECODE THE ROW AND COLUMN ADDRESSES OF THE 64K RAM CHIPS.
PINS 9 AND 10 OF Z73 MUST HAVE THEIR INCOMING TRACES CUT AND REPLACED BY LINKS TO A14 AND A15 AT Z21 PIN 15 AND Z73 PIN 4 RESPECTIVELY.

Z73 PIN 8 OUTPUTS THE (A15 OR A14) CONDITION REQUIRED IN THE FIRST PARAGRAPH TO ADDRESS THE TOP 48K OF THE MEMORY MAP.
THIS SIGNAL IS INVERTED BY THE SPARE NOR GATE OF Z37 (8,9,10) AND COMBINED WITH RAS* IN THE SPARE OR GATE OF Z73 (12,13) WHICH, IN TURN FROM PIN 11 FEEDS THE EXISTING RAM* CONNECTION AT Z74 PIN 10.
NOTE :- PINS 8 AND 9 OF Z37 MUST BE SEPARATED AND ONLY ONE OF THEM MAY BE CONNECTED TO PIN 8 OF Z73, AS ALTHOUGH

PIN 9 APPEARS TO BE 'FLOATING' IT IS NOT POSSIBLE TO BE 100% CERTAIN ABOUT PIN 8, SO IT MUST BE TIED TO PIN 7 (EARTH).
A14 AND A15 MUST ALSO BE LINKED TO PINS 14 AND 13 OF THE DATA SELECTOR Z51, WHILE IT'S OUTPUT AT PIN 12 WILL BE LINKED TO THE NUMBER 9 PINS OF EACH OF THE 64K RAMS.

HERE ARE THE DETAILED CHANGES NECESSARY :-

OPEN UP THE KEYBOARD AND LOCATE THE FOLLOWING CHIPS ON THE 'TRACE' SIDE OF THE BOARD. Z21, Z36, Z37, Z51, Z73 AND Z74. (I STUCK A SMALL NUMBER LABEL ON EACH CHIP TO MAKE SURE I COULD ALWAYS READ THEM AND LOCATE PIN 1).

LAY THE KEYBOARD COMPONENT SIDE DOWN WITH THE KEYS AWAY FROM YOU.

CUT THE FOLLOWING TRACES, BETWEEN PINS 8 AND 9 OF Z37 (A SOLDER SUCKER IS NEED HERE), CLOSE TO PINS 9 AND 10 OF Z73, (PIN 10 ON THE COMPONENT SIDE) AND PIN 5 OF Z74.

***
NOTE:-
SOME LATE ISSUE BOARDS ALSO HAVE THE TRACES OF PINS 12, 13 AND 14 OF Z73 JOINED TOGETHER SO THE TRACE BOTH SIDES OF PIN 13 MUST ALSO BE CUT.
***

USING THIN INSULATED WIRE CONNECT THESE PINS IN THIS ORDER :-
FROM Z73 PIN 11 TO Z74 PIN 10
FROM Z73 PIN 12 TO Z73 PIN 5
FROM Z73 PIN 10 TO Z73 PIN 4 AND ON TO Z51 PIN 13
FROM Z73 PIN 13 TO Z37 PIN 10
FROM Z73 PIN 9 TO Z51 PIN 14 AND ON TO Z21 PIN 15
FROM Z73 PIN 8 TO Z37 PIN 9
FROM Z74 PIN 5 TO Z36 PIN 4
FROM Z37 PIN 8 TO Z37 PIN 7

NOW X3 HAS TO BE MODIFIED, AS MENTIONED EARLIER, STARTING FROM PIN 1, LEAVE THAT AS IS. PINS 2,3,4 AND 5 HAVE TO BE OPEN AND PINS 6,7 AND 8 CLOSED. A SMALL SCREWDRIVER CAN BE USED TO OPEN AND A BLOB OF SOLDER TO CLOSE AS NEEDED.

BEFORE THE 16K RAM CHIPS ARE REMOVED, THE SYSTEM CAN BE TESTED BY SWITCHING THE COMPUTER ON IN LEVEL 2 MODE AND ENTERING 'PRINT MEM', THE ANSWER SHOULD BE EITHER 48340 OR 48338 DEPENDING ON WHICH TYPE OF ROMS ARE FITTED. THE FIGURES ARE ONLY 2 REFLECTIONS OF THE EXISTING 16K AND AS SUCH, CANNOT BE USED.

PROVIDING THE SYSTEM WORKS THIS FAR, THEN FITTING THE 64K RAMS CAN BE CARRIED OUT. REMOVE THE 16K RAMS AND PUT THEM SAFELY AWAY. WITH THE KEYBOARD STILL LAID OUT WITH THE KEYS AWAY FROM YOU, CUT THE 12V TRACE AT PIN 8 OF Z19. CUT THE -5V TRACE AT THE CAPACITOR PIN BELOW PIN 1 OF Z13. FIT A THICK WIRE FROM THE +5V PIN ABOVE PIN 7 OF Z1 TO THE PIN ABOVE PIN 8 OF Z17.
FOLD THE KEYS OVER ON TOP OF THE BOARD AND TURN THE WHOLE ASSEMBLY OVER. VERY CAREFULLY BEND UP PIN 9 OF EACH OF THE 64K RAM CHIPS (NOT TOO FAR OR THEY MAY BREAK) AND INSERT THEM IN TO THE EMPTY RAM SOCKETS THE SAME WAY ROUND AS THE 16K CHIPS, MAKING SURE THAT EACH PIN 9 IS CLEAR OF THE TOP OF THE SOCKET, AS THAT MIGHT BE AT +5V. (CONTACTS MAY BE

SLIGHTLY PROUD ON SOME TYPES OF SOCKETS). WIRE ALL THE NUMBER 9 PINS TOGETHER AND CONNECT A WIRE FROM PIN 12 OF Z51 TO THE NEAREST PIN 9 OF THE RAMS.

CAREFULLY **RECHECK** THAT THE +12V TRACE IS DEFINITELY CUT AND THE SYSTEM IS READY TO GO.
IT IS NOT ESSENTIAL TO CUT THE -5V RAIL BUT IT SHOULD BE DONE AS A SAFETY MEASURE. CUTTING THE ORIGINAL +5V RAIL IS A VERY COMPLICATED JOB AND IS BEST LEFT ALONE, HENCE THE COMMENT ABOUT PINS 9 BEING WELL CLEAR OF THE SOCKET AS THAT IS STILL AT +5V.

***
NOTE:-
DO NOT USE TEXAS 64K RAMS AS THEY HAVE A DIFFERENT REFRESH SYSTEM AND CANNOT BE USED WITHOUT FURTHER MODIFICATIONS TO THE COMPUTER CIRCUITRY.
***

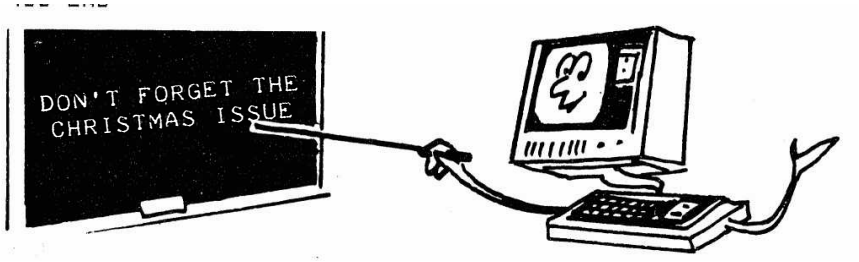
---= o0o =---

I DON'T LIKE MODS

ANON.

MOD IS AN ARITHMETIC FUNCTION FOUND ON THE MODEL II. AS WITH SWAP, THIS FUNCTION IS USEFUL, BUT IT CAN VERY EASILY BE DUPLICATED ON A COMPUTER WITHOUT THIS FUNCTION, SUCH AS THE MODEL I OR MODEL III.
MOD SIGNIFIES THE INTEGER REMAINDER WHEN A DIVISION OPERATION IS PERFORMED. AN ELEMENTARY SCHOOL STUDENT LEARNING DIVISION WOULD LEARN THAT 5 DIVIDED BY 3 IS ONE REMAINDER 2. THUS, 5 MOD 3 EQUALS 2. IN A SIMILAR FASHION, ONE COULD DERIVE THAT 103 MOD 25 EQUALS 3.
THE MOD FUNCTION ON THE MODEL 2 TAKES THE FORM A MOD B. ITS EQUIVALENT ON THE MODEL I OR MODEL 3 WOULD BE A -
INT (A/B) * B. THUS, THE EQUIVALENT OF 103 MOD 25, AS IN OUR PREVIOUS EXAMPLE, WOULD BE 103 - (INT (103/25) * 25), WHICH, AS STATED ABOVE, SIMPLIFIES TO 3.
MOD DOES NOT SAVE A SIGNIFICANT PORTION OF EXECUTION TIME. IT DOES MAKE CODE USING THIS FUNCTION A BIT EASIER TO READ, BUT THIS, I BELIEVE, IS FAR OUTWEIGHED BY THE INCOMPATIBILITY IT CAUSES. SINCE IT IS EXTREMELY EASY TO "CONSTRUCT" A MOD FUNCTION WHICH WILL WORK ON VIRTUALLY ANY VERSION OF BASIC, I WOULD STRONGLY ADVISE AGAINST USING THIS FUNCTION.

---= o0o =---



TRS8BIT "AT A GLANCE" TRS-80 PRINTER COMPARISON CHART (CIRA EARLY 1980'S)

	Centronics 737 LP II	Epson MX-80	Oki Microline 80	CGP-115	Quickprinter 2	DMP-100	Cannon BJ-10sx
WEIGHT (LBS)	12	12	14	1.8		8.6	3.7
SIZE	5 x 14.5 x 11	4.2 x 14.7 x 12	4.25 x 13.5 x 9.75	8.5 x 8.5 x 3	3.5 x 7 x 9.5	16 x 8.25 x 5.3	31 x 22 x 4.85
INTERFACE - PARALLEL/SERIAL	P	P	P	Both	Both	Both	P
SPEED	50 cps @ 10 cpi	80 cps @ 10 cpi	80 cps @ 10 cpi	12 cps	64 cps	50 cps @ 10 cpi	110 cps @ 10 cpi
BI-DIRECTIONAL	N	Y	N	N	N	N	Y
PRINT-HEAD LIFE - CHARS.	150 million	50 - 100 million	200 million	n/a	30 million	?	n/a
DESCENDERS	Y	Y	N	Y	N	N	Y
NO OF WIRES IN HEAD	9	9	9	n/a	n/a	7	n/a
DENSITIES CHARS/LINE	40, 66, 80, 132	40, 66, 80, 132	40, 80, 132	40, 80	9, 18 cpi	40,80	5, 8.5, 10, 12, 17
GRAPHICS	N	Y	Y	Y	Y	Y	Y
SLASH ZERO	N	N	Y	Y	Y	Y	N
LINE SPACING - LINES PER INCH	6	6 or 8	6 or 8	software selectable (fwd & rev)	6	6, 9	variable
MAX PAPER WIDTH	9.5 fanfold, 8.5 cut	10 fanfold	9.5 fanfold, 8.5 cut	4.5	2.38 ins - fixed	9.5 fanfold	8 ins
TRACTOR OR FRICTION FEED	F	T	Both	F	F	T	F
MAX COPIES	3	3	3	1	1	1	1
INK	ribbon, mobius loop	ribbon, cartridge	ribbon, .5 typewriter	4 colour pens	Aluminum coated	inked roller cassette	bubble jet cartridge

HIDE YOUR CODE!

BY PHILIP CASE

HERE'S A TIP ABOUT USING CONTROL CODE FUNCTIONS TO MAKE YOUR PROGRAMS HARDER FOR OTHER PEOPLE TO LIST, EXAMINE AND/OR AMEND. IF YOU REMEMBERS, CHR$(23) PUTS YOUR MODEL 1'S SCREEN INTO THE ENLARGED MODE (32 CHARACTERS PER LINE). CONSEQUENTLY, ANYTHING THAT'S ON THE SCREEN IN THE 62 CHARACTER MODE WILL LOSE EVERY OTHER LETTER WHEN YOU SWITCH INTO THE ENLARGED MODE. YOU CAN ILLUSTRATE THIS BY TYPING ANYTHING ON THE SCREEN, PRESSING SHIFT/ RIGHT ARROW, THEN PRINTING CHR$(23) (HOME CURSOR). BY DOING THIS YOU ARE ENTERING A CONTROL CHARACTER IN THE IMMEDIATE MODE.
YOU CAN PUT THESE CODES IN YOUR BASIC PROGRAM CODING BY ADDING A 'REM'ARK AT THE END OF AN IMPORTANT LINE, THEN AND ASTERISK (*).
HERE'S AN EXAMPLE -

```
10 FOR A=1TO10:NEXTA:REM*
```

THIS LINE WILL STILL FUNCTION IN THE REGULAR WAY. AFTER EDITING THE LINE, GO BACK AND FIND THE LOCATION IN MEMORY WHERE THE ASTERISK ACTUALLY RESIDES. ONE FOUND, POKE THE ADDRESS WITH 23. THEN LIST THE LINE. THE LINE AUTOMATICALLY CAUSES THE COMPUTER TO CHANGE INTO THE ENLARGED FORMAT.
BY USING DIFFERENT COMBINATIONS OF THIS TECHNIQUE, YOU CAN MAKE IT A REAL PAIN FOR ANYONE TO ALTER OR ANALYSE YOUR BASIC CODE. AS AN IDEA, POKING 28 WILL HOME THE CURSOR, AND POKING 31 WILL ERASE TO THE END OF SCREEN!
NICE ONE EH?

--== oØo ==--

CONTINUED FROM PAGE 2

FOR THOSE OF YOU WHO VISIT THE WEB-SITE ON A REGULAR BASIS, YOU'LL ALREADY KNOW I'VE PUT OUT A REQUEST FOR HELP ON AN ARTICLE TO CONVERT A PROGRAM FROM APPLE II BASIC TO LEVEL II BASIC. IT IS SUPPOSED TO PRODUCE A SUDOKU PUZZLE, ON A 9 X 9 GRID.
I JUST THOUGHT THAT PRODUCING SUDOKUS ON A LEVEL 2, MODEL 1 WOULD BE JUST AMAZING AND MAKE A GREAT NOVELTY IDEA FOR THE CHRISTMAS EDITION. (WHICH, BY THE WAY, SHOULD BE AVAILABLE EARLY DECEMBER 2008.)

I ALSO 'WON', ON EBAY, A DPM-100 PRINTER BUT, SO FAR, I HAVEN'T BEEN ABLE TO GET IT RUNNING USING THE TANDY PRINTER TO KEYBOARD INTERFACE CONNECTOR AS YET. IT SEEMS TO POWER UP OK, BUT THEN JUST SITS THERE - LOOKING AMAZING, BUT DOING NOTHING!

I'M STILL LOOKING FOR MORE INFORMATION REGARDING THE 'ACULAB FLOPPY TAPE' PLEASE LET ME KNOW IF YOU COME ACROSS ANYTHING OF INTEREST. IT WOULD BE MOST APPRECIATED.

WELL, I THINK THAT JUST ABOUT WRAPS UP THIS ISSUE. IF YOU'VE ANY SPECIAL REQUESTS OR IDEAS FOR THE XMAS ISSUE PLEASE ASK AND I'LL DO MY BEST TO INCLUDE THEM.
IN THE MEAN TIME -

TAKE CARE

DUSTY



TRS8BIT



DEE AND I WISH

ALL OF OUR FRIENDS
OUT THERE IN TRS-80
LAND

A VERY
MERRY CHRISTMAS
AND A HAPPY
NEW YEAR

I HOPE, ONCE AGAIN, I'VE MANAGED TO PUT TOGETHER SOME INTERESTING ITEMS FOR THIS ISSUE.

'STAR BILLING' IN THIS ISSUE MUST GO TO PETER PHILLIPS WHO HAS GIVEN US ALL A CHRISTMAS PRESENT BY CONVERTING AN APPLE 2 SUDOKU CREATION PROGRAM TO RUN ON A LEVEL 2 16K MACHINE. (OF COURSE, IT WORKS JUST AS WELL WITH DISK BASIC!). ACCORDING TO PETER, THE PROGRAM WAS ORIGINALLY WRITTEN IN APPLE 'INTEGER' BASIC. I MUST ADMIT, IT SEEMED A BIT ODD TO ME. HOWEVER, PETER HAS DONE US PROUD; YET ANOTHER PROGRAM FOR A 30 YEAR OLD COMPUTER. IT WORKS SURPRISINGLY QUICKLY, MANAGING TO PRODUCE A PUZZLE IN APPROX. 10 SECONDS. BECAUSE OF THE PROGRAMS SIZE, I'VE PLACED A ZIP FILE FOR DOWNLOADING IT, ON THE WEB SITE WITH A COPY IN BOTH DISK AND CASSETTE FORMAT. (JUST TO SAVE THOSE RSI FINGERS!). I HOPE YOU ENJOY IT AS MUCH AS DEE AND I HAVE.

BY WAY OF A XMAS TRADITION, IF WE'VE BEEN ABOUT LONG ENOUGH TO HAVE A 'TRADITION', I'VE PRODUCED ANOTHER 'ASCII' PRINT WITH A SUITABLE XMAS FLAVOUR. THE PROGRAM IS NEARLY IDENTICAL TO THE ONE USED LAST XMAS, EXCEPT FOR AN ADDITIONAL LINE FOR YOU LUCKY PEOPLE WITH MORE THAN 16K OF RAM. OTHER THAN THAT, IT'S ONLY THE DATA LINES THAT HAVE CHANGED.

PERHAPS SOME OF YOU WILL REMEMBER THAT A COUPLE OF ISSUES AGO I MENTIONED THAT I WAS TAKING OVER THE JOB AS TREASURE OF MY LOCAL BRANCH OF THE BRITISH PRINTING SOCIETY. THERE ARE A VERY SMALL NUMBER OF TRANSACTIONS THROUGHOUT THE YEAR SO I THOUGHT IT WOULD BE NICE TO TRY AND RUN THE ACCOUNTS USING VISICALC AND ELECTRIC PENCIL. IT ACTUALLY WORKED OUT QUITE WELL. I'D FORGOTTEN JUST HOW GOOD VISICALC IS (WAS?). THE ONLY PROBLEM I ENCOUNTERED WAS WITH THE RESULTING PRINT-OUT. I WANTED TO USE THIS TYPE FACE (TELEPRINTER) AND, NOT HAVING A TELETYPE 33 ANYMORE, I HAD TO 'CHEAT' A LITTLE. I DOWNLOADED A COPY OF MY END OF YEAR STATEMENT TO WINDOWS, USING MATTHEW REED'S 'TRSREAD' PROGRAM AND USED WINDOWS NOTEPAD TO CHANGE TO THE REQUIRED FONT.
JUST TO REFRESH YOUR MEMORY, I'VE INCLUDED A MODEL 3 VISICALC 'CRIB' SHEET (JUST IN CASE YOU WANT TO PLAY TOO!). SORRY IF THE REPRODUCTION IS NOT TOO GOOD, BUT THE ORIGINAL IS ON THE 'WAXY' PAPER WE ALL KNEW AND LOVED SO MUCH!

DEE'S BEEN PLAYING WITH A NEW GAME ON HER NINTENDO DS WHICH HAS, YET AGAIN, INSPIRED ME TO HAVE A GO AT THE ONE LINERS IN THIS ISSUE. THEY ARE ALL ABOUT HAND AND EYE COORDINATION. THE IDEA BEING THAT YOU JUST TYPE IN THE RESPECTIVE SCREEN NUMBER WHEN BITS OF IT LIGHT-UP. THE FIRST ONE WAS QUITE SIMPLE, BUT THE OTHER ONE TOOK A BIT MORE

EFFORT TO GET IT DOWN TO JUST THE ONE LINE! IT WAS THOSE PESKY 'IF' STATEMENTS. THEY WILL KEEP DROPPING YOU THROUGH TO THE NEXT LINE WHEN THEIR CONDITIONS ARE MET!! DON'T FORGET, THE LARGE SPACES IN BOTH PROGRAMS ARE CREATED BY THE DOWN ARROW AND NOT THE SPACE BAR. FOR A LONGER GAME, JUST INCREASE THE 'F' LOOP.

I'VE REVAMPED AN ORIGINAL ARTICLE BY LAURIE SHIELDS WHICH GIVES SOME GREAT IDEAS FOR CHANGING LOWER CASE TO UPPER CASE. THE SMALL ASSEMBLER PROGRAM, WITH NOTATION, IS INCLUDED TOGETHER WITH A BASIC PROGRAM THAT USES A 'USR' ROUTINE TO CALL IT. I HOPE YOU FIND IT OF SOME INTEREST. ANYONE FANCY CHANGING IT TO CONVERT UPPER TO LOWER CASE?

WELL, THAT JUST ABOUT WRAPS UP THIS ISSUE. I'D LOVE TO HAVE YOUR FEEDBACK, THE NEXT ISSUE SHOULD BE OUT EARLY MARCH 2009, SO, IN THE MEANTIME, TAKE CARE



BYE FOR NOW
DUSTY

AT THE
READY>
PROMPT



I WAS BUSY (PLAYING)

WITH MY MODEL1 THE OTHER DAY, FILLING ARRAYS WITH GENERATED RANDOM NUMBERS AND I SUDDENLY THOUGHT IT WOULD BE A GOOD IDEA IF THE MACHINE GAVE ME AN INDICATION THAT IT WAS WORKING OK (OTHER THAN PRINTING ENDLESS RANDOM NUMBERS!). A SIMPLE FLASHING ASTERISK, AS USED FOR THE CASSETTE TAPE I/O WOULD BE JUST THE TICKET. IT CAN BE DONE QUITE EASILY IN BASIC AS:-

```
10 IF PEEK(15423)=42 THEN POKE 15423,32 ELSE POKE 15423,42
```

THE TOP RIGHT HAND CORNER OF THE SCREEN IS MEMORY ADDRESS 15423. IF IT CONTAINS AND ASTERISK (ASCII 42), THEN WRITE A SPACE (ASCII 32) ELSE WRITE AN ASTERISK. THIS WORKS FINE BUT IS NOT VERY ELEGANT WHEN COMPARED TO THE CODE IN ROM WHICH SIMPLY XOR'S THE CONTENTS OF 15423 WITH 10.
WRITING A SMALL MACHINE CODE ROUTINE SHOULD BE QUITE SIMPLE BUT THEN I THOUGHT, AS THE ROUTINE IS ALREADY IN ROM AT, 022CH, (ACCORDING TO JAMES LEE FARVOUR'S M/S BASIC DECODED) I COULD USE THAT! FOR LEVEL 2 USERS JUST

```
POKE16524,44:POKE16527,2
```

(FOR DISK BASIC USERS TYPE

```
DEFUSR0=&H022C)
X=USR(0)
```

THIS LITTLE BIT OF CODE WILL DO THE EQUIVALENT OF A GOSUB AND PUT OR WIPE-OUT AN ASTERISK IN THE TOP R/H CORNER OF THE SCREEN AS PER THE EXAMPLE BELOW.

```
300 POKE 16526,44:POKE16527,2
330 X=RND(1000)
350 IF X/10= INT(X/10) THENY=USR(0)
360 PRINT@300,X
400 GOTO330
```

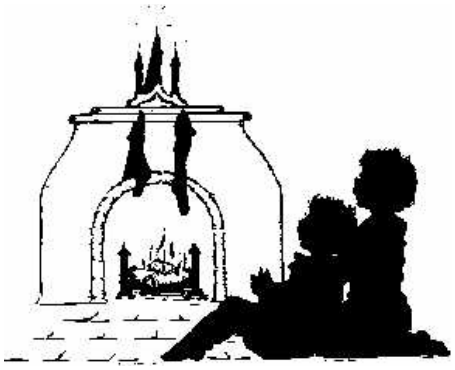
```
1 CLS:PRINT@30,"1 - 2";:FORB=6TO40:SET(65,B):NEXT:FORF=1TO10:X=RND(125):Y=RND(47):SET(X,Y):S=(X<64)+2:FORA=0TO1STEP0:CT=CT+1:A$=INKEY$:IFVAL(A$)=STHENRESET(X,Y):NEXTF:CLS:PRINT"WELL DONE YOUR SCORE WAS";CT:END:ELSENEXTA
```

TEXT CONVERSION

AND OTHER BITS.
FROM AN ORIGINAL IDEA OF
LAURIE SHIELDS

NONE OF MY MODEL 1

COLLECTIONS HAS THE LOWER-CASE MOD FITTED, BUT MATTHEW REED'S EMULATOR COME WITH L/C AS STANDARD. (ALL THAT'S NEEDED IS A DRIVER!). FOR THE BENEFIT OF ANYONE OUT THERE WITHOUT L/C, I THINK YOU'VE ONLY GOT HALF A COMPUTER! ALTHOUGH I LOVE MY TELETYPE FONT, ALL PROGRAM TITLES AND PROMPTS ON THE SCREEN LOOK INFINITELY BETTER WITH L/C, AS DOES ANY PRINTER OUTPUT. L/C IS TAKEN FOR GRANTED ON ALL MODERN COMPUTERS. I DOUBT WHETHER ANYONE EVER THINKS ABOUT IT THESE DAYS. HOWEVER, ONCE L/C IS FITTED (OR BEING USED) PROBLEMS CAN ARISE WHEN STRING COMPARISONS ARE REQUIRED. FOR EXAMPLE A STOCK NUMBER OF 123XYZ IS NOT THE SAME IN UPPERCASE AS IT IS IN LOWER CASE. THIS IS MOST ANNOYING AS A ROUTINE IN BASIC TO CHECK INPUT AND CONVERT ANY LOWER CASE TO UPPERCASE CAN BE RATHER SLOW AND WILL INVARIABLY INVOKE A STRING GARBAGE COLLECTION 'FREEZE UP'.



WELL, THAT WAS THE PROBLEM I CAME ACROSS AND AS A RESULT, THE FOLLOWING USR ROUTINE EMERGED. IN MACHINE CODE, IT IS EASY, ONCE WE KNOW WHERE ABOUTS IN MEMORY THE CHARACTER STRING IS LOCATED AND ALSO HOW MANY BYTES ARE INVOLVED. MICROSOFT MUST HAVE HAD JUST THIS SORT OF PROBLEM IN MIND WHEN THEY PUT TOGETHER LII BASIC, AS THE FUNCTION VARPTR TELLS US ALL THAT WE NEED TO KNOW BY SIMPLY INCLUDING IT WITHIN THE USR BRACKETS.

TO GET THE VALUE ENCLOSED IN THE USR BRACKETS, IN OUR CASE, THE VARPTR OF THE POSSIBLY LOWER CASE STRING, WE HAVE TO CALL A ROM ROUTINE AT ØA7FH AND THE INFORMATION RETURNS IN THE HL REGISTERS. THE INFORMATION IS IN FACT THE ADDRESS OF THE FIRST BYTE OF THREE THAT GIVE DETAILS ABOUT THE STRING VARIABLE AND THIS ONE IS SIMPLY THE LENGTH OF THE STRING. IF IT'S ZERO, THEN WE HAVE A NULL STRING. THE LARGEST WOULD BE 255, WHICH IS WHY THE LONGEST STRING WE ARE ALLOWED IS 255 BYTES, AS ANY MORE WOULD REQUIRE MORE THAN ONE BYTE FOR THE SIZE. THE NEXT TWO BYTES GIVE US THE ADDRESS OF SOMEWHERE DIFFERENT IN MEMORY WHERE THE STRING IS LOCATED. THIS ADDRESS IS STORED IN THE USUAL Z8Ø FORMAT OF LEAST SIGNIFICANT BYTE FIRST FOLLOWED BY THE MOST SIGNIFICANT BYTE.

IF WE WANTED TO BE BULLET PROOF THEN WE OUGHT TO INCLUDE A TEST ON WHETHER OR NOT THE VARPTR SUPPLIED FROM BASIC BELONGS TO A STRING OR SOME OTHER VARIABLE. THIS WE COULD DO BY LOOKING THREE BYTES BEFORE THE SIZE AND CHECKING THAT THE VALUE THERE IS THREE. IF NOT, THE VARPTR BELONGS TO SOMETHING OTHER THAN A STRING. ONCE WE HAVE CHECKED THAT THE SIZE ISN'T ZERO AND THE ACTUAL

ADDRESS HAS BEEN LOADED INTO THE 'D' AND 'E' REGISTERS, THEN WE GO ROUND A SIMPLE LOOP, LOOPING THE 'A' REGISTER WITH EACH CHARACTER IN TURN, CHECKING IF IT IS OUTSIDE THE RANGE 'A' TO 'Z', AND IF NOT, THEN RE-SETTING BIT 5 TO ZERO BY AND-ING WITH 5FH, PUTTING THE RESULT BACK WHERE WE FOUND IT. ON COMPLETION WE SIMPLY RET-URN TO BASIC.

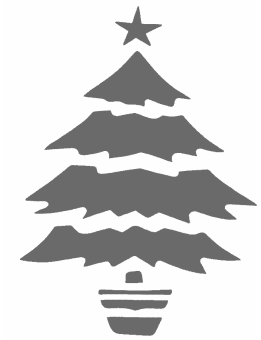
TO GET THIS ROUTINE WITHIN A BASIC PROGRAM, WE CONVERT ALL OF THE BYTES INTO DECIMAL NUMBERS. EG CD = 12*16 + 13, 7F = 7*16 + 15 ETC AND WRITING THEM INTO DATA STATEMENTS, BUT AS THIS SYSTEM OF PUTTING THEM BACK INTO MEMORY REQUIRES AN EVEN NUMBER OF BYTES WE MUST ADD A ZERO AT THE END GIVING 28 BYTES IN TOTAL. WE ARE GOING TO STORE THEM IN AN INTEGER ARRAY WHERE EACH VALUE IN THE ARRAY HOLDS 2 BYTES SO FOR 28 WE NEED TO DIMENSION 13 AS WE WILL USE THE ZERO ONE.

TO GET THE VALUES INTO THE ARRAY, WE MULTIPLY THE SECOND ONE OF THE PAIR BY 256 AND ADD IT TO THE FIRST BUT CHECKING THAT THE RESULT IS NOT GREATER THAN 32767 BEFORE ALLOCATING IT TO THE INTEGER ARRAY. IF THE RESULT IS GREATER THAN 32767 THEN SUBTRACT 65536 FIRST. HOWEVER, I DON'T LIKE THESE BIG NUMBERS AND AS THE ONLY WAY TO END UP GREATER THAN 32767 IS FOR THE SECOND NUMBER OF THE PAIR TO BE GREATER THAN 127 IN THE FIRST PLACE, THEN THIS IS SIMPLER TO SORT OUT,

IF YOU ARE FORTUNATE ENOUGH TO BE USING THE ZEN EDITOR/ ASSEMBLER, THEN INCLUDE A LOAD 8800H COMMAND JUST AFTER THE ORG STATEMENT AND THEN PEEK AT MEMORY ADDRESSES 8800H TO 881AH AND GET THE DECIMAL VALUES OF THE MACHINE CODE BYTES. CONVERT THESE INTO DATA STATEMENTS AND YOUR FINISH PROGRAM SHOULD LOOK SOMETHING LIKE THE ONE ON PAGE 5.

IT IS POSSIBLE TO GO EVEN FURTHER TO MAKE LIFE MUCH EASIER. IT'S AN INTERESTING AND CLEVER LITTLE 'TRICK' IF YOU'RE USING ZEN (OR ANY EDITOR ASSEMBLER THAT ALLOWS YOU TO LOAD CODE INTO MEMORY) AND NEWDOS80 V2. YOU CAN CREATE THE DATA LINES FOR YOUR PROGRAM AUTOMATICALLY VIZ:-

```
OPEN "0",1,"DATA/BAS": ?
£1,"110 DATA ";:FORX=&H8800
TO &H881A: ?£1,PEEK
(X)",":NEXT:£1,CHR$(13):
     CLOSE
```



AND THEN LOAD INTO BASIC "DATA/BAS". LIST THE PROGRAM AND YOU'LL SEE YOU'RE ALREADY HALF WAY TO WRITING YOUR PROGRAM AND NOT HAD TO GO THROUGH THE HASSLE OF CONVERTING A LOAD OF HEX VALUES TO DECIMAL.

I ASSUME THAT A SIMILAR STATEMENT IS ALLOWED IN LDOS. IF ANYONE WOULD CARE TO SEND IT IN, I'LL BE HAPPY TO PUBLISH IT IN THE NEXT EDITION OF TRS80BIT.

NOT WISHING TO 'TEACH MY GRANDMOTHER TO SUCK EGGS', BUT LINE 40 IS ALSO A NICE LITTLE 'WHEEZE' IF YOU HAVEN'T COME ACROSS IT BEFORE.
THE RESULT, IN BASIC, OF THE STATEMENT (M>127) WILL EVALUATE TO -1 IF TRUE OR 0 IF NOT TRUE, EVEN WITHOUT AN

'IF' COMMAND BEFORE IT. THE EXPRESSION 256*(M>127) WILL RESULT IN EITHER -256 OR Ø FOR EITHER TRUE OF FALSE. THEREFORE, M+256*(M>127) WILL, IN ONE STATEMENT, RESULT IN THE VALUE M-256 IF M IS GREATER THAN 127 AND JUST M OTHERWISE. SO BY READING FROM THE DATA STATEMENT PAIRS OF VALUES INTO VARIABLES L AND M (YOU GUESSED IT, LEAST AND MOST SIGNIFICANT) WE CALCULATE THE COMBINED VALUE AS :- 256*(M+256*(M>127))+L

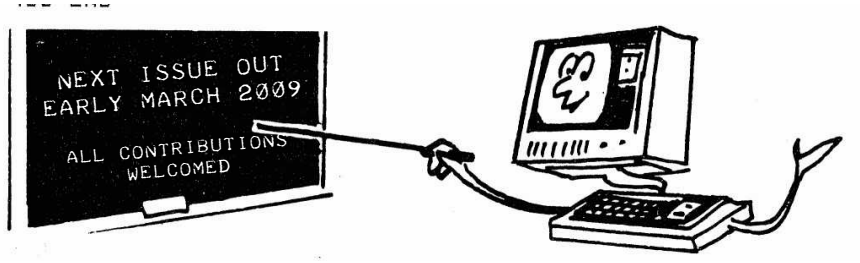
WITHOUT AN IF OR ELSE IN SIGHT!

N.B.
IT IS NECESSARY TO RE-DEFINE THE USR ROUTINE EACH TIME BEFORE USE SINCE INTRODUCING OTHER SIMPLE VARIABLES INTO THE RUNNING OF WHATEVER APPLICATION YOU PUT THIS ROUTINE TO, COULD CAUSE THE ACTUAL BYTES OF MEMORY ALLOCATED BY BASIC FOR THIS ARRAY TO BE CHANGED.

--== o0o ==--

```
 1                          ORG  8B00H
                            LOAD 8800H
 2 8B00 CD7F0A   START:     CALL 0A7FH       ;GET STRING POINTER
 3 8B03 7E                  LD   A,(HL)      ;GET LENGTH OF STRING
 4 8B04 B7                  OR   A           ;TEST IF NULL
 5 8B05 C8                  RET  Z           ;IF ZERO BYTES, RETURN
 6 8B06 47                  LD   B,A         ;PUT BYTE COUNT INTO REG B
 7 8B07 23                  INC  HL          ;POINT TO LSB IF STRING ADDR
 8 8B08 5E                  LD   E,(HL)      ;LOAD IN E REGISTER
 9 8B09 23                  INC  HL          ;POINT TO MSB OF STRING ADDR
10 8B0A 56                  LD   D,(HL)      ;DE NOW HAS ADDR OF STRING
11 8B0B 1A        LOOP:     LD   A,(DE)      ;GET BYTE FROM STRING
12 8B0C FE61                CP   "A"         ;IF < LOWERCASE A
13 8B0E 3807                JR   C,NEXT      ;IGNORE IT
14 8B10 FE7B                CP   "Z"+1       ;IF > LOWERCASE Z
15 8B12 3003                JR   NC,NEXT     ;IGNORE IT
16 8B14 E65F                AND  01011111B   ;MASK-OUT BIT 5
17 8B16 12                  LD   (DE),A      ;PUT BACK INTO STRING
18 8B17 13        NEXT:     INC  DE          ;POINT TO NEXT CHARACTER
19 8B18 10F1                DJNZ LOOP        ;GO ROUND LOOP
20 8B1A C9                  RET              ;ALL DONE - RETURN TO BASIC
21                          END
```

```
10 DEFINT J,L,M,N:DIMJL(13)
20 FORN=0TO13
30 READ L,M
40 JL(N)=256*(M+256*(M>127))+L
50 NEXT
60 INPUT"TEXT STRING ";IN$
70 DEFUSR=VARPTR(JL(0))
80 M=USR(VARPTR(IN$))
90 PRINT IN$
100 GOTO60
110 DATA 205,127,10,126,183,200,71,35,94,35,86,26,254,97
120 DATA 56,7,254,123,48,3,230,95,18,19,16,241,201,0
```



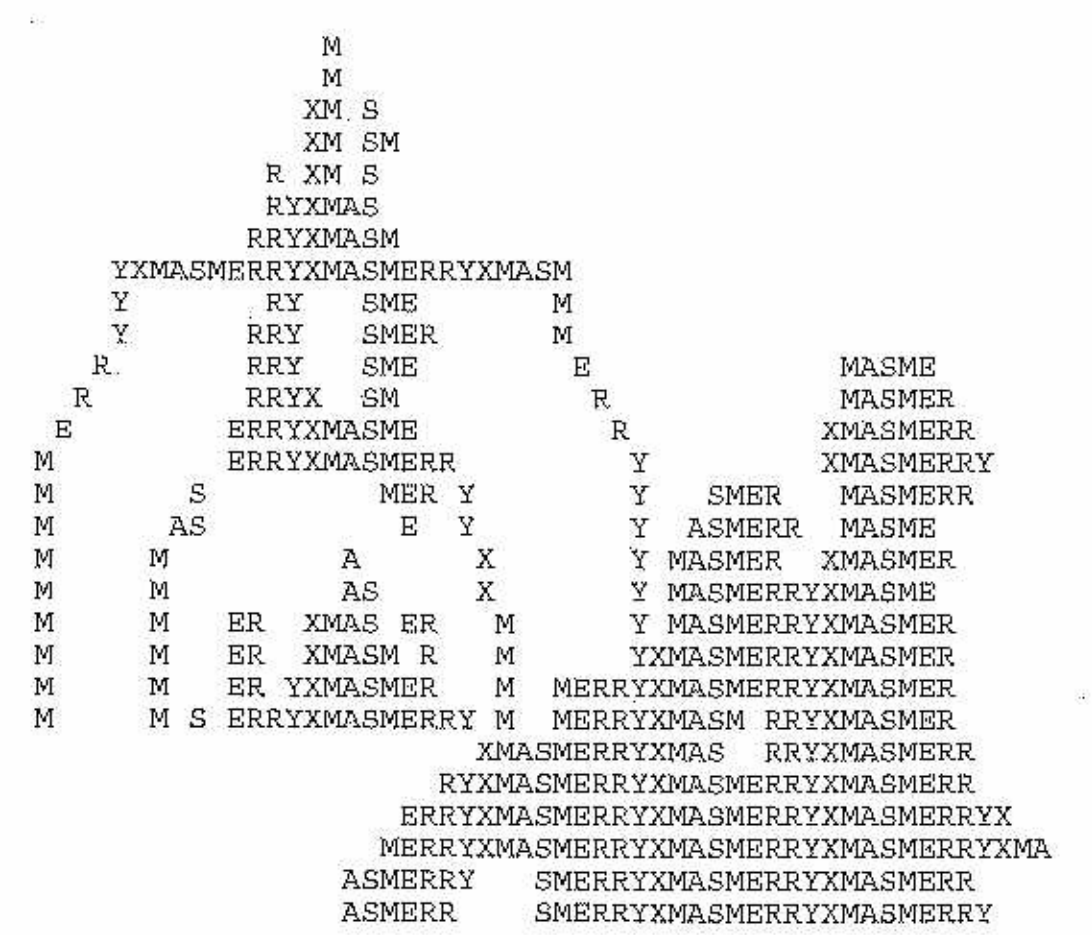
HERE'S THIS YEARS

XMAS PRINT. THE ORIGINAL CAN BE VIEWED ON PAGE 3.

I KNOW YOU NEED TO SQUINT A BIT AT IT BUT, HEY, GIVE ME A BREAK, IT IS CHRISTMAS AFTER ALL!

```
10 CLEAR500
20 M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXM"
30 GOSUB 200
40 READ S
50 IF S=0 GOSUB 150
60 READ L
70 T$=MID$(M$,S,L)
80      X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256
90 REM - THE LINE BELOW IS FOR 16K+ MACHINES -
100      IF X>32767 X=X-65536
110          FOR Y=1TOL
120          POKE X+S+Y,ASC(MID$(T$,Y,1))
130          NEXT Y
140 GOTO 40
150 LPRINT P$
160 GOSUB200
170 READ S
180 IF S=99 THEN END
190 RETURN
200 P$=STRING$(63,32)
210 RETURN
220 DATA16,1,0,16,1,0,15,2,18,1,0,15,2,18,2,0,13,1,15,2,18,1,0
230 DATA 13,6,0,12,8,0,5,24,0,5,1,13,2,18,3,28,1,0
240 DATA 5,1,12,3,18,4,28,1,0,4,1,12,3,18,3,29,1,43,5,0
250 DATA 3,1,12,4,18,2,30,1,43,6,0,2,1,11,10,31,1,42,8,0
260 DATA 1,1,11,12,32,1,42,9,0
270 DATA 1,1,9,1,19,3,23,1,32,1,36,4,43,7,0
280 DATA 1,1,8,2,20,1,23,1,32,1,35,6,43,5,0
290 DATA 1,1,7,1,17,1,24,1,32,1,34,6,42,7,0
300 DATA 1,1,7,1,17,2,24,1,32,1,34,14,0
310 DATA 1,1,7,1,11,2,15,4,20,2,25,1,32,1,34,15,0
320 DATA 1,1,7,1,11,2,15,5,21,1,25,1,39,1,32,17,0
330 DATA 1,1,7,1,11,2,14,8,25,1,28,21,0
340 DATA 1,1,7,1,9,1,11,13,25,1,28,10,39,10,0
350 DATA 24,13,39,11,0,22,28,0,20,32,0,19,35,0
360 DATA 17,7,27,23,0,17,6,27,24,0,99
```

```
1 CLS:PRINT@30,"1 - 2";:PRINT@990,"3 - 4";:FORA=34TO96:SET(A,23)
:NEXT:FORB=6TO40:SET(65,B):NEXT:FORF=1TO10:X=RND(125):Y=RND(47):
SET(X,Y):S=(X<64)+2:S=S+(Y>23)*-2:FORA=0TO1STEP0:D=D+1:A$=INKEY$
:IFVAL(A$)=STHENRESET(X,Y):NEXTF:PRINT"
SCORE=";D:END:ELSENEXTA
```



MERRY CHRISTMAS EVERYONE AND A
HAPPY NEW YEAR

--== oØo ==--

WITH THANKS TO PETER PHILLIPS,

HERE'S YOUR CHRISTMAS PRESENT FROM TRS8BIT.

IT'S A LEVEL II, 16K RUNABLE, VERSION OF A SUDOKU CREATION PROGRAM.

THE MORE I RUN THIS PROGRAM, THE MORE IMPRESSED I BECOME WITH IT! ON A STANDARD LEVEL II MACHINE, OR WITHIN MATTHEW REED'S EMULATOR RUNNING AT 1.7 MHZ IT MANAGES TO PRODUCE A PUZZLE IN APPROX. 10 SECONDS. QUITE AN IMPRESSIVE ACHIEVEMENT FOR A 30 YEAR OLD PIECE OF HARDWARE. RAMP UP THE SPEED ON THE EMULATOR AND IT'S TRULY AMAZING!

ANYONE OUT THERE FANCY IMPROVING IT? HOW ABOUT AN ADDITIONAL OPTION TO ENTER THE NUMBERS FROM AN EXISTING PUZZLE AND CALCULATE THE ANSWER, INSTEAD OF USING ALL RANDOM NUMBERS?

DON'T FORGET, TO SAVE YOU TYPING THE PROGRAM IN, A DOWNLOAD IS AVAILABLE ON THE WEBSITE. JUST CLICK ON THE NAMED BOX.

```
5 DEFINTA-Z
10 DIM A(81),B(81)
20 PRINT "LARRY NELSON'S SUDOKU FOR THE APPLE II"
25 PRINT " (CONVERTED FOR TRS-80 BY PETER PHILLIPS)"
30 PRINT: INPUT"WANT INSTRUCTIONS";D$:IF LEFT$(D$,1)="Y" THEN GOSUB 1000
40 PRINT"CREATING..";
50 GOSUB 4000
60 GOSUB 5000
70 GOSUB 3000
80 GOSUB 7000
90 GOSUB 6100
95 F=0
100 PRINT "  1=NEW PUZZLE": PRINT"  2=ANSWER TO PUZZLE":
105 IF F=0 THEN PRINT "  3=SHOW PUZZLE AGAIN"
107 PRINT "  4=QUIT"
110 INPUT A
120 IF A=1 THEN 40
130 IF A=2 THEN GOSUB 6000
135 IF A=3 AND F=0 THEN GOSUB 6100
140 IF A=4 THEN END
150 GOTO 100
1000 PRINT"SUDOKU IS A NUMBER PUZZLE IN A 9X9 GRID."
1010 PRINT"THE GRID IS SPLIT INTO 9 3X3 MINI-GRIDS."
1020 PRINT"SINGLE DIGITS FILL THE GRID. THE DIGITS"
1040 PRINT"MAY BE IN ANY ORDER. THERE ARE JUST"
1050 PRINT"THREE RULES FOR SOLVING THE GRID--"
1060 PRINT"-EACH ROW MUST HAVE ALL THE DIGITS 1-9."
1070 PRINT"-EACH COLUMN HAS ALL THE DIGITS 1-9."
1080 PRINT"-EACH MINI-GRID MUST ALSO HAVE ALL THE"
1090 PRINT"  DIGITS FROM 1-9."
1100 PRINT:INPUT"  HIT ENTER TO CONTINUE";D$
1110 RETURN
3000 PRINT ".";
3010 FORI=1TO81:A(I)=B(I):NEXT
3020 FOR I=1 TO 7 STEP 3
3030 X=RND(6)-1
3040 IF X=0 THEN 3160
3050 Y=0:Z=2
3060 IF X=1 THEN Y=1
3070 IF X=3 THEN Z=1
3080 GOSUB 3400
3090 IF X<4 THEN 3160
3100 IF X=5 THEN 3130
3110 Z=1:GOSUB 3400
3120 GOTO 3160
3130 Y=1:GOSUB 3400
3160 NEXT I
3170 FORI=1TO81:B(I)=A(I):NEXT
3180 RETURN
3400 FOR J=0 TO 8
3420 W=9*J+I+Y:V=9*J+I+Z
3425 P=A(W):Q=A(V)
3430 A(W)=Q:A(V)=P
3470 NEXT J
3480 RETURN
4000 PRINT ".";
4010 FORI=1TO9:A(I)=I:NEXT
4020 FOR I=1 TO 30
4030 X=RND(9):Y=RND(9)
```

```
4030 X=RND(9):Y=RND(9)
4035 IF X>=Y THEN 4030
4040 B=A(Y):C=A(X)
4050 A(X)=B:A(Y)=C
4110 NEXT I
4120 RETURN
5000 PRINT".";
5005 FORI=1TO81:B(I)=0:NEXT
5010 FORI=1TO9:A(I+9)=A(I):NEXT
5015 P=1
5020 FOR I=0 TO 2
5030 FOR J=1 TO 7 STEP 3
5040 FOR K=0 TO 8:B(P+K)=A(I+J+K):NEXT
5045 P=P+9
5050 NEXT J,I
5060 RETURN
6000 REM
6010 INPUT "ARE YOU SURE YOU WANT THE ANSWER (Y/N)";D$
6020 IF D$<>"Y" THEN RETURN
6030 F=1:FORI=1TO81:A(I)=B(I):NEXT
6100 F$=CHR$(191):L$=STRING$(11,140):T$=CHR$(188)
6105 PRINT : PRINT T$;L$;T$;L$;T$;L$;T$
6110 FOR J=0 TO 8
6120 PRINT F$;
6130 FOR I=1 TO 9:K=9*J+I
6140 IF A(K)<>0 PRINT A(K); ELSE PRINT "   ";
6150 IF I - 3*INT(I/3) =0 THEN PRINT F$;
6160 IF I - 3*INT(I/3) >0 THEN PRINT ":";
6170 NEXT I
6175 PRINT
6177 IF J > 7 THEN F$=CHR$(143)
6190 IF J+1 - 3*INT((J+1)/3) =0 THEN PRINT F$;L$;F$;L$;F$;L$;F$
6200 NEXT J
6210 PRINT
6220 INPUT "HIT ENTER TO CONTINUE";D$
6230 RETURN
7000 REM
7010 FORI=1TO81:A(I)=B(I):NEXT
7020 PRINT:PRINT "HOW DIFFICULT DO YOU WANT YOUR PUZZLE":PRINT "
1=EASY": PRINT "  2=MEDIUM": PRINT "  3=HARD"
7030 INPUT A
7040 IF A<1 OR A>3 THEN 7020
7050 W=2*A+20+RND(2)-1
7060 FOR I=1 TO W
7070 X= RND (81)
7080 IF A(X)=0 THEN 7070
7090 A(X)=0
7100 X=82-X
7110 A(X)=0
7120 NEXT I
7130 RETURN
8000 REM PORTING COMMENTS
8010 REM 4000 CREATES A RANDOM ORDERING OF 1-9
8020 REM 5000 FILLS PUZZLE WITH ROTATIONS OF THE RANDOM ORDERING
8030 REM 3000 MORE-OR-LESS PERMUTES THE COLUMNS
8040 REM 7000 HIDES SOME OF THE ELEMENTS.
8050 REM 6100 PRINTS THE PUZZLE ARRAY.
```

--== o0o ==--

TRS8BIT - VOL 02 NO 04 - CHRISTMAS 2008 - PAGE 10

VISICALC®

POCKET REFERENCE

FOR THE TRS-80® MODEL III



Manufactured for

Radio Shack®

A DIVISION OF TANDY CORPORATION

By

PERSONAL SOFTWARE INC.

Program by

Software Arts, Inc.

VisiCalc® is a Registered Trademark of Personal Software Inc.
DIF™ is a Trademark of Software Arts, Inc.
TRS-80® is a Trademark of Tandy Corporation

Software Copyright © 1979, 1981 Software Arts, Inc.

KEY FUNCTIONS

MOVING THE CURSOR

→ ← ↑ ↓	Moves cursor in designated direction. Used in value entry and some commands to point to an entry position that will be selected when ENTER or a colon is pressed. Move edit cursor in Edit Command (see /E).
;	Moves cursor between windows (see /W).
>*coord*	Go To Command. Cursor moves to designated coordinate. ENTER completes command.

CALCULATING

+	Addition
−	Subtraction
*	Multiplication
/	Division
SHIFT @	Exponentiation character ∧.
#	If coordinate is immediately to left of edit cursor on edit line, replaces coordinate with that location's current value. Otherwise places current value of cursor location on edit line.
!	When the edit line is clear, forces recalculation of entire sheet. When formula is on the edit line, replaces formula on edit line with its current value.

EDITING

BREAK	Exits command.
CLEAR	Edit cursor deletes one position. If edit line is blank, exits command.
SHIFT-ENTER	Invokes Edit Command (see /E).

LABELS

Start with any letter or ". The initial quotation mark does not appear on the edit line. Defines the entry as a label with the numeric value of zero. Labels are left-aligned in the entry position. Correct typing errors while entering with the CLEAR key or SHIFT-ENTER (see /E).

VALUES

Start with a digit (0 through 9), plus (+), minus (−), open parenthesis, period (.), number sign (#), or at sign (@). Values are stored with 11 to 12 significant digits. In general format, program shifts between conventional and scientific notation as required to display the calculated value to the greatest precision. If column is too narrow to display the number in scientific notation, greater-than signs (>) fill the entry position.

VALUE REFERENCE

Entry position coordinate used as element in a formula. Value references are allowed wherever numbers are allowed. To start a value entry with a value reference, begin with +.

FILE NAMES

A file name is one to eight characters beginning with a letter, optionally followed by /VC for a worksheet, /PRF for a print file, or /DIF for worksheet in DIF™ format. The VisiCalc program adds the suffixes /VC, /PRF and /DIF. Disk drive specified by final suffix :0 or :1.

FUNCTIONS

Provide common formulas for use in values. Each function begins with @ followed by the name of the function; most functions also require an expression in parentheses (the argument). An argument includes the values (formulas and numbers) upon which the function calculates. The arguments shown as *v* may be any legal VisiCalc values. The arguments shown as *l* may be any logical values. The arguments shown as *list* may be any combination of values and ranges separated by commas. A *range* is a portion of a row or column specified by its beginning coordinate, a period (shown as an ellipsis), and its final coordinate.

@ABS(*v*)	Absolute value of *v*.
@AVERAGE(*list*)	Average of the non-blank entries in *list*. Maximum of 255 entries in *list*.
@COUNT(*list*)	Number of non-blank entries in list. Maximum of 255 entries in *list*.
@EXP(*v*)	Natural exponent of *v*.
@INT(*v*)	Integer portion of *v*.
@LN(*v*)	Natural logarithm of *v*.
@LOG10(*v*)	Base 10 logarithm of *v*.
@MAX(*list*)	Maximum value in *list*.
@MIN(*list*)	Minimum value in *list*.
@NPV(*dr,range*)	Net present value of the cash flow in *range*, discounted at the rate specified in the expression *dr* (discount rate).
@SQRT(*v*)	Square root of *v*.
@SUM(*list*)	Sum of the values in *list*.

TRIGONOMETRIC FUNCTIONS

All angles are calculated in radians.

@ACOS(*v*)	Arccosine of *v*.
@ASIN(*v*)	Arcsine of *v*.
@ATAN(*v*)	Arctangent of *v*.
@COS(*v*)	Cosine of *v*.
@SIN(*v*)	Sine of *v*.
@TAN(*v*)	Tangent of *v*.

SEARCH FUNCTIONS

@CHOOSE (*v,list*)	Returns the *v*th element of *list*. If *v* is greater than the number of elements in *list*, NA is returned.
@LOOKUP (*v,range*)	Compares *v* to the successive values in *range* and returns the corresponding value from the column or row immediately to the right or below the entries in *range*.

FUNCTIONS WITHOUT ARGUMENTS

@ERROR	Makes all expressions referencing the value display ERROR.
@FALSE	Logical value FALSE.
@NA	Makes all expressions referencing the value display NA.
@PI	3.1415926536.
@TRUE	Logical value TRUE.

LOGICAL FUNCTIONS

@AND(*list*)	TRUE if all values in *list* are TRUE, otherwise FALSE.
@IF(*l,v1,v2*)	*v1* if *l* is TRUE; *v2* if *l* is FALSE.
@ISERROR(*v*)	TRUE if *v* is ERROR; otherwise FALSE.
@ISNA(*v*)	TRUE if *v* is NA; otherwise FALSE.
@NOT(*l*)	TRUE if *l* is FALSE; FALSE if *l* is TRUE.
@OR(*list*)	TRUE if any value in *list* is TRUE, otherwise FALSE.

COMMANDS

/B Blank – Erases the highlighted entry position. ENTER or arrow key completes command.

/C Clear – Clears the worksheet, restores the General format and single window. Y completes command, any other key cancels.

/D Delete – Deletes all the entries in the row (/DR) or column (/DC) on which the cursor rests.

/E Edit – Allows editing of entry position contents. Places the contents of the highlighted entry position on the edit line. The left and right arrow keys (→ and ←) move the edit cue over the characters without changing them. The ↑ arrow key moves the edit cue to the beginning of the edit line; the ↓ arrow key moves the edit cue to the end of the edit line. Enter or delete to the left of the edit cue. SHIFT-ENTER invokes the Edit Command while entering values and labels.

/F Format – Sets local format for the highlighted entry position but does not affect contents in memory. Remains until changed or the worksheet is cleared.

- /FD Default – Resets to the global format.
- /FG General – Maximum precision.
- /FI Integer – Integer.
- /FL Left-aligned.
- /FR Right-aligned.
- /F$ Dollars-and-cents (two decimal places).
- /F* Graph – Replaces value with asterisks equal to its integer value.

/G Global – Affects entire window or sheet.

- /GC Column – Sets column width of all columns to specified number of characters. ENTER completes command.
- /GF Format – Sets display format for all entry positions not specifically formatted. See /F for format specifications.
- /GO Order of recalculation – Specifies whether the worksheet is to be recalculated down the columns (/GOC) or across the rows (/GOR).
- /GR Recalculation priority – Sets recalculation to automatic (/GRA) or manual (/GRM). Typing ! causes recalculation.

/I Insert – Inserts a new blank row (/IR) or column (/IC) at the cursor position.

/M Move – Moves an entire row or column to another position on the sheet. The first coordinate on the edit line is the row or column to be moved. The final coordinate specifies where the row or column is to be moved. ENTER completes command.

/P See "Printing."

/– Repeating Label – Fills the entry position with the designated character(s).

/R See "Replicating."

/S See "Storing."

/T Titles – Fixes rows and columns in place on the screen so they remain in view when the window is scrolled. The position of the cursor determines which column(s) and/or row(s) to be fixed. The Go To command (>) moves the cursor into a fixed area.

- /TH Horizontal – Fixes rows at and above the cursor.
- /TV Vertical – Fixes columns at and to the left of the cursor.
- /TB Both – Fixes columns and rows.
- /TN None – Unfixes all rows and columns.

/V Version – Displays the copyright notice and version number on the prompt line.

/W Windows – Splits the screen vertically or horizontally at the current cursor position. The semicolon (;) moves the cursor from one window to the other. Global commands affect only the window containing the cursor.

- /WH Horizontal – Splits the window just above the row with the cursor.
- /WV Vertical – Splits the window just to the left of the column with the cursor.
- /W1 One – Returns to one window.
- /WS Synchronized scrolling – Split windows scroll together.
- /WU Unsynchronized scrolling – Cancels synchronized scrolling.

PRINTING

The Print command sends a rectangular portion of the worksheet to a printer or to diskette. Formulas are not sent (to print formulas and formats, see /SS). To print all or part of a VisiCalc worksheet:

1. Place the cursor on the upper-left coordinate of the rectangular area to be printed and type /P.
2. Type F to send to diskette file.

   Type P to send to parallel interface printer.

   Type R to send to an RS-232 interface printer.
3. To send setup codes, type ", the setup string, then ENTER. The following special characters can be used:

∧C *char*	Sends control *character*.
∧E	Sends the Escape character.
∧Hnn	Sends the single ASCII character defined by the hexadecimal digits nn.
∧R	Sends a return character.
∧L	Sends a line feed character.
∧∧	Sends one carat character (∧).

   The setup codes must be reentered each time the Print command is used. Invalid codes are sent to the printer – no error indication is given.
4. The VisiCalc program automatically sends a line feed with each carriage return.

–	Turns off the line feed.
&	Restores line feed.
5. Enter lower-right coordinate of the rectangle to be printed, either by typing it or pointing the cursor to it, and type ENTER.

BREAK stops printing.

REPLICATING

The Replicate command copies the entry contents of the source range (the position(s) to be copied) into the target range (the position(s) to which it will be copied).

A range is a portion of a row or column specified by its beginning coordinate, a period (shown as an ellipsis), and its final coordinate.

A colon separates the source range from the target range and is entered by typing either colon or ENTER.

	Source	Target
To copy one entry once	coordinate	: coordinate
To copy one entry several times	coordinate	: range in same row or column
To copy a row or column once	row or column range	: starting coordinate
To copy a row several times	row range	: range in same column
To copy a column several times	column range	: range in same row

ENTER completes command.

If the source range includes reference to other locations, indicate how each reference is to be treated:

N No Change - Copies the reference as it appears in the source entry.

R Relative - Changes the reference so that it is relative to the target position.

STORING

The Storage command writes a file to diskette, reads a saved file from diskette, writes and reads files in the DIF™ format, deletes files from a diskette, and quits the VisiCalc program. File names can be displayed from the diskette by pressing the right arrow key when prompted for file name.

/SL Load - Loads a file previously saved with /SS. The file is loaded over the current sheet. New entries replace old ones. Any entries that are not replaced remain the same.

/SS Save - Saves the information on the worksheet to diskette under the specified file name. To print out formulas and formats type /SS:P.

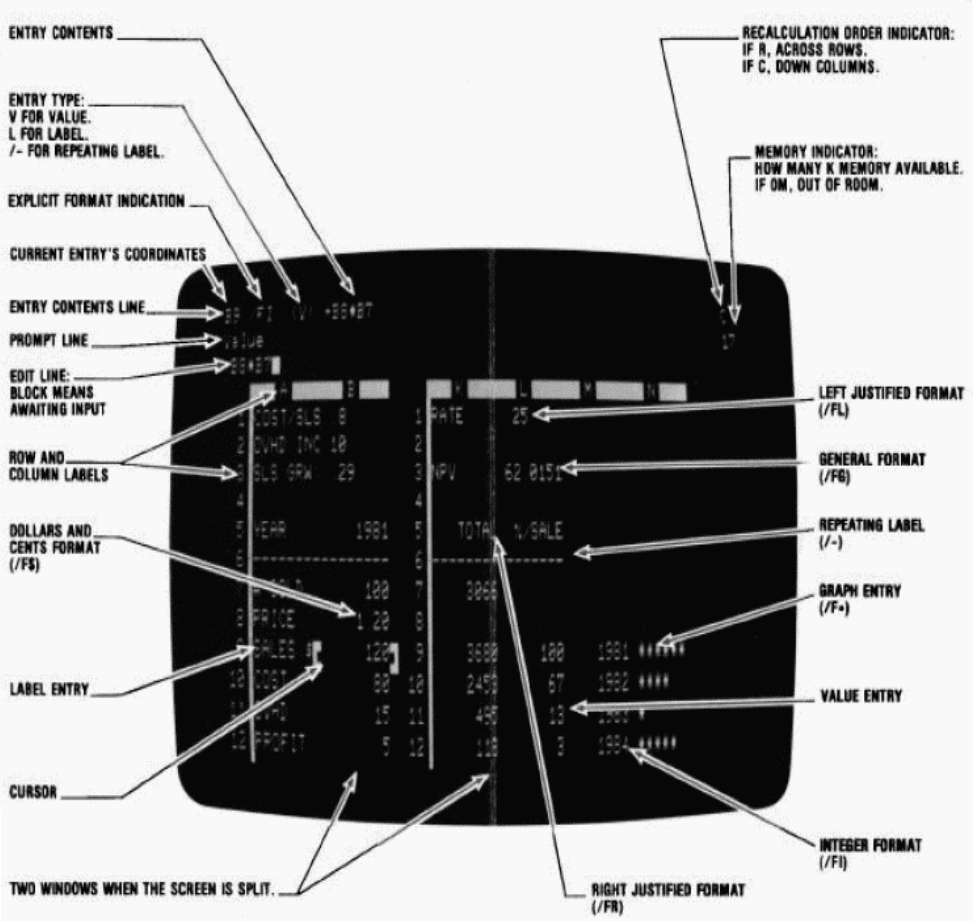
/SD Delete file - Deletes the specified file from the diskette.

/SQ Quit - Quits the VisiCalc program, and returns to the operating system. Y completes command, any other key cancels.

/S#S Saves the specified area of the worksheet to diskette as a file in the DIF™ format. R or ENTER saves by rows, C saves by columns.

/S#L Loads a file in the DIF™ format. R or ENTER loads the information by rows, C loads by columns.

VISICALC®



TRS8BIT



WELCOME TO THE LATEST EDITION OF TRS8BIT, IT'S THE START OF OUR 3RD YEAR AND STILL NEW AND EXCITING HARD AND SOFTWARE APPLICATIONS ARE MAKING THEIR APPEARANCE. THE MODEL 1'S STILL GOING STRONG AFTER 30 YEARS AND THE MODEL 100 AFTER 25+!

IF YOUR DOUBT IT, CHECKOUT PAGE 3 AND BE AMAZED BY THE APPEARANCE OF A 'NADSBOX'. IT'S KEN PETTIT'S EXTERNAL STORAGE DEVICE FOR THE MODEL 100, WHICH USES THE LATEST SD MEMORY CARDS! THERE'S A COUPLE OF SHORT REVIEWS TO WET YOUR APPETITE.
HAS ANYONE IN THE UK ORDERED ONE YET, OR ARE YOU UP AND RUNNING? A SHORT REVIEW WOULD BE MOST WELCOMED BY THE REST OF US!
IF ANYONE IN THE 'STATES' WOULD CARE TO GIVE US THEIR FIRST IMPRESSIONS, IT WOULD BE APPRECIATED.
ALSO, IS THERE ANYONE OUT THERE WORKING ON SOMETHING SIMILAR FOR THE MODELS 1,3 OR 4? PLEASE LET US KNOW IF YOU ARE.

ON THE SOFTWARE FRONT, FOR NEWDOS80 USERS, P. EREAUT HAS WRITTEN A COUPLE OF PROGRAMS. ONE THAT WILL DISPLAY EMULATOR DSK FILES, AND ANOTHER THAT WILL MAKE CHECKING YOUR NEWDOS80 PDRIVE SETTING A BREEZE. YOU CAN DOWNLOAD A COPY FROM IRA'S WEBSITE. THERE'S A LINK TO IRA'S FROM TRS-80.ORG.UK. IF YOU HIT ANY PROBLEMS, EMAIL ME AND I'LL TRY TO GET A COPY TO YOU.

EBAY HAS BEEN BUSY WITH TRS-80 ITEMS OVER THE LAST FEW MONTHS, SINCE THE START OF THE NEW YEAR IN FACT. HARDWARE WISE, MODEL 1'S & 4'S TOGETHER WITH A COUPLE OF VIDEO GENIES SEEM TO MAKE REGULAR APPEARANCES, WITH PRICES RANGING FROM £30 TO £50. ONCE AGAIN THOUGH, EBAY.COM HAS MUCH MORE STUFF THAN THE CO.UK VERSION, (AND, BY THE WAY, HAS A MUCH BETTER DISPLAY OPTION). KEEP YOUR EYES PEELED, WITH A BIT OF LUCK, YOU COULD WELL FIND YOURSELF A BARGAIN.

PETER STEVENS HAS GENEROUSLY DONATED A NUMBER OF TRS-80 RELATED ITEMS TO THE SITE. THERE ARE 2 MODEL 1 KEYBOARDS WHICH WERE ASSUMED TO BE FOR SPARES BUT WHICH SEEM TO WORK FINE. ONE HAS 48K INSIDE! I HAVEN'T OPENED IT UP, YET, BUT WILL KEEP YOU INFORMED! THERE'S ALSO AN ACULAB FLOPPY TAPE AND SOME WAFERS BUT, AS YET, IT'S NOT RESPONDING TO MY TLC. (AND UNFORTUNATELY, THERE IS NO PAPERWORK TO GO WITH IT EITHER!).
THERE ARE NEARLY 20 EDITIONS OF 80-MICRO AND ABOUT THE SAME NUMBER OF MICRO-80'S, MOST IN FAIRLY GOOD CONDITION.
IF THERE IS ANY PARTICULAR EDITION YOU WOULD LIKE TO BORROW, PLEASE EMAIL ME WITH DETAILS AND I'LL GET SOME INFORMATION ON POSTAGE COSTS.
THERE'S ALSO AN OKI MICROLINE-80 PRINTER WITH HALF A DOZEN SPARE RIBBONS, SO IF YOU NEED A SPARE RIBBON, ONCE AGAIN, PLEASE, LET ME KNOW AND I'LL GET ONE SHIPPED OUT TO YOU ASAP.

I'VE FOUND A COUPLE OF MODEL 4 TIPS IN 'AT THE READY PROMPT' TOGETHER WITH A COUPLE OF OTHER SNIPPETS.

FOR THE FIRST TIME, ALL THE ONE-LINERS (AND OTHER PROGRAMS) WERE DEVELOPED USING MATTHEW REED'S MODEL1 EMULATOR.
THERE'S A DECIMAL TO HEX CONVERTER. A SCREEN BUBBLE SORT USING A BIT OF ASSEMBLER, WHICH, BY THE WAY, I SEEMED TO RUN INTO TROUBLE WITH, WHEN USING ZEN TO ASSEMBLE THE SOURCE CODE. FOR SOME REASON, I HAD TO ADD ZERO TO THE CONTENTS OF THE IX REGISTER WHEN LOADING THE VALUE INTO A AND WHEN REVERSING THE PROCEDURE INTO D. (SEE LINES 10 & 16). IS THIS NORMAL, OR IS IT JUST MY LACK OF KNOWLEDGE USING ASSEMBLER?
THERE'S ALSO A MEMORY TEST PROGRAM AS PART OF AN ARTICLE ON HOW TO INCREASE THE MEMORY IN AN EXPANSION INTERFACE.

AND FINALLY, THERE'S A LITTLE BASIC AND MACHINE CODE PROGRAM TO INTRODUCE YOU TO THE AMAZING WORLD OF BELL RINGING!
I KNOW VERY LITTLE ON THE SUBJECT, BUT I THOUGHT IT WOULD BE FUN TO PUSH THE STANDARD MODEL 1 TO THE MAX AND SEE JUST WHAT IT SOUNDED LIKE (WITH NO EXTRA ADDED HARDWARE TO HELP). TINNY IT MIGHT BE, BUT I HOPE YOU ENJOY IT NONETHELESS.

THAT'S ABOUT IT FOR NOW, DROP ME A LINE IF YOU'RE UP TO MISCHIEF WITH YOUR TRS-80, WE'D ALL LOVE TO KNOW ABOUT IT !!



BYE FOR NOW

DUSTY

AT THE
READY>
PROMPT



I'VE NEVER OWNED OR USED A MODEL 4 SO I CAN'T VOUCH FOR THIS, BUT ACCORDING TO PAUL SVENTEK THIS IS A HANDY LITTLE ROUTINE TO ENABLE YOUR MODEL 4, RUNNING IN MODEL 3 MODE, UNDER TRSDOS 1.3, TO RUN AT THE HIGHER 4MHZ CLOCK SPEED.
AT THE BEGINNING FOR YOU BASIC PROGRAM, TYPE
X=PEEK
(16912):X=XOR64:POKE16912,X

ALSO, RATHER THAN GETTING YOUR MODEL 4 TO MAKE SOUNDS VIA JCL COMMANDS, YOU CAN MAKE IT GENERATE TONES IN BASIC BY USING THE 'SOUND' COMMAND. THE SYNTAX OF THE COMMAND IS SOUND TONE, DURATION. THE TONE MAY BE ANY VALUE THROUGH Ø TO 7, AND THE DURATION MAY BE ANY VALUE THROUGH Ø TO 31.
EXAMPLE - SOUND 4,22.

I'VE JUST COME ACROSS A REVIEW (IN AN OLD EDITION, FEBRUARY 1984, OF 80-MICRO) OF AN EXATRON/ ACULAB MICRO DRIVE FOR THE MODEL 100. IT WAS MADE BY HOLMES ENGINEERING AND WAS BATTERY POWERED. IT WAS ATTACHED TO THE MODEL 100 VIA THE 100'S RS232 PORT. IT SEEMS TO HAVE HAD 'RAVE REVIEWS' AT THE TIME. DID ANYONE OUT THERE IN TRSDOME EVER USE OR COME ACROSS ONE?

HERE'S ANOTHER LITTLE GEM I'VE ONLY JUST FOUND OUT ABOUT. ACCORDING TO EARLES L MCCAUL, LEVEL 2 BASIC USES THE IX REGISTER ONLY 21 TIMES AND DOES NOT USE THE IY REGISTER AT ALL.

EXPERNAL STORAGE FOR THE MODEL 100

NADSBOX

THE TARGET DATE FOR THE NEW AGE DIGITAL STORAGE BOX (NADSBOX) IS DECEMBER 1ST, 2008. THANKS TO OVER SIX YEARS OF DEDICATION, PERSONAL INVESTMENT AND HALF-MILLION LINES OF MACHINE CODE, KEN PETTIT IS FINALLY IN THE HOME STRETCH. BETA NADSBOXES HAVE GONE OUT AND THUS FAR ALL IS WELL. YOURS TRULY HAS ONE AND ... WOW. WE ARE NOW WAITING FOR A PRODUCTION RUN OF THE ENCLOSURES. EVERYTHING ELSE IS READY, I.E., PCB BOARDS, CARDS, ADAPTERS, ETC. THUS, NADSBOX IS REAL.

DUE TO KEN'S LOCATION (CHINA) AND WORK COMMITMENTS, CLUB 100 WILL BE THE ORDERING HEADQUARTERS FOR THE NADSBOX. CLUB 100'S TRACK RECORD AND COMMITMENT TO THE MODEL 100 OWNERS FOR OVER 25 YEARS MEANS EASY ORDERING, FAST FULFILLMENT AND COST EFFECTIVE SHIPPING.

TWO REASONS TO OWN A NADSBOX:

1) PORTABLE MASS STORAGE TO A MEDIA CARD, AND
2) 2) MEDIA CARD IS READABLE BY A PC OR MAC CARD READER, I.E., EASY FILE TRANSFERS.

PLUGS INTO THE RS232 PORT OF ANY MODEL 100, 102, 200 (AKA MODEL "T") ... AND NOW SUPPORTS THE WP-2 AS WELL.

READY TO RUN ... OUT OF THE BOX. COMES WITH CONNECTER, MEDIA CARD, 2 AA BATTERIES AND A USB TO DC POWER ADAPTER CABLE.

LOOKS LIKE A TPDD TO THE MODEL "T".

SAVE, LOAD, RENAME, ETC. FILES VIA ANY DOS, SUCH AS TS-DOS OR FROM A COMMAND LINE IN TELCOM (WOW!) ... LIKE THE OLD DOS PROMPT!

USES AN OFF-THE-SHELF SD, SDHC OR MULTIMEDIA CARD, SUCH AS SANDISK, KINGSTON, LEXAR, ETC. SDHC MEANS SUPPORT FOR THE NEWEST 16GB CARDS! THAT'S 167,000 TPDD FLOPPIES!

WHEN REMOVED FROM THE NADSBOX AND PLUGGED INTO A PC, MAC OR LINUX CARD READER, THE MEDIA CARD LOOKS LIKE A FOLDER ... NO-BRAINER FILE TRANSFERS!!!

THE NADSBOX FIRMWARE CAN BE EASILY UPGRADED IN-THE-FIELD TO TAKE ADVANTAGE OF NEW FEATURES AND CODE CHANGES.

ALL SUPPORT FILES WILL BE HOSTED AT WWW.CLUB100.ORG.

LET ME SAY SOMETHING REALLY IMPORTANT IN SUPPORT OF OUR MODEL 100S, 102S, 200S, WP-

---

HERE'S A ONE-LINER TO CONVERT DECIMAL TO HEX WHICH I HAVE CREATED, FROM AN ORIGINAL MULTI-LINED PROGRAM IDEA BY BILL SHORT. THERE'S NO ERROR CHECKING ETC., BUT EVEN IN ONE LINE, THERE'S STILL PLENTY OF ROOM LEFT IF YOU WISH TO PUT IT IN!

```
1 H$="":INPUT"DECIMAL NO. ";D:FORI=1TO4:H=DAND15:H$=CHR$(H+48-7*
(H>9))+H$:D=D/16:NEXTI:PRINTH$
```

2S AND NEC 8201A/8300S. THE MODEL 100 CAME TO US VIA BILL GATES AND TANDY / RADIO SHACK IN MARCH 1983. IT'S BEEN OVER 25 YEARS AND WE ARE STILL HERE. WOW! AND OVER ALL THIS TIME, THE TOP WISH LIST ITEMS HAVE BEEN PORTABLE MASS STORAGE AND FILE TRANSFERS.

KEN PETTIT'S NADSBOX DOES BOTH VERY WELL. NADSBOX "IS" THE #1 TOOL THAT TAKES MODEL "T" COMPUTING INTO THE 3RD DECADE AND BEYOND. EVERY SERIOUS MODEL "T"ER NEEDS A NADSBOX. WHY? IT WORKS, IT'S COOL AND WE NEED TO STAND UP AND SUPPORT KEN. LET ME ALSO ADD THAT IN THE WINGS IS REX BY STEVE ADOLPH. REX AND NADSBOX CROSS-SUPPORT EACH OTHER, I.E., YOU WILL BE BLOWN AWAY BY THE TEAM WORK.

NADSBOX PRICING

THE NADSBOX HAS IS PRICED REASONABLY AT $195. SHIPPING IS $10 FLAT RATE WORLD WIDE!! CLUB 100 OFFERS, VISA, MASTER CARD, CHECKS, MONEY ORDERS AND PAYPAL. PLEASE NOTE: PAYPAL IS PREFERRED.

PRE ORDER ... STARTING NOW!!!

CLUB 100 IS TAKING PRE-ORDERS NOW. AS OF THIS WRITING, THERE WILL BE 32 UNITS AVAILABLE IN THE FIRST BATCH GOING OUT NOVEMBER 1ST ACCORDING TO KEN.

TO GET ON THE LIST, SIMPLY SEND AN EMAIL TO RICK@CLUB100.ORG. STATE HOW MANY YOU WISH. CONFIRMATION WILL BE SENT IMMEDIATELY.

DO NOT PAY FOR YOUR NADSBOX UNTIL ASKED. I WILL SEND A PAYPAL INVOICE WHEN I ABSOLUTELY KNOW THAT I CAN SHIP A NADSBOX TO YOU.

MODEL "T"S FOREVER...
-RICK-

NADSBOX

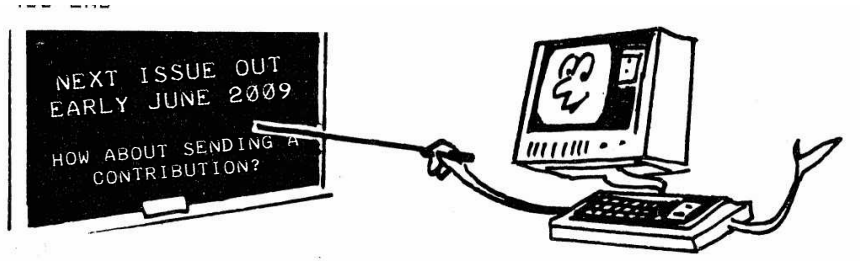
THROW YOUR TPDD/TPDD2 AWAY AND FORGET CABLES AND PROGRAMS FOR FILE TRANSFER. NEW AGE DIGITAL STORAGE BOX ... SOLID STATE STORAGE TO A MEDIA CARD WHICH CAN BE READ ON A PC! ... PLUGS INTO THE RS232 PORT AND IS ACCESSED BY TS-DOS JUST LIKE A DRIVE. THIS IS THE PRODUCT OF THE CENTURY FOR MODEL T COMPUTING. NADSBOXES ARE PRICED AT $195 EACH.
THE FIRST BATCH OF NADSBOXES ARE SOLD OUT. A SECOND BATCH IS IN PROCESS. TO GET ON THE WAITING LIST FOR A NADSBOX SEND AN EMAIL TO
RICK@CLUB100.ORG
NAMES ON THE 2ND BATCH LIST BEFORE 1/1/09 WILL GET A NADSBOX AT THE INTRODUCTORY PRICE EVEN IF IT IS SHIPPED AFTER THE FIRST.

--== o0o ==--

HERE'S A LITTLE ONE-LINER JUST FOR FUN! IT PUTS 255 RANDOM CHARACTERS IN THE FIRST 255 SPACE ON THE SCREEN AND PERFORMS A 'BUBBLE' SORT ON THEM. IN THE EARLY DAYS OF COMPUTING, BEFORE THE MODEL 1 AND 16K RULED, I RECALL, A BUBBLE SORT WAS ALSO REFERRED TO AS A 'FRIEND' SORT, AFTER THE GUY WHO, IN 1956, DESCRIBED IT AS A 'NEW' SORTING ALGORITHM.

```
1 DATA 14,0,221,33,0,60,6,254,221,126,0,221,86,1,186,40,10,56,8,
14,1,221,114,0,221,119,1,221,35,16,233,203,65,32,221,201:FORI=32
512TO32547:READA:POKEI,A:NEXTI:DEFUSR0=32512:CLS:FORI=15360TO156
14:A=RND(95)+32:POKEI,A:NEXTI:A=USR(0):RUN
```

```
                ;
                ;SCRNSORT
                ;BUBBLE SORTS THE FIRST 255 BYTES OF SCREEN
                ;
                          ORG   8000H
                          LOAD  8000H
8000 0E00       START:    LD    C,0         ;C IS THE CHANGE FLAG
8002 DD21003C             LD    IX,3C00H    ;START OF SCREEN ADDRESS
8006 06FE                 LD    B,254       ;# OF ENTRIES -1
8008 DD7E00     SORT01:   LD    A,(IX+0)    ;GET 1ST ENTRY
800B DD5601               LD    D,(IX+1)    ;GET NEXT CHARACTER
800E BA                   CP    D           ;COMPARE A TO D
800F 280A                 JR    Z,SORT02    ;JUMP IF EQUAL
8011 3808                 JR    C,SORT02    ;JUMP IF I < I+1
8013 0E01                 LD    C,1         ;SET SWAP FLAG
8015 DD7200               LD    (IX+0),D    ;SWAP CHARACTERS
8018 DD7701               LD    (IX+1),A    ;NEXT CHARACTER
801B DD23       SORT02:   INC   IX          ;SCREEN ADDRESS +1
801D 10E9                 DJNZ  SORT01      ;DO FOR 254 PAIRS
801F CB41                 BIT   0,C         ;TEST IF C=0 OR 1 FOR CHANGE
8021 20DD                 JR    NZ,START    ;SWAP - ANOTHER PASS
8023 C9                   RET               ;BACK TO BASIC
                          END
```



EXPANSION INTERFACE MEMORY

(NO NAME TO ATTRIBUTE)

ADDING MEMORY TO THE MODEL 1'S EXPANSION INTERFACE IS USUALLY QUITE STRAIGHT FORWARD. USE 4116 RAM CHIPS (AT LEAST 200 NS) AND PLACE THE CHIPS IN AN EMPTY SOCKET, ALIGNING THE NOTCH ON THE CHIP WITH THE ONE ON THE SOCKET.
A QUICK TEST TO SEE IF ALL'S OK IS TO PRINT MEM. THE ANSWER SHOULD BE ABOUT 48340 (FOR A FULL 48K MACHINE). IF YOU SUSPECT A PROBLEM AND DON'T HAVE A MEMORY TEST PROGRAM, THE ONE LINER BELOW MIGHT BE OF HELP.

SINCE EACH CHIP IS 1 BIT OF EVERY 8-BIT RAM BYTE, A NUMBER OTHER THAN ZERO TELL YOU WHICH CHIP IS BAD. CHIPS Z1-Z8 ARE ADDRESSES 49,152 TO 63,535. AS YOU CAN'T POKE OR PEEK OVER 32,767, YOU MUST USE NEGATIVE NUMBERS TO ACCESS RAM ABOVE 32,767; (IN THIS CASE - 16,384 TO -1).
CHIPS Z9-Z16 ARE ADDRESSES -32,768 TO -16,383. Z1 AND Z9 ARE BITS 7; Z2 AND Z10 ARE BITS 6; Z8 AND Z16 ARE BITS ZERO. THEREFORE, IF YOU GET A 3 IN THE ADDRESS RANGE -32,767 TO -16,384, YOU NEED TO REPLACE CHIPS Z15 AND Z16.

```
1 CLS:FORI=-32767TO-1:PRINT@350,I:POKEI,0:IFPEEK(I)<>0THENPRINT"
BAD ADDRESS ";I;"VALUE =";PEEK(I)ELSENEXT
```

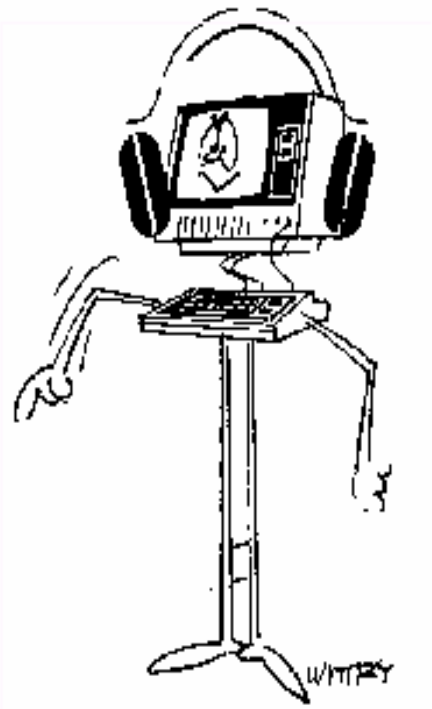
---== o0o ==---

CAMPANOLOGY - BELL RINGING ON A MODEL 1

OR A TIN(NY)TINNABULATION

BEING A TOTAL NOVICE AS REGARDS TO BELL RINGING, I JUST THOUGHT IT WOULD BE A NOVEL IDEA TO SEE IF IT WAS POSSIBLE TO 'RING BELLS' ON A MODEL 1. (I GET THESE MAD IDEAS FROM TIME TO TIME AS YOU KNOW, HENCE THE EAR MUFFS).

HAVING PLAYED AROUND WITH A SIMPLE SOUND GENERATION PROGRAM, ORIGINALLY WRITTEN BY HARDING BROTHERS, I THOUGHT AN 8-BELL PEAL, OF SORTS, WAS A POSSIBILITY. I ALSO FOUND THE FOLLOWING INTRODUCTION TO START THINGS OFF.



THE PURPOSE OF THESE BRIEF FEW WORDS, A PRéCIS OF AN ARTICLE BY MATTHEW SORELL, IS TO INTRODUCE NON BELL RINGERS, LIKE MYSELF, TO CHANGE-RINGING AND TO THE PERMUTATIVE STRUCTURE INHERENT IN CHANGE-RINGING. THE AIM IN CHANGE-RINGING MAY BE SUMMARISED AS RINGING A SET OF PERMUTATIONS, WITHOUT REPETITION OR PAUSE, ENTIRELY FROM MEMORY BY A BAND OF BELL RINGERS. BY ANALOGY, CONSIDER TAKING A RUBIK'S CUBE THROUGH AN ORDERED SEQUENCE OF PATTERNS; BUT WITH EIGHT PEOPLE LOOKING AFTER THE TASK IN SYNCHRONY. IT IS CLEAR THAT CHANGE-RINGING IS A UNIQUE MIX OF MUSIC, TEAMWORK AND MATHEMATICS.

CHANGE-RINGING HAS ITS BEGINNINGS AS EARLY AS THE FOURTEENTH CENTURY. AS MANY AS THREE BELLS WOULD HAVE BEEN IN A CHURCH STEEPLE, FORMING A COMBINATION CHURCH AND SECULAR PUBLIC ADDRESS SYSTEM. THE BELLS WERE NOT WELL TUNED AND WHEN RUNG TOGETHER WOULD HAVE BEEN CACOPHONOUS. THE BELL HANGERS ALSO FOUND THAT BY SWINGING A BELL, RATHER THAN HITTING IT WITH A CLAPPER, THE BELL WAS ABLE TO RING CLEARLY. OVER TIME, BELLS BEGAN TO BE SWUNG HIGHER AND HIGHER UNTIL EVENTUALLY A MECHANISM WAS DEVELOPED TO ALLOW BELLS TO BE SWUNG FULL-CIRCLE, AS THEY ARE TODAY.

FULL-CIRCLE RINGING HAS THE ADDITIONAL ADVANTAGE OF TIMING CONTROL. ONE CAN IMAGINE THE EARLY BELL RINGERS NOW RINGING THE BELLS IN A DESCENDING SCALE (THE POOR TUNING OF THE BELLS WOULD HAVE MADE CHORDS HIGHLY UNDESIRABLE), AND, ONE DAY, DECIDING TO CHANGE THE ORDER OF THE BELLS. AND FROM THERE, CHANGE-RINGING WAS BORN.

EACH BELL SWINGS FULL CIRCLE BUT DOES SO IN A FORWARD AND REVERSE DIRECTION. THIS IMPLIES THAT THERE ARE TWO STROKES

WHICH NEED TO BE CONSIDERED; ONE IN WHICH THE ROPE IS WRAPPED AROUND THE WHEEL (THE "HANDSTROKE") AND THE OTHER IN WHICH THE ROPE UNWRAPS (THE "BACKSTROKE"). THE TWO STROKES ALLOW BELL RINGERS TO FOLLOW WHAT IS HAPPENING BY WATCHING THE ROPES AROUND THEM ("ROPESIGHT") AS WELL AS LISTENING TO THE BELLS.

THE SYSTEM BY WHICH THE DIFFERENT UNIQUE ROWS CAN BE ORDERED WAS FORMALISED BY THE END OF THE SEVENTEENTH CENTURY. THE BELL MECHANISM HAS IMPROVED SIGNIFICANTLY, ESPECIALLY WITH THE INTRODUCTION OF STEEL FRAMES (INSTEAD OF OAK) AND ROLLER BEARINGS (INSTEAD OF PLAIN GREASED BEARINGS). THE NUMBER OF BELLS IN A TOWER HAS INCREASED TO A TYPICAL RING OF SIX OR EIGHT AND IN SOME CASES TO TWELVE. FINALLY, A SYSTEMATIC METHOD OF BELL TUNING WAS DEVELOPED AROUND 1900, BRINGING BELLS AND THEIR HARMONICS INTO STANDARD TUNING. ALAS, NOT SO READILY NOTICEABLE ON THE MODEL 1!

THE HEAVIEST BELL IN A RING MAY WEIGH ANYWHERE AROUND A TONNE (THE HEAVIEST RING, LIVERPOOL CATHEDRAL, HAS A FOUR TONNE TENOR).

THE LAST DECADE HAS SEEN NEW INNOVATIONS. RINGS OF UP TO 16 BELLS HAVE APPEARED FOR THE FIRST TIME. THE NUMBER OF METHODS AND PEAL COMPOSITIONS HAS EXPLODED, THANKS IN NO SMALL PART TO THE AVAILABILITY OF COMPUTER SOFTWARE TO ASSIST IN THE COMPOSITION PROCESS. IF YOU WISH TO KNOW MORE ABOUT BELL RINGING OR EVEN ABOUT THE 300-YEAR-OLD STEDMAN TRIPLES PROBLEM, THERE ARE NUMEROUS WEB-SITES AVAILABLE. A GOOD STARTING POINT IS

WWW.ABELSIM.CO.UK

WHICH HAS ONE OF THE MOST IMPRESSIVE AND VALUE FOR MONEY BELL RINGING SIMULATIONS AVAILABLE.

I HAVEN'T PLACED THE PROGRAMS ON THE WEB-SITE AS DOWNLOADS BECAUSE THEY ARE QUITE SMALL. IF YOU HAVE A PROBLEM HOWEVER, PLEASE EMAIL ME AND I'LL SORT SOMETHING OUT FOR YOU.

LINES 1 TO 4 LOAD THE MACHINE CODE PROGRAM INTO THE VARIABLE A$. N.B. NORMAL LISTING OF THE PROGRAM AFTER TYPING IN 'RUN' WILL BE CHANGED.

LINES 120-140 SET THE TONE FOR THE 8 BELLS AND THEIR NUMBER.

LINES 170-280 RUN THROUGH 2 LOOPS TO SET THE NECESSARY CHANGES TO THE ORDER OF THE BELLS.
LINE 190 SETS THE CHANGES IN ORDER IF J HAPPENS TO BE ODD OR EVEN AND ALLOWS FOR A SMALL DELAY TO ASSIST THE SOUND OF THE BELL, AND 200 ALLOWS FOR A VARIATION WHEN THE TREBLE IS AT THE LEAD OF THE ROW. (THE '*' MARKS IT).

LINE 310 IS A SUBROUTINE TO PRINT THE BELL ORDER ON THE SCREEN AND SEND TO CORRECT 'BELL SOUND' TO THE SPEAKER.

IN THE WORDS OF 'ALEXANDER MEERKAT' "SIMPLES".

DON'T HESITATE TO SEND IN ANY IMPROVEMENTS OR CHANGES (NO PUN INTENDED) YOU DEVISE.

```
 1 A$="  22 SPACES HERE  ":A=VARPTR(A$):B=PEEK(A+1)+PEEK(A+2)*256
 2 DEFUSR=B:FORC=BTOB+21:D=C:READE:POKED,E:NEXT
 3 DATA 205,127,10,76,69,62,1,211,255,16,254,69,62,16,211
 4 DATA 255,16,254,13,32,239,201


 100 REM   BELL RINGING - PLAIN BOB MAJOR
 110 REM       THE 1ST PLAIN COURSE
           112 OF THE 40320 PERMUTATIONS!

 120 A(1)=70:A(2)=74:A(3)=84:A(4)=96:A(5)=106:A(6)=114
 130 A(7)=128:A(8)=142:T=0:L1=0:L2=0
 140 FORZ=1TO8:B(Z)=Z:NEXT
 150 REM

 160 GOSUB310
 170 FORH=1TO7
 180    FORJ=1TO16
 190       IFJANDNOT-2 THEN L1=1:L2=7ELSEL1=2:L2=6:FORD=1TO150:NEXT
 200       IFJ=16L1=3:L2=7:PRINT"    *"
 210          FORX=L1TOL2STEP2
 220             T=B(X+1)
 230             B(X+1)=B(X)
 240             B(X)=T
 250          NEXTX
 260          GOSUB310
 270    NEXTJ
 280 NEXTH
 290
 300 END
 310      FORY=1TO8:PRINTB(Y);:C=USR(A(B(Y))):NEXTY:PRINT
 320 RETURN
```

```
1                   ; HARDING BROTHER'S
2                   ; SIMPLE RELOCATABLE SOUND ROUTINE
3                   ;
4                            ORG  8000H      ;ANYWHERE WILL DO
5                            LOAD 8000H      ;FIND IT QUICKLY HERE
6  8000 CD7F0A               CALL 0A7FH      ;GET VALUE FROM USR CALL
7  8003 4C                   LD   C,H        ;GET HIGH BYTE INTO C
8  8004 45         LOOP1:    LD   B,L        ;GET LOW BYTE INTO B
9  8005 3E01                 LD   A,01H      ;TURN ON OUTPUT BIT
10 8007 D3FF                 OUT  (0FFH),A   ;AND SEND TO PORT
11 8009 10FE       LOOP2:    DJNZ LOOP2      ;WAIT FOR HALF A CYCLE
12 800B 45                   LD   B,L        ;RELOAD WITH LOW BYTE
13 800C 3E10                 LD   A,10H      ;AND SEND TO PORT
14 800E 10FE       LOOP3:    DJNZ LOOP3      ;WAIT HALF A CYCLE
15 8010 0D                   DEC  C          ;DROP DURATION COUNT
16 8011 20F1                 JR   NZ,LOOP1   ;GO BACK UNTIL ZERO
17 8013 C9                   RET
18                           END
```

NEXT ISSUE DUE OUT, EARLY JUNE 2009

TRS8BIT



HI EVERYONE, AND WELCOME TO THIS, THE JUNE 2009 EDITION OF TRS8BIT.

FIRST, BOTH DEE AND I WOULD LIKE TO THANK EVERYONE WHO HAVE CONTACTED US OVER THE LAST 3 MONTHS. IT REALLY HAS BEEN SUPER TO HEAR FROM YOU ALL. PEOPLE FROM, QUITE LITERALLY, OVER THE WORLD HAVE EMAILED US WITH IDEAS, COMMENTS AND SUGGESTIONS (ALL OF WHICH HAVE BEEN COMPLIMENTARY!) FOR FUTURE ISSUES. I'LL TRY MY BEST TO FIND BITS OF INTEREST FOR YOU ALL. BUT PLEASE REMEMBER, IF YOU HAVE ANY NEWS OF TANDY RELATED INTEREST, PLEASE LET US ALL HEAR FROM YOU. THE ONLY WAY WE CAN CONTINUE THE SUCCESS AND MOMENTUM WE HAVE WITH TRS8BIT, IS, QUITE SIMPLY, TO CARRY ON USING IT!

THIS ISSUE IS HEAVILY BIASED TO NEWDOS80 AND ITS UPGRADES AND MODIFICATION. THERE'S NO PARTICULAR REASON FOR THIS, EVERYTHING SEEMED TO APPEAR ALL AT THE SAME TIME, STARTING WITH A RESPONSE TO MY QUESTION ASKING FOR INFORMATION ON NEWDDOS86.

SO, BEFORE I PROGRESS, AND JUST TO BALANCE THINGS OUT A BIT, AND FOR ALL YOU LDOS FANS, HAVE YOU CHECKED OUT TIM MANN'S WEB SITE?

HTTP://TIM-MANN.ORG/TRS80.HTML

THERE IS MASSES OF PUBLIC DOMAIN MATERIAL FOR ALL 'LDOS'ERS'. IT IS QUITE FANTASTIC. THERE'S A LINK ON OUR WEBSITE FOR YOU TO USE. A BIG THANK YOU MUST GO THE BOTH ROY SOLTOFF FOR PUTTING THE INFORMATION IN THE PUBLIC DOMAIN AND TO TIM FOR HOSTING IT.

A COUPLE OF ISSUES AGO I MENTIONED I CAME ACROSS A PART BOX OF 8 INCH DISKS DURING A HOUSE CLEARANCE I WAS HELPING ON. WELL, I'VE JUST FOUND 3 MORE DISKS, SO, IF YOU'RE STILL SHORT OF ANY, PLEASE LET ME KNOW ASAP AND I'LL SEND THEM OFF TO YOU.

BILL ALLEN HAD WRITTEN TO US ALL FROM AUSTRALIA WITH A MOST BRILLIANT ARTICLE ON HOW NEWDOS86 CAME TO BE WRITTEN BY WARWICK SANDS AND ITS FURTHER DEVELOPMENT INTO NEWDOS90.
THIS IS ALL NEW TO ME, WITH BILL KINDLY RESPONDING TO A REQUEST FOR INFORMATION ON THE WEBSITE. THE FIRST TIME I CAME ACROSS NEWDOS86 BEING MENTIONED WAS IN AN ADVERT IN NATGUG NEWS.

WARWICK SANDS HAS CONTACTED ME VIA EMAIL, AND HAS STATED THAT ALL NEWDOS86/90 PROGRAMS AND DOCUMENTATION WERE PLACED IN THE PUBLIC DOMAIN SOME TIME AGO.
MUCH OF THIS IS AVAILABLE FROM FRED JAN KRAAN'S SITE. (THERE'S A LINK FOR YOU TO USE ON OUR SITE). I BELIEVE BILL AND A COUPLE OF FRIENDS ARE ABOUT TO REVAMP THE MANUAL INTO A .TXT AND .PDF DOCUMENT AS THE EXISTING PDF FILE IS QUITE LARGE AND, ACCORDING TO BILL, THERE ARE NUMEROUS ERRORS AND TYPO'S TO CORRECT. AS SOON AS I CAN, I'LL PLACE THE NEW .PDF FILE FOR DOWNLOADING ON THE WEBSITE.

THERE'S A REVIEW AND SOME PHOTOS OF THE NADS BOX I CAME ACROSS ON THE WEB,

WRITTEN BY SHAUN WHEELER. HE SEEMS TO BE THRILLED TO BITS WITH IT.
NOW THAT THE DOLLAR/POUND VALUES ARE RETURNING TO SOMETHING LIKE SENSIBLE, IT IS BECOMING A MORE ATTRACTIVE BUY. HAS ANYONE IN THE UK GOT ONE YET?

THERE'S A COUPLE OF REVAMPED ARTICLES FROM NATGUG NEWS, THE FIRST BY KEN ROBINSON, FOR CHANGES TO THE VIDEO GENIE TO ALLOW THE ND8Ø V2 'BOOT' COMMAND TO WORK.
THE SECOND IS FROM DAVE WASHFORD, FONDLY REMEMBERED BY US ALL AS 'WASHY' ON HOW TO PERSONALIZE YOU ND8Ø DISKS

AND FINALLY ..

PETER STONE HAS COME UP WITH THE FIRST (OF MANY, I HOPE) INSTALMENT OF HIS RE-LEARN Z8Ø ASSEMBLER ARTICLE.
THIS IS A TREMENDOUS EFFORT ON PETER'S PART, AND I'M SURE I SPEAK FOR EVERYONE, WHEN I SAY THANKS PETER. I'M LOOKING FORWARD TO GETTING THE 'LITTLE GREY CELLS' WORKING AT Z8Ø AGAIN.

PETER HAS VERY KINDLY OFFERED TO LOOK AT SPECIFIC REQUEST FOR ARTICLES AND HAS PUT HIS EMAIL ADDRESS ON THE FINAL PAGE. THIS IS AN OPPORTUNITY NOT TO BE MISSED, SO PLEASE USE IT!

WELL, THAT ABOUT WRAPS-UP THIS EDITION. I HOPE YOU FIND IT OF INTEREST.
PLEASE LET US ALL KNOW HOW YOU GET ON.



TAKE CARE

DUSTY

AT THE
READY>
PROMPT

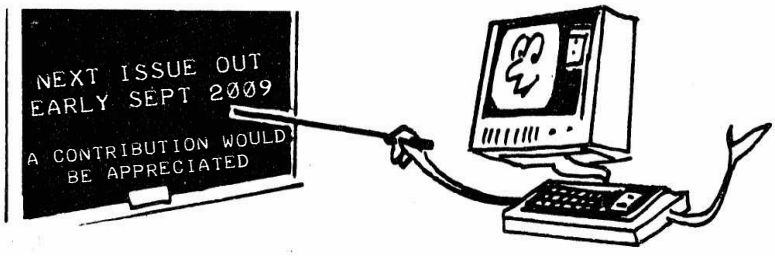


THANKS TO ALL WHO REMINDED ME THAT THE ASSEMBLER STATEMENT LD A,(IY) NEEDS THE +Ø INCLUDING, VIZ. LD A,(IY+Ø). I'D FORGOTTEN THAT IY AND IX FOR THAT MATTER, USUALLY DON'T POINT TO THEIR ACTUAL MEMORY ADDRESS, BUT ONES THAT ARE ROUND ABOUT. THEY NEED WAKING UP SO TO SPEAK!

HERE'S A LITTLE TIP WHICH I FIRST ENCOUNTERED AT ONE OF THE MANY WORKSHOPS I ATTENDED IN THE MID 1980'S. WHEN RUNNING VISICALC UNDER ND8Ø, IF, EVEN ON A 48K SYSTEM THERE'S NOT MUCH ROOM FOR YOUR SPREADSHEET, FIRST CHECK YOUR SYSTEM SETTING FOR AP AND MAKE SURE IT'S AP=Ø/ØH. THIS WAS QUITE OFTEN CHANGED TO SOMETHING LIKE 62128 WHEN USING SOUTHERN SOFTWARE'S EDIT PROGRAM.

MATTHEW REED HAS UPGRADED TRS32,THE TRS-8Ø MODEL I/III/4 EMULATOR FOR WINDOWS, TO VERSION 1.19. IN ADDITION TO A FEW BUG FIXES, THE PRIMARY ADDITION IS A NEW MODEL III, LEVEL I MODE THAT WORKS WITH THE ROM IMAGE RECENTLY EXTRACTED BY DAVID COOPER. IF YOU ARE INTERESTED, YOU CAN DOWNLOAD IT AT MATTHEW'S WEB SITE.

(THERE'S A LINK ON OUR WEBSITE FOR YOU TO USE-DUSTY)



NADS BOX REVIEW
BY
SHAUN WHEELER

MY OLD TRS-80 MODEL 100 (AKA "GIBRALTAR") JUST BECAME MORE USEFUL.

TODAY, AFTER A WEEK OF EDGE-OF-MY-SEAT ANTICIPATION, MY LATEST PERIPHERAL ARRIVED, ONE THAT WILL MAKE MY MOBILE BLOGGING LIFE A LOT EASIER.

MY NADSBOX ARRIVED.

"WHAT THE HELL IS A NADSBOX?" YOU MAY ASK, "WHAT DOES IT DO, APART FROM INVITE PUERILE COMMENTS AS TO ITS NAME?" (INCLUDING, I MIGHT ADD, AN OFFICIALLY SANCTIONED ONE...)

SO GOOD OF YOU TO ASK! THE NADSBOX, OR NEW AGE DIGITAL STORAGE BOX, IS A MODERN REPLACEMENT FOR THE AGING TANDY PORTABLE DISK DRIVE. IT BASICALLY EMULATES THE TPDD, USING SD, MMC, SDHC CARDS IN PLACE OF FLOPPY DISKS.

THE MAIN BENEFIT IS STORAGE CAPACITY - MY OLD TPDD2 COULD STORE 200K ON A SINGLE FLOPPY. THE NADSBOX CAN HANDLE MEMORY CARDS UP TO 16 GIGABYTES! NOT THAT I'D EVER NEED THAT KIND OF STORAGE, MIND YOU, BUT IT'S NICE TO KNOW THAT THE CAPABILITY IS THERE. HEH, I DOUBT THAT I'LL EVER FILL THE 512MB CARD THAT CAME WITH THE UNIT...

ANOTHER ADDED BENEFIT (AND THIS WAS THE DECIDING FACTOR FOR ME) IS THAT THE CARDS USED IN THE NADSBOX CAN BE READ IN ANY OLD CARD READER, SO I CAN BLOG WITH MY MODEL 100 TO MY HEART'S CONTENT, SAVE MY WORK TO THE NADSBOX, PULL THE SD CARD, POP IT INTO MY USB2 CARD READER AND TRANSFER ALL MY WORK ONTO MY LINUX BOX (OR IMAC) FOR UPLOADING TO BLOGGER.

OF COURSE, THIS WORKS BOTH WAYS, TOO. I CAN DOWNLOAD A BUNCH OF MODEL 100 APPS ONLINE USING MY LINUX BOX (ETC) AND SIMPLY WRITE EVERYTHING TO THE SD CARD FOR TRANSFER TO MY MODEL 100!



NO MORE FIGHTING WITH NULL-MODEM CABLES OR SLOOOOOWWWW MODEM TRANSFERS ANYMORE! WOOOHOOOO!

MY HEARTFELT THANKS TO KEN PETTIT FOR DESIGNING THE SINGULARLY MOST USEFUL COMPUTING PRODUCT I'VE BOUGHT IN THE PAST DECADE OR SO, AND A SPECIAL "THANK YOU" GOES TO RICK HANSON OF CLUB 100 FOR BEING THE SOLE DISTRIBUTOR AND ALL-AROUND NICE GUY.

I'D ALSO LIKE TO EXTEND A BIG "HELLO" TO ALL WHO HAVE COME TO MY LITTLE CORNER OF THE BLOGOSPHERE VIA THE LINK ON THE CLUB 100 SITE. WELCOME!

SO FAR, THE NADSBOX HAS PAID FOR ITSELF WITH AN INCREASE IN PRODUCTIVITY. I'VE BEEN WRITING A LOT MORE THAN USUAL, AND I'VE BEEN TRANSFERRING A BUNCH OF MY OTHER WORK FROM MY OLD TPDD DISKS ONTO THE NADSBOX, SO I CAN UPLOAD THEM TO MY LINUX BOX FOR LATER ARCHIVING ONTO CD OR DVD.



I'VE ALSO BEEN PORING OVER SOME OF THE MODEL 100'S TECHNICAL DOCUMENTS ONLINE, SO I CAN HAVE A BETTER UNDERSTANDING OF THE MACHINE AND HOW IT WORKS - BUT MORE ON THAT LATER. I HAVE TWENTY MORE DISKS WORTH OF MATERIAL TO ARCHIVE, TOMORROW'S BLOG POST TO FINISH, AND A SPREADSHEET TO UPDATE.

GONNA BE A LATE NIGHT...

--== oØo ==--

HERE'S A SHORT ARTICLE, ORIGINALLY WRITTEN BY KEN ROBINSON, TO PATCH ND80 V2 TO ALLOW CHANGES FOR THE VIDEO GENIE.
*** AS ALWAYS - DO NOT USE ON YOUR ONLY OR 'BEST' COPY OF ND80. MAKE A COPY FIRST AND WORK ONLY ON THAT! ***
(NO TEARS BEFORE BEDTIME PLEASE - DUSTY)

NEWDOS80 V2 - 'BOOT' FOR THE VIDEO GENIE 1

KEN ROBINSON

I HAVE INVESTIGATED THE CODE WITHIN NEWDOS80 V2 TO SEE IF I COULD ALTER THE PROGRAM SO THAT THE 'BOOT' COMMAND WORKED ON THE VIDEO GENIE 1.

HERE ARE THE ALTERATIONS-

NEWDOS80 V2
SYS0/SYS SEC09 BYTE 25H
CHANGE 76H TO C7H
SYS9/SYS SEC00 BYTE D3H
CHANGE 76H TO C7H

TRSDOS V2.1
BACKUP/CMD SEC06 BYTE 18H
CHANGE 76H TO C7H
FORMAT/CMD SEC05 BYTE 83H
CHANGE 76H TO C7H

SUPER UTILITY V3
SEC 27 BYTE 73H CHANGE 76H TO C7H
SEC 8F BYTE 3AH CHANGE 76H TO C7H
SEC 90 BYTE 30H CHANGE 76H TO C7H

TASMON
SEC 08 BYTE 4DH CHANGE 76H TO C7H

HERE'S A ONE LINER FROM NICK DILISI AND ANTHONY ABATE. NO NEED TO PRESS ANY KEYS ON THIS ONE, JUST ENJOY IT! THEY'VE CALLED IT 'RACING'. I LIKE THE 'ENDLESS' LOOP FORT=1TO1E9 AS APPOSED TO FORT=0TO1STEP0. 11 BYTES AS AGAINST 14! 3 BYTE SAVED, WELL DONE LADS.

```
1 P=26:C$="><":FORT=1TO1E9:Q=RND(11):IFT/4=INT(T/4)PRINT@916+Q,"
O":NEXTELSEA$="I              I":PRINT@980,A$:PRINT@1023,:PRINT@P,
C$:IFPEEK(15489+P)=79ORP>32ORP<21PRINT@P,"BOOM":FORW=1TO99:NEXT:
RUNELSEK=PEEK(14400):IFK=32P=P-1:NEXTELSEIFK=64P=P+1:NEXTELSENEX
T
```

PERSONALISING DISKS

BY

DAVID WASHFORD

FROM AN ORIGINAL ARTICLE FIRST PUBLISHED IN NATGUG NEWS, SEPT '82

I HAVE BEEN ASKED BOTH AT LUTON AND IN THREE PHONE CALLS SINCE, HOW I MANAGED TO PERSONALISE MY NEWDOS DISKS. SO FOR ALL THOSE WHO STILL HAVEN'T VENTURED INTO SUPERZAP HERE GOES.
I HAVE ZAPPED MY DISKS SO THAT, 1, THE BANNER INCORPORATES THE LATEST ZAP (PATCH) NUMBER THAT HAS BEEN APPLIED. 2, SO THAT THE DISK TITLES AND THE DOS READY SIGNS ARE PERSONALIZED.

FIRST OF ALL, MAKE A BACKUP COPY! PUT IT IN DRIVE Ø. AND TAKE THE DISKS OUT OF ANY OTHER DRIVES THAT YOU HAVE.
THIS ISN'T REALLY NECESSARY, BUT IF YOU DON'T KNOW HOW TO PUT YOUR NAME ON A DISK, THEN IT FOLLOWS THAT YOU DON'T YET KNOW HOW TO BE SURE THAT YOU DON'T WIPE OUT A GOOD DISK.
LOAD SUPERZAP, PRESS <ENTER> AND TYPE IN Ø,17 (THE COMMA IS IMPORTANT). THE SCREEN IS NOW FILLED WITH THE HEX AND ASCII DETAILS OF SECTOR 17, DOWN THE BOTTOM ON THE RIGHT YOU WILL SEE THE NAME APPARAT,. INC.
TYPE IN M O (ALPHA) D F2 AND THE CURSOR IS RAPIDLY WINKING AT YOU OVER A NUMBER 41. 41 IS THE HEX CODE FOR LETTER A, AND USING HEX CODES YOU CAN NOW TYPE IN WHAT YOU LIKE IN THOSE 12 SPACES - AND IF YOU ONLY WANT TO USE 6 LETTERS THEN FOLLOW THEM WITH 2Ø (THE HEX CODE FOR SPACE), SIX TIMES. AS YOU TYPE IN NEW NUMBERS YOU WILL SEE THE NAME CHANGE, IF YOU MAKE A MISTAKE IT IS OK TO USE THE BACKSPACE ARROW TO RETYPE THE CORRECT ENTRY. WHEN YOU ARE SATISFIED WITH YOUR EFFORTS, PRESS <ENTER> AND YOU WILL SEE THE MESSAGE 'PRESS Y IF OK TO WRITE TO DISK' - MAKE SURE YOU HAVEN'T GOT A WRITE-PROTECT TAB ON THE DISK BEFORE PRESSING Y.

THE SCREEN NOW SHOWS THE PREVIOUS PICTURE OF HEX AND ASCII CODES; PRESS X AND YOU WILL RETURN TO THE SUPERZAP MENU. PRESS <ENTER>, FOLLOWED BY Ø,18 <ENTER> AND YOU WILL SEE A SIMILAR PICTURE. ON THE RIGHT YOU WILL SEE THE MESSAGE 'VERSION 2.Ø' ,AND LOWER DOWN (SPREAD OVER TWO LINES) 'MODEL.I'. I CHANGED MY MESSAGE TO READ 'MOD 1 VN2,Ø' AND 'ZAP Ø79' RESPECTIVELY; TO CHANGE THE FIRST ONE YOU TYPE IN MOD33 AND THE CURSOR WILL BE OVER THE 56.

ENTER UP TO ELEVEN HEX CODES, USING 2Ø(SPACE) WHERE NECESSARY. IF YOU ARE CONFIDENT ENOUGH YOU CAN NOW USE THE ARROWS TO MOVE THE CURSOR DOWN TO THE SEVENTH ROW, OVER THE 4 OF THE LAST ENTRY ON THE ROW (4D) AND CHANGE THESE SEVEN LETTERS. IF YOU ARE NOT SURE OF THIS, THEN JUST PRESS ENTER AND Y AGAIN TO PUT THE FIRST CHANGE TO DISK, AND WHEN THE SCREEN COMES BACK TYPE IN MOD6E - THIS WILL PUT THE CURSOR OVER THAT 4.

WHEN THESE ARE DONE, PRESS X TO RETURN TO THE SUPERZAP MENU. PRESS <ENTER> AND TYPE IN Ø,164 <ENTER>. BY NOW YOU SHOULD KNOW WHAT TO LOOK FOR! DOWN ON THE RIGHT YOU

CAN SEE THE LETTERS LTO.MINI-NEWDOS/80.READY. THIS NEWDOS/80 READY IS THE MESSAGE THAT APPEARS WHEN YOU REBOOT, SO IF YOU WANT TO SEE 'GET A MOVE ON' OR 'JOHNNY IS READY' THEN THIS IS THE BIT TO CHARGE. (I MUST SAY THAT I WAS IMPRESSED BY SOMEONE'S MACHINE AT LUTON - HIS MESSAGE SIMPLY, BUT USEFULLY SAID 'YOU ARE IN DOS'). SO TYPE IN MODC9 AND THE CURSOR WILL JUMP TO OVER THE 4 OF 4E. NOW TYPE IN THE PHRASE OF YOUR CHOICE, TAKING CARE NOT TO GO ANY FURTHER THAN THE Y IN READY. PRESS <ENTER>, ENTER Y, PRESS X AND YOU ARE BACK AGAIN AT THE SUPERZAP MENU.

SO FAR WE HAVE CHANGED SCREEN MESSAGES. THE LAST TWO ALTERATIONS ARE FOR DISK NAMES. PRESS <ENTER>, TYPE IN 0,170 <ENTER> AND YOU WILL SEE AN ALMOST EMPTY RIGHT HAND SIDE OF THE SCREEN. DOWN THE BOTTOM IS NEWDOS80MM/DD/YY AND WHILST THE DATE IS THAT OF THE DISK CREATION, THE NAME IS THAT OF THE DISK. TO CHANGE IT TYPE IN MODD0 AND ENTER UP TO EIGHT HEX CODES FOR THE NAME YOU WANT, <ENTER>, Y,X BACK TO THE MENU. <ENTER>, 0,217 <ENTER> WILL SHOW US THE NAME USED WHEN FORMATTING DISKS; ON THE FOURTH LINE AND CONTINUING INTO THE FIFTH IS THE LEGEND 'BNOTNAMED2:/4:/::. AGAIN, TYPE MOD, FOLLOWED BY 3D, AND THE CURSOR IS OVER THE 4 OF 4E - NOT THE 4 OF 42, YOU DON'T CHANCE THE B! TYPE IN EIGHT HEX CODES FOR WHATEVER YOU WANT YOUR DATA DISKS TO BE CALLED, PRESS <ENTER>, Y. PRESS X TO RETURN TO THE MENU, TYPE EXIT TO RETURN TO DOS, AND IT'S ALL DONE!

--== o0o ==--

HISTORY OF NEWDOS86/90

BY BILL ALLEN

THE HISTORY OF WARWICK SANDS' WORK ON THE TRS-80 STARTS WITH MODIFYING THE DISK OPERATING SYSTEM OF NEWDOS-80. WE BELONGED TO A COMPUTER CLUB DEVOTED TO TANDY'S TRS-80 AND DICK SMITH'S SYSTEM 80 (NAMED GENIE IN EUROPE, I BELIEVE). THE NAME OF THE CLUB WAS TRS-80 USERS GROUP, BASED IN BRISBANE, AUSTRALIA. FROM ABOUT 1984 TO THE EARLY 90S, I WAS EDITOR OF ITS MONTHLY NEWSLETTER (USUALLY ABOUT 12 OR MORE A4 PAGES AND WROTE MANY OF THE ARTICLES IT CONTAINED -- SHARING INFORMATION ABOUT THESE COMPUTERS AND PROGRAMMING THEM AS INFO WAS DISCOVERED.

ANYHOW, SOME MEMBERS ASKED WHY THEIR TAPE MACHINES WITH LEVEL 2 BASIC COULDN'T USE THE ENHANCEMENTS IN THE DISK BASIC IN THE NEWDOS-80 DOS (IT WAS A FAIRLY EXPENSIVE TO CONVERT FROM TAPE TO DISK IN THOSE DAYS). WARWICK PUT UP HIS HAND AND SAID HE WOULD HAVE A GO AT DOING WHAT THE MEMBERS WISHED FOR AND CAME UP WITH WHAT WAS CALLED THE TOS (TAPE OPERATING SYSTEM). EVENTUALLY AFTER SEVERAL FURTHER ENHANCEMENTS IT WAS NAMED CUSTOM BASIC, WHICH BY THEN REPLACED NEWDOS-80'S DISK BASIC. IT HAD MANY EXTRA PROCEDURES BORROWED FROM IDEAS IN PASCAL AND OTHER LANGUAGES AND WORKED VERY WELL.

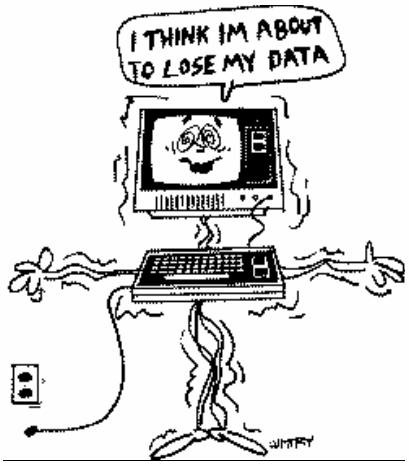
THEN WARWICK DECIDED TO TRY HIS HAND AT ENHANCING THE DISK OPERATING SYSTEM AS WELL, AS APPARAT, THE AUTHORS OF NEWDOS-80 HAD RETIRED FROM THE FIELD. AS TRSDOS WAS HOPELESS AND MOST CLUB MEMBERS WERE ALREADY USING NEWDOS-80, IT BECAME THE BASE ON WHICH HE BUILT AND THE DOS BECAME NEWDOS86 AND INCORPORATED CUSTOM BASIC WITH EVEN MORE ENHANCEMENTS. ONE OF ITS NICE FEATURES ABOUT THIS DOS WAS IT COULD READ JUST ABOUT ANY FORMAT OF DISK PROVIDED IT PHYSICALLY SUITED THE DRIVE. IF YOU HAD A DOUBLE SIDED 80-TRACKER DRIVE AVAILABLE, NO PROBLEM (SINGLE DENSITY, DOUBLE DENSITY, SINGLE SIDED, DOUBLE SIDED, 40 OR 80 TRACKS WERE ALL FAIR GAME). MOST OTHER DOS FORMATS FOR TRS-80'S COULD BE READ AS WELL.

WARWICK IS A MASTER CODER AND TAUGHT ME A LOT ABOUT ASSEMBLER LANGUAGE. HE HAD A SIMPLE METHOD OF WRITING CODE WHICH COULD NEVER CRASH THE MACHINE BY TESTING FOR AN ERROR (1 BYTE) AFTER EACH PROCEDURE. IF A PROCEDURE FAILED FOR ANY REASON, IT WOULD BACK OUT CLEANLY EVEN IF IT HAD TO GO TO THE START OF THINGS, BUT NEVER NEEDING A RESET OR SUFFER A 'BLUE SCREEN' TYPE OF FAILURE. VERY FEW BUGS WERE EVER EVIDENT AND ANY FOUND WERE QUICKLY CURED. IF ANYONE SUGGESTED SOMETHING WOULD BE NICE IF ADDED (AND THAT 'ANYONE' WAS, MORE OFTEN THAN NOT, YOURS TRULY), THE MODIFICATION WOULD TURN UP AT THE NEXT MONTHLY MEETING. SO, NEWDOS86 WAS IN CONTINUOUS DEVELOPMENT FROM ITS RELEASE IN 1986 UNTIL IT WAS ALTERED SO MUCH THAT HE RENAMED IT NEWDOS90. THE FINAL DEVELOPMENT WAS WHEN I CONNECTED A HARD DRIVE TO MY 4P AND HE WROTE CODE TO HANDLE THE I/O OF THE HARD DRIVE SO IT COULD REAP THE BENEFITS OF NEWDOS90. NEEDLESS TO SAY, I WROTE A GOOD MANY BASIC PROGRAMS, INCLUDING A DATABASE PROGRAM WITH ND86-90. I HAD SOME PARTICIPATION IN WRITING ASSEMBLER CODE ROUTINES FOR WARWICK TO INCORPORATE IN HIS SYSTEM. HE USUALLY GOT ME TO DO THE ONES THAT WERE MAINLY BASED ON COMPLEX LOGIC PROCESSES.

ALL IN ALL IT WAS A GREAT ADVENTURE IN THOSE EARLY DAYS OF PERSONAL COMPUTING. I STARTED IN 1982 WITH A SYSTEM 80 (TAPE) AND SOON GOT INTO MACHINE CODE (WHAT WERE ALL THOSE NUMBERS ABOUT?), BUT SOON WORKED OUT ENOUGH TO FIND LITTLE HOLES IN RESERVED MEMORY TO SPLIT UP A LITTLE PROGRAM NAMED TCOPY INTO MODULES WITH JRS AND MADE IT FIT INTO THE HIDDEN BITS OF RESERVED MEMORY SO I COULD COPY TAPES BEHIND THE SYSTEM'S BACK WITHOUT WALKING OVER ANYTHING. THIS WAS WHAT WE ALL HAD TO DO, BECAUSE INCREDIBLY IT WAS ALMOST IMPOSSIBLE TO BUY GENUINE TAPES AS TANDY NEVER STOCKED MORE THAN A FEW. THEY BUILT MARVELLOUS MACHINES FOR THAT TIME BUT HAD NO IDEA HOW TO MARKET THEM AND EVENTUALLY WERE DROWNED IN THE RUSH TO PCS. EVEN THEIR FIRST PCS WERE LOTS BETTER THAN THE EARLY IBM'S, BUT THE 'BIG BLUE' BEAT THEM IN DISTRIBUTION AND PUBLICITY.

I HAVE AN EMULATOR INSTALLED ON MY OLDEST PC (AMD ATHLON 2800+), BUT I DON'T REMEMBER OFFHAND WHO IS THE AUTHOR OF THE EMULATOR, AS I HAVEN'T RUN IT SINCE I INSTALLED IT SEVERAL YEARS AGO.

I HAVE A PROJECT TO RESCUE A FAMILY TREE I GENERATED IN THE EARLY 90S WITH GCRAB (RUN IN LDOS) AND IT HAD 2500 PEOPLE RECORDED



I STILL HAVE THE DISKS HOLDING THESE RECORDS -- ONE WITH THE LDOS OPERATING SYSTEM AND THE GCRAB PROGRAM AND THE SECOND DISK HOLDING THE DATA FILES. THESE ARE 80-TRACK DOUBLE SIDED DISKS. IT TOOK SUCH A SIZE DISK TO HOLD THE DATA FOR 2500 PEOPLE -- IT'S JUST ABOUT FULL. I RECENTLY INSTALLED A B: DRIVE (1.2 MB 5.25 IN.) ON THIS PC MACHINE, WITH THE IDEA OF COPYING THESE DISKS TO THE VIRTUAL DRIVES GENERATED BY THE EMULATOR AND VIEWING AND EXTRACTING THE DATA FROM THERE.

I'LL NEED LOTS OF LUCK TO BE ABLE TO READ THE DISKS. I DON'T KNOW WHETHER THE ENHANCED NEWDOS86 OR NEWDOS90 DISKS ARE AROUND STILL, BUT I NOTICED I HAVE UTILITY DISKS BELONGING TO ND90.
I STILL HAVE THE 4P DOWNSTAIRS, WHICH MAY STILL WORK, BUT I WOULD HAVE TO REMOVE THE FLOPPY DRIVES AND THOROUGHLY CLEAN THEM BEFORE I DARE INSERT A DISK TO READ. THIS MACHINE IS ALSO CONNECTED TO A 40 MB SCSI DRIVE VIA A SCSI CARD MADE BY ROY SOLTOFF. IT ALSO HAS ND90 AS ITS OPERATING SYSTEM, BUT I DON'T KNOW IF I REMEMBER HOW TO BOOT THAT SYSTEM UP. THE BOOT UP DISK IS THERE, BUT I DON'T KNOW WHETHER I CAN REMEMBER HOW. WARWICK DIDN'T FINALISE AN AUTOMATED BOOT METHOD BEFORE HE LEFT US TO GO TO A JOB IN AMERICA AND ABOUT 2 YEARS LATER, I STARTED INTO PCS WITH A 486 SX AND PROCEEDED THROUGH A SUCCESSION OF PC COMPUTERS EVER SINCE. I STILL HAVE A GENUINE ORIGINAL NEWDOS-80 DISK AND MANUAL AND A GENUINE ORIGINAL LDOS DISK (BY MISOSYS) AMONGST OTHERS.

THERE IS A FORMER MEMBER OF THE TRS-80 CLUB (NICK MARENTES) WHO WAS RUNNING A TANDY COMPUTER MUSEUM SOMEWHERE IN BRISBANE. SOME OF US DONATED A LOT OF GEAR TO THIS PROJECT. I DON'T KNOW IF IT STILL EXISTS, AS I HAVEN'T HAD CONTACT WITH HIM FOR SOME YEARS. I ALSO HAVE A TANDY DAISYWHEEL PRINTER SITTING ON A SHELF DOWN BELOW IN THE DUNGEON. TAPES WOULD BE A PROBLEM IF IT STILL WORKS. I BOUGHT A LOT OF STUFF CHEAP WHEN THE LOCAL TANDY SHOP SOLD OUT ALL ITS TRS-80 STOCKS AROUND 1990.

--== o0o ==--

ASSEMBLY LANGUAGE MADE EASY?

BY PETER STONE

I LOOKED AT DUSTY'S WEBSITE IN MARCH & NOTICED HIS PLEA FOR SOMEONE TO WRITE A SERIES OF ARTICLES ON MODEL 1 ASSEMBLY LANGUAGE PROGRAMMING. I WILL ADMIT THAT I HAD SOME HESITATIONS IN SAYING I WOULD ATTEMPT A SERIES OF ARTICLES, AS I DIDN'T KNOW WHAT LEVEL EVERYONE IS AT, & ALSO I HAD SOME CONCERNS OVER MY OWN COMPETENCE. HOWEVER I TOOK THE PLUNGE, & THIS IS MY FIRST ATTEMPT - HOPE YOU LIKE IT. I'M TRYING TO DEVELOP A SERIES OF SUBROUTINES THAT CAN BE PUT TOGETHER TO MAKE LARGER PROGRAMS. THE FIRST IS A SUBROUTINE TO DISPLAY A MESSAGE ON THE SCREEN.
ALL THE BOOKS ON BEGINNING PROGRAMMING THAT I'VE SEEN, ALWAYS SEEM TO START OFF WITH A PROGRAM THAT PRINTS HELLO WORLD ON THE SCREEN. SO, I'VE DECIDED NOT TO BUCK THE TREND, & DO SOMETHING SIMILAR. THE WAY I TEND TO PROGRAM IN ASSEMBLER, IS TO WRITE A BASIC PROGRAM FIRST, THEN WRITE THE ASSEMBLER VERSION NEXT. THE MAIN REASON FOR THIS IS THAT BASIC IS A VERY QUICK, IF SOMEWHAT SLOW, (EXECUTION WISE), WAY TO WRITE & DEBUG A PROGRAM. THE OUTLINE OF THE LEVEL 2 BASIC PROGRAM IS SHOWN BELOW.

```
10 CLS
20 LET B=474
40 GOSUB 100
30 END
100 PRINT @ B, A$
110 RETURN
120 A$="HELLO WORLD"
```

I WON'T INSULT ANYONE'S INTELLIGENCE BY EXPLAINING HOW IT WORKS. I'VE WRITTEN IT THIS WAY SO THAT HOPEFULLY IT WILL BE EASY TO CORRELATE HOW THE TWO PROGRAMS EQUATE TO EACH OTHER. THE ASSEMBLY LANGUAGE VERSION IS SHOWN NEXT.

```
        ORG 7000H           ;SEEMS LIKE A GOOD PLACE TO START
INKEY   EQU 002BH           ;GET SINGLE CHAR, NO WAIT
INPUT   EQU 0049H           ;GET SINGLE CHAR, WAIT FOR IT
LININP  EQU 0040H           ;GET LINE INPUT, CR TERMINATES
DISP    EQU 0033H           ;DISPLAY SUBROUTINE
CLRSCR  EQU 01C9H           ;CLS
TXTPRT  EQU 28A7H           ;PRINT TEXT TO SCREEN
DOSEXT  EQU 402DH           ;EXIT TO DOS
START   CALL    CLRSCR      ;CLEAR THE SCREEN
        LD  BC,3DDAH        ;LOAD SCREEN VALUE OF 474
        LD  (CURLOC),BC     ;STORE VALUE OF 474
        LD  HL,MESS1        ;GET START OF MESSAGE
        CALL    DSPMSG      ;LET'S PRINT IT
        NOP
        NOP
        JP  DOSEXT          ;QUIT PROGRAM
;
;*** DISPLAY MESSAGE SUBROUTINE       ***
;*** TERMINATES WITH A NULL CHARACTER ***
;*** ON ENTRY (BC) = CURSOR POSITION  ***
;***          (HL) = MESSAGE LOCATION ***
;
```

```
;DSPMSG PUSH    AF          ;SAVE REGISTERS
       PUSH    BC
       PUSH    HL
       LD  BC,(CURLOC)     ;GET 1ST PRINT LOCATION
 DSPØ1 LD  A,(HL)          ;GET MESS CHAR
       OR  A               ;TEST FOR ZERO
       JR  Z,DSPØ2         ;RETURN IF DONE
       LD  (BC),A          ;STORE CHARACTER
       INC BC              ;BUMP SCREEN POINTER
       INC HL              ;BUMP MESSAGE POINTER
       LD  (CURLOC),BC     ;SAVE POINTER
       CALL    DSPØ1       ;CONTINUE
 DSPØ2 POP HL              ;RESTORE REGISTERS
       POP BC
       POP AF
       RET
;*** MESSAGES STORED HERE ***
MESS1  DEFM    'HELLO WORLD'
       DEFB    ØH
;*** VARIABLES STORED HERE ***
CURLOC DEFW    3CØØH            ;STORE CURSOR LOCATION
       END START       ;FINISH
```

IF WE BREAK DOWN THE LISTING ABOVE INTO SECTIONS, WE GET THE FOLLOWING. THE FIRST SECTION (SHOWN BELOW), LISTS VARIOUS CALLS TO SUBROUTINES WITHIN THE ROM & OPERATING SYSTEM. I'VE CHOSEN 7ØØØH AS THE STARTING POINT AS A CONVENIENT NUMBER. I'LL DISCUSS VARIOUS VALUES TO USE HERE IN A LATER ARTICLE.

```
       ORG 7ØØØH           ;SEEMS LIKE A GOOD PLACE TO START
INKEY  EQU ØØ2BH           ;GET SINGLE CHAR, NO WAIT
INPUT  EQU ØØ49H           ;GET SINGLE CHAR, WAIT FOR IT
LININP EQU ØØ4ØH           ;GET LINE INPUT, CR TERMINATES
DISP   EQU ØØ33H           ;DISPLAY SUBROUTINE
CLRSCR EQU Ø1C9H           ;CLS
TXTPRT EQU 28A7H           ;PRINT TEXT TO SCREEN
DOSEXT EQU 4ØØDH           ;EXIT TO DOS
```

A WORD AT THIS POINT ABOUT GOOD PROGRAMMING PRACTICE WOULDN'T GO AMISS. FOR A SMALL PROGRAM LIKE THIS IT'S NOT REALLY WORTH USING EQUATES, AS THE VALUES COULD BE ENTERED DIRECTLY INTO THE VARIOUS LINES IN THE PROGRAM. HOWEVER, IN A LARGER PROGRAM OR ONE FOR A DIFFERENT COMPUTER, (IE A MODEL 3), THE ONLY VIABLE WAY OF CUSTOMISING THE PROGRAM IS TO SIMPLY CHANGE THE VALUES IN THE EQUATES & RE-ASSEMBLE THE PROGRAM. PLUS THE LABELS USED INDICATE THE FUNCTION THAT IS DONE WHEN THE CALL IS MADE, WHICH GREATLY AIDS DEBUGGING. A FURTHER, PERHAPS MORE IMPORTANT REASON FOR USING EQUATES IS THAT IT MAKES THE CODE EASILY RE-LOCATABLE.

BEFORE WE GET TOO FAR INTO THE PROGRAM, THE VARIOUS FIELDS IN A LINE OF SOURCE CODE NEED TO BE EXPLAINED. THE VARIOUS FIELD NAMES ARE SHOWN BELOW, WITH A TYPICAL LINE OF SOURCE CODE SHOWN BELOW IT FOR COMPARISON.

```
LABEL   OP-CODE OPERAND COMMENTS

INKEY   EQU     ØØ2BH   ;GET SINGLE CHAR, NO WAIT
```

THE LINES OF SOURCE CODE ARE IN A FREE FORMAT. THAT IS THE FIELDS DO NOT HAVE TO START AT SPECIFIC PLACES. ALL THAT IS NEEDED IS TO PUT A SINGLE SPACE BETWEEN THE VARIOUS FIELDS, THAT AN OP-CODE DOES NOT START IN COLUMN 1 & THAT A SEMICOLON PRECEDES A COMMENT. HOWEVER, USUALLY WE USE THE TAB KEY TO SPACE THINGS OUT AS THIS MAKES READING THE SOURCE CODE & ERROR CHECKING EASIER.

I'VE USED A NUMBER OF CALLS TO ROM & DOS SUBROUTINES & I'LL NOW EXPLAIN WHAT EACH ONE DOES.

INKEY (ØØ2BH) SCANS THE KEYBOARD & RETURNS ZERO IN REGISTER A IF NO KEY IS PRESSED, ELSE RETURNS THE CHARACTER. USES AF & DE.

INPUT (ØØ49H) SCANS THE KEYBOARD & WAITS FOR A KEY TO BE PRESSED. RETURNS CHARACTER IN A. USES AF & DE.

LININP (ØØ4ØH) ACCEPTS AN ENTIRE LINE OF INPUT TERMINATED BY ENTER OR BREAK. DISPLAYS THE CHARACTERS TYPED, RECOGNISING CONTROL FUNCTIONS (BACKSPACE ETC.). WHEN CALLED HL POINTS TO A BUFFER WHERE THE TEXT IS TO BE PLACED, B EQUALS THE MAXIMUM NUMBER OF CHARACTERS IN THE LINE. ON EXIT, B = THE NUMBER OF CHARACTERS TYPED, INCLUDING THE TERMINATOR. C IS SET IF THE LINE IS TERMINATED WITH BREAK. USES AF, DE.

DISP (ØØ33H) PRINTS THE ASCII CHARACTER IN A AT THE CURRENT CURSOR POSITION ON THE VDU. TH CURRENT CURSOR LOCATION IS STORED AT 4Ø2ØH. USES AF, DE, IY.

CLRSCR (Ø1C9H) CLEARS THE SCREEN & HOMES THE CURSOR TO THE TOP LEFT HAND CORNER OF THE SCREEN. USES AF.

TXTPRT (28A7H) PRINTS ALL THE TEXT POINTED TO BY HL UP TO A CARRIAGE RETURN, (ENTER KEY = ØDH), OR NULL (ØØ) AT THE CURRENT CURSOR POSITION. USES HL, AF. HOWEVER, AS MICROSOFT BASIC DECODED DRYLY STATES, .."IT (THIS SUBROUTINE) SHOULD NOT BE CALLED IF THE COMMUNICATIONS REGION & THE STRING AREA ARE NOT PROPERLY MAINTAINED".

DOSEXT (4Ø2DH) NORMAL EXIT TO DOS.
I TENDED TO KEEP THESE (& ANY I ADDED) INTO A FILE & SAVE IT TO A DISK, THEN EVERY TIME I STARTED A NEW PROGRAM I WOULD LOAD THIS FILE FIRST & CARRY ON FROM THERE, (REMEMBERING TO SAVE THE FINAL FILE WITH A DIFFERENT FILENAME, OF COURSE, WHEN I'VE FINISHED). THE EDAS DISK HAD DONE THIS ALREADY BY HAVING THE FILES EQUATES1.EQU & EQUATES3.EQU ON IT. HOWEVER, THEY SEEM TO BE MISSING FROM THE VERSION THAT IS AVAILABLE ON THE INTERNET.

THE NEXT AREA OF THE PROGRAM I WANT TO LOOK AT IS THE BOTTOM PART, AS THERE ARE SOME THINGS I'VE DONE HERE WHICH ARE WORTH EXPLAINING.

```
;*** MESSAGES STORED HERE ***
MESS1   DEFM    'HELLO WORLD'
        DEFB    ØH
;*** VARIABLES STORED HERE ***
CURLOC  DEFW    3CØØH           ;STORE CURSOR LOCATION
        END START               ;FINISH
```

I'VE CREATED TWO AREAS THAT AGAIN REALLY AREN'T NEEDED IN SUCH A SHORT EXAMPLE, BUT COME INTO THEIR OWN WITH LARGER PROGRAMS & ALSO FIT IN WITH WHAT I AM ATTEMPTING TO ACHIEVE WITH THIS COLUMN. THE FIRST IS AN AREA TO STORE MESSAGES IN. IT USES THE DEFM (DEFINE MESSAGE) PSEUDO-OP TO DEFINE A MESSAGE (OBVIOUSLY), & THE MESSAGE IS TERMINATED BY A NULL CHARACTER USING THE DEFB (DEFINE BYTE) PSEUDO-OP. THE NEXT AREA IS ONE TO STORE PROGRAM VARIABLES IN. HERE I'M USING THE DEFW (DEFINE WORD) PSEUDO-OP.
PSEUDO-OPS ARE NOT Z8Ø INSTRUCTIONS BUT ARE ASSEMBLER DIRECTIVES. WE HAVE ALREADY SEEN SOME OF THEM, & I'LL EXPLAIN THEM HERE.

ORG NN - SETS THE ADDRESS REFERENCE COUNTER TO THE VALUE NN (ORG 7ØØØH)

EQU NN - SETS THE VALUE OF A LABEL TO NN, IT CAN ONLY OCCUR ONCE IN ANY PROGRAM (EQU ØØ2BH)

END SIGNIFIES THE END OF THE SOURCE PROGRAM. ANY FOLLOWING STATEMENTS ARE IGNORED (END START)

DEFB N - DEFINES THE CONTENTS OF A BYTE TO BE N (DEFB ØØH OR DEFB FFH)

DEFW NN - DEFINES THE CONTENTS OF A TWO BYTE WORD NN (DEFW 3CØØH)

DEFM 'S' - DEFINES AN AREA OF MEMORY THAT CONTAINS THE ASCII REPRESENTATION OF STRING 'S'. THE LENGTH OF STRING 'S' MUST BE IN THE RANGE OF Ø > N < 63

THE VERY LAST LINE INDICATES THE END OF THE PROGRAM.

MOVING ON TO THE PROGRAM, LET'S LOOK AT THE FIRST PART.

```
START   CALL    CLRSCR          ;CLEAR THE SCREEN
        LD  BC,3DDAH            ;LOAD SCREEN VALUE OF 474
        LD  (CURLOC),BC         ;STORE VALUE OF 474
        LD  HL,MESS1            ;GET START OF MESSAGE
        CALL    DSPMSG          ;LET'S PRINT IT
        NOP
        NOP
        JP  DOSEXT              ;QUIT PROGRAM
```

THE FIRST LINE HAS THE LABEL START, (OBVIOUSLY, AS IT'S THE START OF THE PROGRAM :)). IF YOU USE A LABEL HERE, THEN THE END STATEMENT HAS TO INCLUDE THE LABEL. I'VE ALWAYS BEEN TAUGHT THAT THIS IS GOOD PROGRAMMING PRACTICE, & SO PASS IT ON. THIS LINE CLEARS THE SCREEN BY CALLING CLRSCR

SUBROUTINE, WHICH IS LOCATED IN ROM. THE NEXT LINE LOADS THE REGISTER PAIR BC WITH THE SCREEN LOCATION WERE WE WANT TO PRINT THE TEXT MESSAGE. REMEMBER THAT THIS IS A MEMORY LOCATION, & THAT THE SCREEN MEMORY BEGINS AT 3CØØH SO THE VALUE THAT NEEDS TO BE STORED IS 3CØØH PLUS 1DAH (474 BASE1Ø) WHICH GIVES 3DDAH, WHICH IS LOADED INTO REGISTER PAIR BC. THE NEXT LINE STORES THIS VALUE IN THE MEMORY ADDRESS POINTED TO BY CURLOC. (THAT'S WHAT THE BRACES INDICATE). THE NEXT LINE LOADS THE HL REGISTER PAIR WITH THE FIRST CHARACTER OF MESS1. THIS IS FOLLOWED BY A CALL TO THE MESSAGE DISPLAY SUBROUTINE, (DSPMSG). I WAS ALWAYS TAUGHT TO LIMIT LABEL NAMES TO A LENGTH OF 6 CHARACTERS, & WITHIN THIS LIMITATION, MEANINGFUL LABELS CAN BE CREATED. BELIEVE ME, BEFORE WRITING THE CODE FOR THIS ARTICLE I HADN'T TOUCHED ANY Z8Ø ASSEMBLER PROGRAMMING SINCE 1994. & I HADN'T DONE ANY TRS8Ø ASSEMBLER SINCE THE MID TO LATE 198Ø'S. SO WHEN I CAME TO REFRESH MY MEMORY BY REREADING THE PRINTOUTS THAT HAD SURVIVED VARIOUS PURGES, HAVING MEANINGFUL LABELS MADE THE JOB MUCH, MUCH EASIER. THE NEXT TWO LINES ARE SIMPLY NOP'S (NO OPERATION), THEY SIMPLY INCREMENT THE PROGRAM COUNTER, AFTER THE DISPLAY SUBROUTINE HAS BEEN EXECUTED. I FOUND I HAD TO ADD THEM TO GET THE PROGRAM TO WORK PROPERLY. THEY ALSO PROVIDE A WAY OF ADDING BREAKPOINTS WHEN DE-BUGGING A PROGRAM. THE LAST LINE CAUSES THE PROGRAM TO END BY CALLING THE DOS EXIT VALUE.

```
;
;*** DISPLAY MESSAGE SUBROUTINE       ***
;*** TERMINATES WITH A NULL CHARACTER ***
;*** ON ENTRY (BC) = CURSOR POSITION  ***
;***          (HL) = MESSAGE LOCATION ***
;
DSPMSG  PUSH    AF          ;SAVE REGISTERS
        PUSH    BC
        PUSH    HL
        LD  BC,(CURLOC)     ;GET 1ST PRINT LOCATION
 DSPØ1  LD  A,(HL)          ;GET MESS CHAR
        OR  A               ;TEST FOR ZERO
        JR  Z,DSPØ2         ;RETURN IF DONE
        LD  (BC),A          ;STORE CHARACTER
        INC BC              ;BUMP SCREEN POINTER
        INC HL              ;BUMP MESSAGE POINTER
        LD  (CURLOC),BC     ;SAVE POINTER
        CALL    DSPØ1       ;CONTINUE
 DSPØ2  POP HL              ;RESTORE REGISTERS
        POP BC
        POP AF
        RET
```

THE FIRST FEW LINES ARE SIMPLY COMMENTS, (THEY START WITH A ;) STATING THE PURPOSE OF THE SUBROUTINE, WHAT REGISTERS CONTAIN WHAT ON ENTRY, (& IF NEEDED, WHAT THEY CONTAIN ON EXIT, THOUGH IT WOULD BE BETTER TO SAVE THEM IN THE VARIABLES PART OF THE PROGRAM), & ANY OTHER INFORMATION THAT MIGHT BE USEFUL. AGAIN, THIS IS SIMPLY FOR EASE OF DEBUGGING, & TO HELP YOU REMEMBER THE DETAILS OF THE SUBROUTINE. DSPMSG IS THE LABEL THAT DEFINES THE FIRST LINE OF THE SUB-

ROUTINE, & THE FIRST THREE LINES DO NOTHING MORE THAN SAVE THE VALUES OF THE REGISTER PAIRS TO THE STACK. THE NEXT LINE LOADS THE BC REGISTER PAIR WITH THE VALUE STORED AT THE MEMORY LOCATION ATTACHED TO THE LABEL CURLOC. THE NEXT NEEDS EXPLAINING. THE LABEL DSPØ1 IS ONE THAT IS MEANT TO BE USED WITHIN THE SUBROUTINE ONLY, FOR LOOPING & JUMPING PURPOSES. THIS LINE LOADS THE A REGISTER WITH THE FIRST CHARACTER OF THE MESSAGE. IN THE NEXT, IT IS OR'D WITH ITSELF, & THEN IF THE RESULT IS ZERO, (AS IT WOULD BE IF THE CHARACTER IN THE A REGISTER WAS ØH), IT JUMPS TO THE LABEL DSPØ2 (ANOTHER INTERNAL LABEL). THE OR IS A LOGICAL ONE, & IS DONE ON A BIT BY BIT BASIS ON THE REGISTER.

A	B	OR	AND
Ø	Ø	Ø	Ø
Ø	1	1	Ø
1	Ø	1	Ø
1	1	1	1

IF THE CHARACTER IS NOT ZERO, THEN THE NEXT LINE LOADS THE VALUE IN THE A REGISTER INTO THE MEMORY LOCATION POINTED TO BY THE VALUE IN THE BC REGISTERS, (THE SCREEN MEMORY), SO IT IS DISPLAYED ON THE SCREEN WHEN THE SCREEN MEMORY IS SCANNED BY THE HARDWARE. THE NEXT TWO LINES INCREMENT THE VALUE STORED IN THE BC REGISTER, (WHICH POINTS TO THE NEXT MEMORY LOCATION TO BE PRINTED AT), & THE VALUE STORED IN THE HL REGISTER, (WHICH POINTS TO THE NET CHARACTER TO BE DISPLAYED. THE NEXT LINE SAVES THE CURRENT (INCREMENTED) VALUE OF SCREEN MEMORY, FOLLOWED BY A LOOP TO DSPØ1 IN THE NEXT, REPEATING THE WHOLE PROCESS AGAIN, UNTIL A NULL (ØH) CHARACTER IS ENCOUNTERED, AT WHICH POINT A JUMP TO DSPØ2 IS CARRIED OUT. AT THE LINE DSPØ2, THE ORIGINAL VALUES STORED IN THE REGISTER PAIRS AF, BC & HL ARE RESTORED (POP), REMEMBERING THAT THE STACK IS LIFO. (LAST IN, FIRST OUT), THEN THE PROGRAM RETURNS (RET) TO THE LINE AFTER THE CALL.

THE SCREEN DUMP BELOW SHOWS WHAT TO EXPECT TO BE DISPLAYED.



FINALLY, IF ANYONE HAS ANY TOPICS THEY WANT ME TO COVER, THEN PLEASE LET ME KNOW BY EMAILING ME AT PETER_STONE45@HOTMAIL.COM.

NEWDOS80 & EDTASM
QUICK REFERENCE SHEET

FUNCTION	FORMAT
ASSEMBLE	A(/SW(/SW...)) SWITCHES: LP -- LINE PRINTER NL -- NO LISTING NO -- NO OUTPUT NS -- NO SYMBOL TABLE WE -- WAIT ON ERRORS
BYE	B
DELETE	D(LINE REF OR RANGE)*
EDIT	E(LINE REF) SUBCOMMANDS: L -- LIST WORKING COPY OF LINE (N) <SPACE> -- ADVANCE (N) SPACE(S) BACKSPACE -- MOVE EDIT POINTER BACK ONE SPACE (N)SC -- SEARCH FOR THE (NTH) OF "C" (N)D -- DELETE NEXT (N) CHARACTER(S) (N)KC -- KILL UP TO (NTH) "C" (N)C -- CHANGE NEXT (N) CHR(S) A -- ABORT CHANGES & RESTART I -- INSERT CHRS: EXIT BY ENTER OR S-UP* X -- EXTEND LINE (ENTERS INSERT MODE) H -- HACK OF REMAINDER (ENTERS I MODE) E OR <ENTER> -- EXIT; SAVE CHANGES Q OR <BREAK) -- EXIT AND ABORT CHANGES S-UP* WILL ESCAPE FROM ANY EDIT MODE SUBCOMMAND
FIND	F(STRING)
HARD COPY	H(LINE REF OR RANGE)*
INSERT	I(LINE REF(,INCREMENT))
LOAD FILE	L D=FILESPEC
NUMBER	N(LINE REF(,INCREMENT))
PRINT	P(LINE REF OR RANGE)*
REPLACE	R(LINE REF)*
TEXT OUTPUT	T
WRITE FILE	W D=FILESPEC
* LINE REFERENCES:	NNNNN -- ACTUAL LINE NUMBER # -- FIRST LINE OF TEXT * -- LAST LINE OF TEXT . -- CURRENT LINE : -- FROM: TO SEPARATOR FOR RANGE REF S-UP -- SHIFT KEY + UP ARROW KEY

TRS8BIT



THE NIGHTS ARE DRAWING IN, IT'S ALREADY SEPTEMBER, BUT ONE OF THE GOOD THINGS ABOUT IT IS, IT'S TIME FOR YET ANOTHER EDITION OF TRS8BIT! I HOPE YOU AGREE WITH ME, THAT ONCE AGAIN, I THINK I'VE COME ACROSS A SUPER SELECTION OF ARTICLES FOR YOUR AMUSEMENT AND DELIGHT!

ONCE AGAIN, STAR BILLING GOES TO PETER STONE WITH PART 2 OF HIS Z80 ASSEMBLER PROGRAMMING REFRESHER COURSE.
PETER HAS RESPONDED TO VARIOUS QUESTIONS HE'S BEEN ASKED AND ONCE AGAIN, IT'S A VERY WELL PRODUCED ARTICLE FOR OUR ENJOYMENT. THANKS PETER.

THERE'S THE USUAL BITS IN 'AT THE READY PROMPT' WITH A MOST INTERESTING SMALL ARTICLE, WRITTEN BY DON BANNISTER EXPLAINING JUST HOW 'MEM' WORKS.

I ALSO CAME ACROSS (YET ANOTHER) OLD PHOTOCOPY OF AN ARTICLE HELPING TO CORRECT TAPE LOADING VOLUME PROBLEMS. SORRY, BUT I'VE NO ACCREDITATION FOR THIS

WHILST ON MY REGULAR TRAWLS OF EBAY, I FOUND SOMEONE OFFERING TO WRITE EPROMS FOR, WHAT I THOUGHT, WAS A VERY REASONABLE FEE. WELL WORTH A READ IF YOU'RE GOING DOWN THAT ROAD.
THERE HAS BEEN QUITE A BIT OF ACTIVITY ON THE UK'S EBAY ON THE TRS-80 SALES FRONT OVER THE LAST 3 MONTHS, WITH BOTH M1, M4 AND RELATED SOFTWARE SELLING FOR VERY MODEST AMOUNTS. I NOTICED A M4 COMPLETE WITH A TANDY HARD DRIVE AND A LOAD OF SOFTWARE WHICH DIDN'T GET A BIT. THE STARTING PRICE WAS ONLY £50

THERE HAS BEEN SOME MASSIVE UPGRADES ON THE EMULATOR FRONT OVER THE LAST FEW WEEKS. TIM MANN HAS UPGRADED HIS UNIX EMULATOR TO VERSION 4.9D.
MATTHEW REED HAS UPGRADED HIS TO RELEASE V1.21. THIS NEW RELEASE INCORPORATES AN EXATRON STRINGY FLOPPY (WHICH REQUIRES A COPY OF THE S/F ROM, SEE FURTHER DOWN). I WAS SO THRILLED WITH THIS IT GAVE ME NEW IMPETUS TO ATTACK MY ACULAB FLOPPY TAPE, WHICH I WON ON EBAY, AND TRY, AGAIN, TO GET IT TO WORK.
WELL, THERE'S BEEN GREAT EXCITEMENT AT MILLER TOWERS AS THE WONDERFUL BIT OF KIT NOW POWERS-UP. ALL I NEED TO DO NOW, IS GET A COPY OF THE ROM DOWN TO MY PC FOR THE EMULATOR TO USE!

MANY OF YOU HAVE BEEN KIND ENOUGH TO EMAIL ME FROM ALL OVER THE WORLD WITH TANDY RELATED ITEMS OF INTEREST. IT'S REALLY SUPER TO HEAR FROM YOU.
TERRY STEWART IN NEW ZEALAND RUNS A WEBSITE :- WWW.CLASSIC-COMPUTERS.ORG.NZ

I'VE PUT A LINK ON OUR SITE FOR YOUR CONVENIENCE, IT'S BRILLIANT AND WELL WORTH A VISIT. TERRY HAS BEEN WORKING TO RUN THE EXATRON STRINGY FLOPPY WITH MATTHEW REED'S EMULATOR. HE SENT ME A COPY OF THE ROM, REFERRED TO AS THE I/O PROGRAM. I'LL PUT THIS AS A DOWNLOAD ON THE WEBSITE ASAP. HE ALSO EMAILED ME THE METHOD HE USED TO COPY THE I/O PROGRAM TO A PC FOR USE

WITH THE EMULATOR. I'VE LISTED THE INSTRUCTIONS HE SENT ME IN THIS ISSUE.

ALSO; JOHN BENSON, IN AUSTRALIA IS LOOKING FOR A SET OF INSTRUCTION TO ASSEMBLE THE TC-8 CASSETTE SYSTEM FROM THE KIT-FORM THAT WAS AVAILABLE IN THE EARLY 1980'S. CAN YOU HELP? EMAIL ME WITH DETAILS PLEASE IF YOU HAVE A COPY OF THE INSTRUCTIONS OR KNOW WHERE THEY CAN BE OBTAINED AND I'LL PASS THEM ON.

MIGUEL DUTRA HAS RELEASED, THROUGH HIS WEB-SITE, A NEW PROGRAM FOR 'VIRTUAL DISK' READING, DIRECT FROM WINDOWS. IT'S A NEAT LITTLE PROGRAM AND WELL WORTH CHECKING OUT. I'VE PUT A LINK ON THE WEB-SITE FOR YOU ALL TO USE.

WE HAD A LITTLE BIT OF SAD NEW DURING JULY09. IT WAS REPORTED THAT THE MONTHLY MAGAZINE "PERSONAL COMPUTER WORLD" HAD CEASED PRODUCTION, THE LAST ISSUE BEING AUG09. I'VE BEEN A REGULAR SUBSCRIBER TO PCW SINCE ITS INCEPTION IN 1978 AND I SHALL MISS IT. IT WAS ONE OF THE BEST MAGAZINES AROUND ESPECIALLY THEIR 'HANDS-ON' REGULAR FEATURE. THEY HAD JUST STARTED A NEW, REGULAR FEATURE ON HISTORIC COMPUTERS! AH WELL..

WELL, I THINK THAT ABOUT WRAPS UP THIS ISSUE. DON'T FORGET US HERE AT TRS8BIT IF YOU'RE UP TO ANYTHING WONDERFUL WITH YOUR TANDY- KEEP IN TOUCH



BEST WISHES

DUSTY

AT THE
READY>
PROMPT



HERE'S A LITTLE PROGRAM IF YOU MANAGE TO FORGET THE NAME WRITTEN ON A SYSTEM TAPE.

```
10 INPUT #-1,A$:PRINT A$
```

YOU COULD, OF COURSE. VISIT KNUTH'S WEB SITE AND DOWNLOAD A COPY OF HIS PLAYCASS PROGRAM. THERE'S A LINK ON TRS-80.ORG.UK FOR YOUR CONVENIENCE!

I CAME ACROSS A SMALL ARTICLE, ORIGINALLY APPEARING IN THE EARLY 1980'S IN NATGUG NEWS, WRITTEN BY DON BANNISTER EXPLAINING HOW MEM WORKS ON THE MODEL1. THE COMPUTER PUTS SOMETHING IN EVERY LOCATION AND THEN TESTS TO SEE IF IT'S BEEN STORED. IN PRACTICE, IT ONLY DOES THIS ONCE, AS PART OF THE INITIALIZING, AND THEN PUTS THE ANSWER IN 40B1/2H. THE SLIGHT DELAY AFTER ENTERING A MEMORY SIZE, BEFORE 'R/S' ETC. APPEARS IS MOSTLY THE TIME TAKEN FOR THIS TEST. THE RELEVANT CODE STARTS IN ROM AT 00C7H. STRINGS ARE STORED BELOW THIS AREA (SIZE SET BY THE 'CLEAR' COMMAND), AND BELOW THIS GOES BASIC'S STACK. A BASIC PROGRAM STARTS FROM A POINT LOW IN MEMORY (JUST WHERE DEPENDS IF YOU HAVE DISKS) AND THE VARIABLES FOR THE PROGRAM GO ABOVE THIS. 'MEM' IS ACTUALLY THE AMOUNT OF ROOM LEFT BETWEEN THIS AND THE STACK.

I REMEMBER, DON ALSO FOUND HOW TO SOLVE THE CHR$(0) PROBLEM WITH EPSON PRINTERS. I'LL TRY AND LOOK IT OUT FOR THE NEXT EDITION.

CASSETTE LOADING PROBLEMS - A HARDWARE FIX

SORRY, BUT I'VE NO ACCREDITATION FOR THIS ARTICLE

If you have Level II, you are familiar with trying to load tapes that give little stars in the upper right of the screen - and the left one turns into a "C", or they both remain on. So you rewind your tape, change the volume setting slightly and try again, and again!

If you save one of your programs and the tape is good, you can usually reload it without problems. But when you try a tape made on another system, or you try to load system tapes (like the program conversion tape) you have trouble!

Being quite dissatisfied with this situation, I finally hooked up the oscilloscope and found large overshoots and huge amounts of ripple and hum. Disregarding what the manufacturer designed into the '80 to obtain data from pulses like these, I tried a small RC filter with a diode to clean up the waveform. It loaded the offending tape on the very first try! Fantastic, right? Well, no, not really...I found that tapes generated on my own system would not load now. So out came the oscilloscope again, and I found that the polarity needed to be reversed for my tapes. So I installed a reversing switch around the diodes and tried again. By paying attention to the polarity, all my "hard to load" tapes loaded fine.

Now hooking an oscilloscope to the filter just to load a tape just doesn't make a lot of sense. So I added a meter to give the necessary indications. Now I have a little box with the filter, the switch and a meter. Just plug the box into the earphone output of the recorder and the TRS-80 into the box. Turn the tape on manual, flip the polarity switch and leave it in the position which gives the highest reading. Adjust the volume level of the recorder for 15 microamps, rewind the tape and watch that hummer load!



I built up my box and went around to several other owners in the area and it worked great. One friend had the In-Memory programs which were always hard to load. Now they loaded on the first pass. It should be noted that this will do nothing for tapes with "drop outs".

Sweet success? For a while anyway. I finally ran into one that would not load. Putting the oscilloscope on it again indicated that it needed a much larger signal than was there. It was at this point that I changed the RC filter and eliminated the capacitor, and it appeared to clamp the overshoot better. Then I happened to put pressure on the right front corner of the cassette, and WOW!!, the amplitude of the signal jumped up where it should be! This seemed to be an indication of head alignment differences between the originating tape recorder and the one playing it back. Carefully looking at this new addition to tape loading procedures, I found that the tape cassette can actually twist a little from beginning to end due to differences in torque. This could explain why programs at the beginning or end might be easier to load. If the head is aligned with the tape, the amount of torque does not make any noticeable difference. When they are not together,

PARTS LIST

2 Germanium Signal Diodes
2 10K 5% Resistors
1 1K Resistor
1 82 Ohm Resistor
1 Double pole double throw Switch
1 0-50 Microamp Meter (RS Micronta Cat #22-051 or equiv
1 Miniature Jack
1 Miniature Plug
1 2" X 2" circuit board to mount parts
18" shielded wire
18" tie wire

Box for parts & meter - Archer Cat #270-233 or equiv

Note: Use of Silicon Diodes will work, but give signal saturation at 10 microamps instead of 15 microamps.

OPERATING INSTRUCTIONS

1 - Flip switch to determine highest reading.
2 - Leave switch at this setting adjust volume for 15 microamps.
3 - Rewind tape to beginning and Load it!

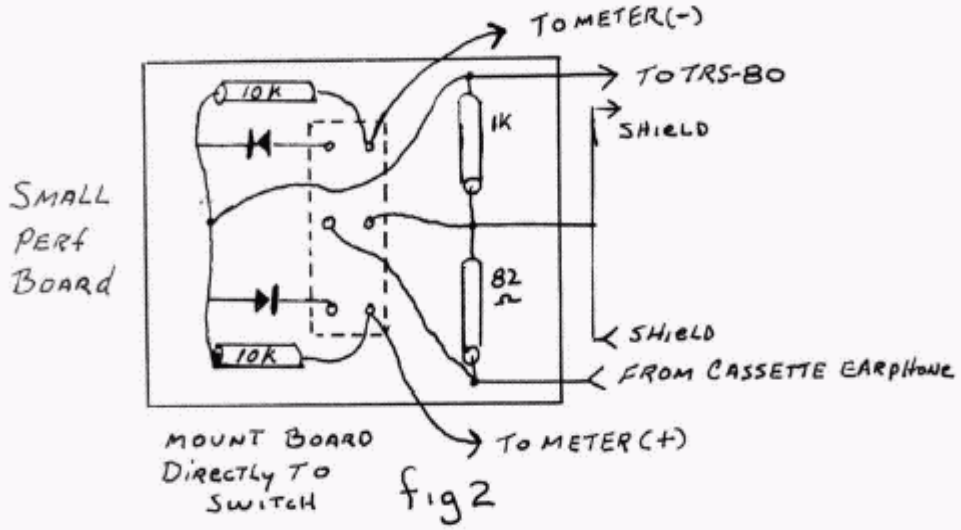


fig 2

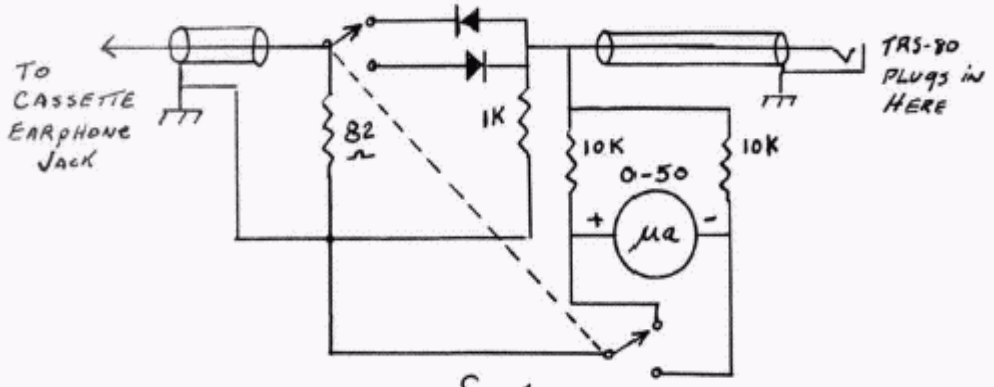
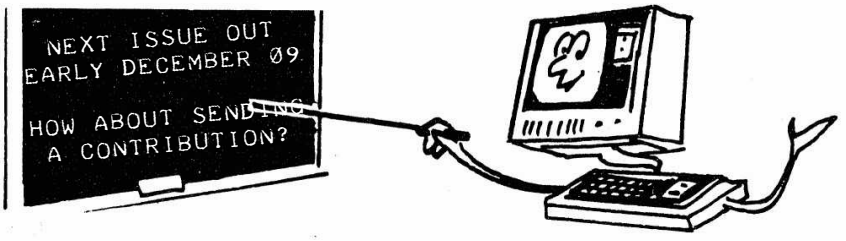


fig 1



there is a whole lot of difference.

Now the second modification: Take the bottom of the cassette recorder off first, then remove the top cover and drill a little hole in it so that the head alignment can be done thru the hole with a small screwdriver. Then decide which tape you will use as a standard. Most likely it will be one of those made on an expensive machine. I decided on the Program Conversion tape. Now if you get your old "C's" back, just slip your small screwdriver down the slot, watch the meter and peak the head into maximum reading. If the tape does not have actual "holes" in it, it will load for certain.

One other modification you may wish to do to remove ground loops, is to cut the circuit board connection between the AUX line return and the MIC return. Then add a jumper wire from the AUX line return to the EARphone return.

If you are going to put any kind of a "box" between your TRS-80 and the cassette recorder, put in one that will actually do something!

--

---== oØo ==---

HERE IS A COPY FROM AN EBAY LISTING OFFERING TO PROGRAM EPROMS, WHICH I THOUGHT WAS PERHAPS OF INTEREST.

THIS LISTING IS FOR A 2764 EPROM, PROGRAMMED WITH THE CODE OF YOUR CHOICE (YOU PROVIDE THE CODE [I.E. ROM IMAGE, ETC.)]. AS AN EXAMPLE, PERHAPS YOU HAVE A MODIFIED OR PATCHED DISK EXTENDED COLOUR BASIC FOR THE TANDY RADIO SHACK TRS-80 COLOUR COMPUTER THAT YOU WOULD LIKE TO HAVE IN AN EPROM FOR PLUGGING INTO YOUR COCO DISK CONTROLLER. ALL YOU WOULD NEED TO DO AFTER PURCHASING THE ITEM FROM THIS LISTING WOULD BE TO EMAIL ME THE ROM IMAGE (AS AN ATTACHMENT) THAT YOU NEED BURNED INTO THE EPROM (OR POINT ME TO AN URL FOR DOWNLOADING SAID EPROM IMAGE) AND I WILL PROCEED TO PROGRAM AND SHIP YOUR NEWLY PROGRAMMED 2764 EPROM CHIP.

IMPORTANT NOTES:

1 - I MENTION THE COCO ABOVE AS AN EXAMPLE, MAINLY BECAUSE THE PRIMARY FOCUS OF MY BUSINESS IS TO SUPPORT THE TRS-80 COLOUR COMPUTER, AS THIS RECENTLY BECAME A HOBBY OF MINE. THIS DOES NOT MEAN THAT I WILL ONLY PROGRAM COCO RELATED EPROM'S. I WILL, IN FACT, PROGRAM THE EPROM WITH ANY BINARY FILE YOU PROVIDE.

2 - IF YOU INTEND TO USE THIS EPROM IN A TRS-80 COLOUR COMPUTER FLOPPY DISKETTE CONTROLLER, YOU SHOULD BE AWARE THAT THIS EPROM IS A 28-PIN, 8K X 8BIT DEVICE. MOST COCO FDC'S MANUFACTURED BY RADIO

SHACK (TANDY) WERE DESIGNED FOR THE MOTOROLA MCM68AXX SERIES OF MASK ROM'S WHICH ARE 24-PIN DEVICES. THIS IS TRUE OF ALL THE TANDY DISK CONTROLLERS UP TO, AND INCLUDING THE FD-501. IT IS THEREFORE NOTED THAT A 2764 EPROM WILL NOT WORK IN SUCH A CONTROLLER, WITHOUT THE USE OF A 24 TO 28 PIN SOCKET ADAPTER. THE FD-502 (THE LAST COCO FDC MANUFACTURED BY TANDY BEFORE THEY DECIDED TO DISCONTINUE THE COCO) AND MOST THIRD PARTY COCO FDC'S CONTAIN ONE OR MORE 28-PIN EPROM SOCKETS, AND SO THE 2764 EPROM MAY BE USED IN MANY OF THESE MODELS WITHOUT ISSUE. I BELIEVE THAT ALL OF THE FDC'S MANUFACTURED FOR AND BY DRAGON DATA FOR THEIR TANO DRAGON (COCO CLONE) COMPUTER USED 28-PIN ROM'S, HOWEVER, I AM NOT 100% CERTAIN. YOU SHOULD, THEREFORE, VERIFY THAT YOUR FDC CAN TAKE A 28-PIN EPROM PRIOR TO ORDERING.

3 - THESE EPROMS ARE SALVAGE - THAT MEANS THEY WERE REMOVED FROM WORKING EQUIPMENT AND BULK ERASED, MAKING THEM READY TO BE PROGRAMMED BY ME FOR YOU. AS SUCH, IT IS LIKELY THAT SOME OF THESE WILL HAVE SOME BENT PINS, BUT NOTHING THAT WOULD KEEP IT FROM PROPERLY PLUGGING INTO A SOCKET. I WILL DO MY BEST TO MAKE SURE THAT ALL OF THE PINS ARE AS STRAIGHT AS POSSIBLE, BUT YOU SHOULD BE AWARE OF THIS. I STRIVE FOR AN HONEST, ACCURATE DESCRIPTION.

4 - THE SPEED GRADE OF THESE DEVICES ARE 200NS OR BETTER. MANUFACTURERS VARY, AND MAY BE ANY OF THE FOLLOWING: ST M2764A, INTEL D2764A, INTEL D2764 (OR OTHER MANUFACTURERS.)

5 - SHIPPING IS ONLY $1 TO ANYWHERE IN THE UNITED STATES. EXACT COST ELSEWHERE. I WILL COMBINE SHIPPING, SO WAIT FOR AN INVOICE ON MULTIPLE ITEM PURCHASES. DELIVERY CONFIRMATION IS REQUIRED, THEREFORE, THERE IS AN ADDITIONAL CHARGE OF $1.00 PER SHIPPED PACKAGE. DELIVERY CONFIRMATION IS NOW FREE (THE ADDITIONAL DOLLAR IS NO LONGER ADDED.

6 - IF YOU ARE UNCERTAIN AS TO HOW TO GET THE ROM IMAGE TO ME, CONTACT ME AND I'LL HELP YOU WORK IT OUT. IN ADDITION TO STANDARD FORMATS, I CAN ALSO ACCEPT THE ROM IMAGE ON A COCO COMPATIBLE FLOPPY DISKETTE (3.5" OR 5.25").

2764 8K X 8 EPROM, 200NS OR BETTER, 28-PIN DIP PACKAGE. PROGRAMMED TO YOUR LIKING, WITH THE CODE OF YOUR CHOICE.

PLEASE BE CERTAIN THAT YOUR CODE WILL FIT INTO AN 8KX8 EPROM.



--== o0o ==--

COPYING THE EXATRON I/O PROGRAM

TERRY STEWART

0. FIRST CHECK THAT YOU CAN LOAD THE DATA I/O PROGRAM SUCCESSFULLY AND THE WAFER IS OK

1. HOOK UP THE TRS80 MODEL 1 VIA IT'S CASSETTE INTERFACE TO THE PC'S SOUNDCARD AND MAKE SURE YOU CAN BOTH RECORD AND LOAD PROGRAMS AS WAV FILES. I SUGGEST USING THE FREE PROGRAM AUDACITY AUDACITY.SOURCEFORGE.NET AS YOUR SOUND RECORDER/PLAYER. YOU WILL PROBABLY NEED TO PLAY AROUND WITH SOUND LEVELS A BIT UNTIL YOU CAN RELIABLY SAVE AND LOAD A BASIC PROGRAM. MAKE SURE YOU CAN DO THIS BEFORE GOING ANY FURTHER.

2. FROM WWW.CLASSIC-COMPUTERS.ORG.NZ/TEMP/TASMON.WAV YOU CAN DOWNLOAD A HIGH MEMORY VERSION OF THE MACHINE LANGUAGE MONITOR PROGRAM TASMON AS A WAV FILE. THE WAV CONTAINS A RELOCATED VERSION OF TASMON WHICH LOADS INTO HIGH MEMORY WELL CLEAR OF THE BOTTOM 16K OF RAM. YOU CAN PICK UP THE MANUAL HERE WWW.CLASSIC-COMPUTERS.ORG.NZ/TEMP/MANUALS-TASMON.PDF.

3. ENSURE THE TRS-80 AND PC ARE SET UP SO THAT THE TRS-80 IS READY TO LOAD TASMON FROM THE PC VIA AUDACITY WHEN REQUIRED (A BIT LIKE HAVING A CASSETTE LOADED AND READY). MAKE SURE THE ESF IS ATTACHED TO THE TRS-80 ALSO.

4. BOOT THE TRS-80. IF YOU USING THE EXPANSION INTERFACE TO GET THE 48K NEEDED, YOU'LL NEED TO HOLD BREAK DOWN SO IT BOOTS TO LEVEL 2 BASIC INSTEAD OF HANGING OF COURSE.

5. INITIALISE THE STRINGY FLOPPY IN THE NORMAL WAY (>SYSTEM THEN >/12345)

6. LOAD IN THE DATA I/O TAPE HOLDING THE SHIFT KEY DOWN. THIS WILL STOP IT INITIALISING AND RELOCATING. ONCE THE TAPE STOPS (ASSUMING NO ERRORS) THE PROGRAM WILL BE NOW SITTING IN AN AREA AROUND THE MIDDLE OF THE BOTTOM 16K OF RAM (I CAN'T REMEMBER THE EXACT LOCATION). THREE NUMBERS WILL APPEAR ON THE SCREEN. THESE WILL BE THE MEMORY LOCATION START ADDRESS, THE LENGTH OF THE FILE, AND THE ENTRY ADDRESS, ALL IN DECIMAL. WRITE THOSE NUMBERS DOWN SOMEWHERE. YOU WILL NEED THEM LATER.

7. WHATEVER YOU DO, DON 'T REBOOT THE MACHINE. NOW LOAD IN TASMON FROM THE PC USING THE SAME COMMANDS ON THE TRS80 YOU WOULD USE AS IF IT'S A MACHINE LANGUAGE CASSETTE FILE (>SYSTEM <RETURN> THEN T, THEN <RETURN>). TASMON SHOULD LOAD INTO HIGH MEMORY RIGHT AT THE TOP OF RAM (E000 TO FFFF) AND WELL CLEAR OF THE ESF DATA I/O PROGRAM.

8. ONCE TASMON HAS LOADED, START THE PROGRAM (WITH /). YOU'LL NOW NEED TO WRITE OUT THE DATA IN THE AREA OF RAM CONTAINING THE DATA I/O PROGRAM.

9. RECONFIGURE THE PC AND AUDACITY FROM PLAYBACK MODE TO RECORD MODE. HAVING LOADED A FILE IN (TASMON) YOU ARE NOW ABOUT TO WRITE ONE OUT. (AS IF YOU ARE PUTTING A BLANK CASSETTE IN THE TAPE RECORDER).

10. TAKING THE NUMBERS SHOWN IN STEP 6, WORK OUT THE STARTING ADDRESS, ENDING ADDRESS (NOT LENGTH!) AND

ENTRY ADDRESS OF THE DATA I/O PROGRAM ALL IN HEX (NOT DECIMAL). TASMON WILL NEED THESE FIGURES. THE ENDING ADDRESS CAN BE WORKED OUT BY ADDING THE LENGTH TO THE STARTING ADDRESS OF COURSE.

11. SET UP TASMON TO WRITE A WAV OF THE MEMORY SEGMENT WHERE THE DATA I/O PROGRAM IS NOW RESIDING. DO THIS BY TYPING "W <ENTRY ADDY IN HEX> <END ADDRESS IN HEX> <ENTRY ADDRESS IN HEX>" THEN PRESS <RETURN> THEN TYPE DATAIO BUT DONT PRESS RETURN AGAIN JUST YET. (YOU MIGHT WANT TO CHECK THIS WRITE TO CASSETTE COMMAND IN THE TASMON MANUAL.)

12. CLICK THE RECORD BUTTON IN AUDACITY ON THE PC

13. PRESS RETURN ON THE TRS80. TASMON WILL START WRITING DATA OUT THE CASSETTE PORT TO THE PC WHERE AUDACITY WILL BE CAPTURING IT.

14. WHEN FINISHED, STOP AUDACITY FROM RECORDING AND WRITE OUT THE FILE ON THE PC AS A WAV.

15. BOOT UP MATTHEW REED'S EMULATOR WITH STRINGY FLOPPY ENABLE AND DISK DRIVES DISABLED

16. INITIALISE THE EMULATED STRINGY FLOPPY IN LEVEL 2 BASIC JUST AS YOU DID ON THE REAL MACHINE

17. CREATE A NEW STRINGY FLOPPY WAFER IN THE EMULATOR. MAKE THIS 5 INCHES AND INITIALISE IT BY TYPING "@NEW" IN LEVEL 2 BASIC

18. USING THE SYSTEM COMMAND (NAME DATAIO) LOAD IN THE WAV FILE YOU JUST MADE FROM THE REAL TRS-80. WHEN FINISHED LOADING DON'T START IT, BUT INSTEAD SELECT WARM BOOT FROM THE FILE MENU, WHICH WILL RETURN TO BASIC PROMPT BUT LEAVE THE NEWLY LOADED MEMORY CONTENTS INTACT.

19. USING THE SAME FIGURES YOU WROTE DOWN IN STEP 6, SAVE THE SECTION OF MEMORY ON THE NEW WAFER BY TYPING @SAVE1,<START ADDRESS>,<LENGTH>,<ENTRY ADDRESS> IN DECIMAL.

20 TYPE @LOAD TO CHECK IF THE PROGRAM WORKS. IF IT WORKS, YOU'LL KNOW IT.

WELL, THAT'S SIMILAR TO THE METHOD I USED TO GET THE ESF MONITOR PROGRAM ONTO VIRTUAL WAFER (-: IT'S A BIT OF A PROCEDURE BUT IT SEEMS TO WORK. IT MIGHT BE HELPFUL TO A FEW OTHERS WANTING TO GET M/L PROGRAMS OFF WAFER SO I'LL PROBABLY PUT THESE TIPS ON MY BLOG IN THE NEAR FUTURE.

IF YOU ONLY HAVE A 16K TRS-80 YOU COULD ALSO USE THE SAME METHOD EXCEPT USING A LOW MEMORY VERSION OF TASMON THAT LOADS IN AT 4300H. I'M SURE IT DOESN'T GO HIGH ENOUGH TO CONFLICT WITH THE DATA I/O PROGRAM AS IT LOADS IN. I DON'T HAVE A LOW MEM VERSION OF TASMON BUT I COULD EASILY MAKE ONE. LET ME KNOW IF YOU NEED IT.

GOOD LUCK! I ENCOURAGE YOU TO HAVE A GO. NO ONE HAS COME OUT OF THE WOODS WITH AN IMAGE OF THE DATA I/O PROGRAM YET! IT WILL BE A GREAT THING FOR THE TRS-80 COMMUNITY IF WE CAN RECOVER THIS PROGRAM.

--== o0o ==--

ASSEMBLY LANGUAGE MADE EASY? PART 2

by Peter Stone

Welcome to the second instalment in a series aiming to refresh your assembly programming skills.

In the last article I published my email address with a request for topics that you wanted me to cover. I received an email from John Peet, asking about an assembler equivalent to the Basic IF THEN ELSE command. This article hopefully will provide a solution.

The way I've done this in the past, usually to make choices in menus is as follows.

```
GETINF  CALL INPUT          ;GET A SINGLE CHAR
        RES  5,A            ;FORCE UPPERCASE
        CP   'R'            ;IS IT RIGHT?
        JP   Z,RIGHT
        CP   'L'
        JP   Z,LEFT
        CP    'X'
        JP   Z,QUIT
        JP    ERRMSG
        JR   GETINF
```

Going through the program, the first line calls the input subroutine (in ROM). This waits for a single character to be entered. The RES instruction is used to reset (force to zero,) a single bit in the named register. This is the complimentary instruction to SET which is used to set a bit (force to 1,) a single bit in the named register. In this case we are resetting bit 5 of the A register, to force the character entered into uppercase. Otherwise, we would have to test for both R & r which would increase the size & complexity of the program. (& the risk of forgetting to actually include all the options.)

A brief look at the RES instruction would be useful at this point. The basic format of instruction is RES b,r where b it the bit required & r is one of the A,B,C,D,E,H,L registers. Just to add to the confusion, as far as this instruction is concerned, the least significant bit (lsb) of the register is called 0, & the most significant bit is 7. So as far as our example is concerned, the bit pattern for r is 01110010 & the bit pattern for R is 01010010. So the instruction RES 5,A actually resets bit 6 of the byte. For completion I should point out that there are variations on this instruction, RES b,(HL) resets the b bit of the memory address pointed to by the HL register pair. There is also RES b,(IX+disp) & RES b,(IY+disp).

The next line in the program compares the value in the A register with the character R. The next line jumps to the subroutine RIGHT if there is a match. if there isn't a match, then the program goes on to compare the value in the A register against the value L & jumps to the subroutine LEFT if there isn't a match. The final comparison in the short example above does a compare against the character X, & exits the program via the QUIT subroutine. (The reason for exiting in this way, is that you may wish to save variables or text before actually exiting the program.) The last two lines simply display an error message if no match is found, & then jumps back to the start to get another value. To actually make this into a "proper" IF THEN ELSE subroutine, simply make the following changes.

```
SUB1    CALL INPUT
        CP   'R'            ;Replace R with the value to compare
        JP   Z,SUB2
        JP   SUB3
        RET
```

A complete program to show how this all works follows. This is actually based on my hello program from the last issue of TRS8BIT.

```
        ORG  7000H          ;SEEMS LIKE A GOOD PLACE TO START
INKEY   EQU  002BH          ;GET SINGLE CHAR, NO WAIT
INPUT   EQU  0049H          ;GET SINGLE CHAR, WAIT FOR IT
LININP  EQU  0040H          ;GET LINE INPUT, CR TERMINATES
DISP    EQU  0033H          ;DISPLAY SUBROUTINE
CLRSCR  EQU  01C9H          ;CLS
TXTPRT  EQU  28A7H          ;PRINT TEXT TO SCREEN
DOSEXT  EQU  402DH          ;EXIT TO DOS
LINE1   EQU  3C00H          ;START OF LINE 1 OF SCREEN MEM
LINE2   EQU  3C40H          ;START OF LINE 2 OF SCREEN MEM
LINE3   EQU  3C80H          ;START OF LINE 3 OF SCREEN MEM
LINE4   EQU  3CC0H          ;START OF LINE 4 OF SCREEN MEM
LINE5   EQU  3D00H          ;START OF LINE 5 OF SCREEN MEM
LINE6   EQU  3D40H          ;START OF LINE 6 OF SCREEN MEM
LINE7   EQU  3D80H          ;START OF LINE 7 OF SCREEN MEM
LINE8   EQU  3DC0H          ;START OF LINE 8 OF SCREEN MEM
LINE9   EQU  3E00H          ;START OF LINE 9 OF SCREEN MEM
LINE10  EQU  3E40H          ;START OF LINE 10 OF SCREEN MEM
LINE11  EQU  3E80H          ;START OF LINE 11 OF SCREEN MEM
LINE12  EQU  3EC0H          ;START OF LINE 12 OF SCREEN MEM
LINE13  EQU  3F00H          ;START OF LINE 13 OF SCREEN MEM
LINE14  EQU  3F40H          ;START OF LINE 14 OF SCREEN MEM
LINE15  EQU  3F80H          ;START OF LINE 15 OF SCREEN MEM
LINE16  EQU  3FC0H          ;START OF LINE 16 OF SCREEN MEM
START   CALL CLRSCR         ;CLEAR THE SCREEN
        LD   A,08CH         ;GRAPHICS CHAR
        LD   DE,LINE13      ;START OF LINE13 (OBVIOUSLY)
        LD   BC,64          ;NUM OF CHARS TO PRINT
        CALL PRNTCH         ;LETS DO IT
        LD   BC,LINE1       ; GOTO START OF DISPLAY
        LD   (CURLOC),BC    ; STORE IT
        LD   HL,MLINE1      ;GOTO START OF MENU
        CALL DSPMSG         ;PRINT IT
        LD   BC,LINE3       ;ETC.
        LD   (CURLOC),BC
        LD   HL,MLINE2
        CALL DSPMSG
        LD   BC,LINE4
        LD   (CURLOC),BC
        LD   HL,MLINE3
        CALL DSPMSG
        LD   BC,LINE5
        LD   (CURLOC),BC
        LD   HL,MLINE4
        CALL DSPMSG
        LD   BC,LINE6
        LD   (CURLOC),BC
        LD   HL,MLINE5
        CALL DSPMSG
        LD   BC,LINE7
        LD   (CURLOC),BC
        LD   HL,MLINE6
        CALL DSPMSG
        LD   BC,LINE9
        LD   (CURLOC),BC
        LD   HL,MLINE7
        CALL DSPMSG
GETINF  CALL INPUT          ;GET A SINGLE CHAR
```

```
        RES  5,A            ;FORCE UPPERCASE
        CP   'R'            ;IS IT RIGHT?
        JP   Z,RIGHT
        CP   'L'
        JP   Z,LEFT
        CP   'F'
        JP   Z,FWARD
        CP   'B'
        JP   Z,BACK
        CP   'X'
        JP   Z,QUIT
        JP   ERRMSG
        JR   GETINF
DSPMSG  PUSHAF              ;SAVE REGISTERS
        PUSHBC
        PUSHHL
        LD   BC,(CURLOC)    ;GET 1ST PRINT LOCATION
DSP01   LD   A,(HL)         ;GET MESS CHAR
        OR   A              ;TEST FOR ZERO
        JR   Z,DSP02        ;RETURN IF DONE
        LD   (BC),A         ;PRINT CHARACTER
        INC  BC             ;BUMP SCREEN POINTER
        INC  HL             ;BUMP MESSAGE POINTER
        LD   (CURLOC),BC    ;SAVE POINTER
        JR   DSP01          ;CONTINUE
DSP02   POP  HL             ;RESTORE REGISTERS
        POP  BC
        POP  AF
        RET
PRNTCH  LD   (DE),A         ;PRINT CHAR
        INC  DE             ;BUMP POINTER
        DEC  BC             ;DECREMENT COUNT
        PUSHAF              ;SAVE CHAR WE'RE PRINTING
        LD   A,B            ;TEST FOR ZERO
        OR   C
        JR   Z,PR01         ;END IF DONE
        POP  AF             ;RESTORE CHAR WE'RE PRINTING
        JR   PRNTCH         ;LOOP
PR01    POP  AF             ;RESTORE A REG
        RET
RIGHT   LD   BC,LINE15      ;LOAD BC WITH POSITION OF LINE 15
        LD   (CURLOC),BC    ;STORE VALUE IN BC IN CURLOC
        LD   HL,MRHT        ;LOAD HL WITH LOCATION OF MESSAGE
        CALL DSPMSG         ;DISPLAY MESSAGE
        JP   GETINF         ;GET NEXT CHARACTER
LEFT    LD   BC,LINE15
        LD   (CURLOC),BC
        LD   HL,MLFT
        CALL DSPMSG
        JP   GETINF
FWARD   LD   BC,LINE15
        LD   (CURLOC),BC
        LD   HL,MFWD
        CALL DSPMSG
        JP   GETINF
BACK    LD   BC,LINE15
        LD   (CURLOC),BC
        LD   HL,MBCK
        CALL DSPMSG
```

```
            JP    GETINF
QUIT        CALL  DOSEXT                :FINISHED WITH PROGRAM
ERRMSG      LD    BC,LINE15             ;OOPS, INCORRECT VALUE ENTERED
            LD    (CURLOC),BC
            LD    HL,MEMSG
            CALL  DSPMSG
            JP    GETINF
;*** MESSAGES STORED HERE ***
MLINE1      DEFM        'Do you want to go'
            DEFB 0
MLINE2      DEFM        '<R>ight'
            DEFB 0
MLINE3      DEFM        '<L>eft'
            DEFB 0
MLINE4      DEFM        '<F>orward'
            DEFB 0
MLINE5      DEFM        '<B>ack'
            DEFB 0
MLINE6      DEFM        'e<X>it to dos'
            DEFB 0
MLINE7      DEFM        'Enter the letter of your choice '
            DEFB 0
MRHT        DEFM        'You want to go right      '
            DEFB 0
MLFT        DEFM        'You want to go left        '
            DEFB        0
MFWD        DEFM        'You want to go forward  '
            DEFB        0
MBCK        DEFM        'You want to go back     '
            DEFB 0
MEMSG       DEFM        'Your choice is not valid '
            DEFB 0
;*** VARIABLES STORED HERE ***
CURLOC      DEFW        3C00H           ;STORE CURSOR LOCATION
            END   START                 ;DONE
```

The first change to the program, occurs in the equates part of the program. I've added a series of lines that define the starting locations of the 16 lines of screen memory. This was done to help with another program I've been working on. In the hello world program, you will recall that I used the screen position of 474. As you can imagine, this took an amount of trial & error to get right. An easier way of doing this, which I've used programs in the past, is instead of using LD BC,474, use LD BC,LINE8+26. Which can be calculated from simply looking at a video worksheet & remembering there are 64 characters to the line.

If we look at the first part of the actual program, (shown below,) The first line uses the ROM subroutine to clear the screen & position the cursor at the home location.

```
START       CALL  CLRSCR            ;CLEAR THE SCREEN
            LD    A,08CH            ;GRAPHICS CHAR
            LD    DE,LINE13         ;START OF LINE13 (OBVIOUSLY)
            LD    BC,64             ;NUM OF CHARS TO PRINT
            CALL  PRNTCH            ;LETS DO IT
```

The next line loads the A register with the graphic character 140. I'm mindful of the size of this article & Dusty's download limits, so I'll deal with graphic characters in my next article. The next line loads the DE register with the start of line 13, & the following line loads the BC register with the number of characters to print, (remember there are 64 characters to the line.) Then there is a call to a

new subroutine PRINTCH.

```
PRNTCH  LD    (DE),A          ;PRINT CHAR
        INC   DE              ;BUMP POINTER
        DEC   BC              ;DECREMENT COUNT
        PUSH  AF              ;SAVE CHAR WE'RE PRINTING
        LD    A,B             ;TEST FOR ZERO
        OR    C
        JR    Z,PR01          ;END IF DONE
        POP   AF              ;RESTORE CHAR WE'RE PRINTING
        JR    PRNTCH          ;LOOP
PR01    POP   AF              ;RESTORE A REG
        RET
```

The first line of PRINTCH takes the value in the A register & prints it at the memory location pointed to by the DE register pair. The next line increments the DE register pair to point at the next screen memory location, then the next decrements the BC register pair, (which, we said, contain the number of characters to print.) The next line saves the AF register pair on the stack, so we can save the character stored in the A register. Next, the value in the B register is loaded into the A register, & then or'd with the C register. From last issues discussion on how the or instruction works, we know that if either the A register or the C register contain a 1, then the Z (zero) flag is not set, & the program continues. If both registers contain 0, then the Z flag is set, & the JR Z,PR01instruction terminates the program, by restoring the AF register pair & then doing a return. If the Z flag is not set, then the AF register pair is restored, & the program loops round again.

If we apply a bit of thought, we can see that this is a versatile routine. Load the A register with the value 0BFH, (191) & the DE register with the value of line 1, (LD DE,LINE1) then load BC with the value 0400H, (1024), & then call the PRINTCH subroutine, to white out the screen. If we replace the value in the A register with 080H, leaving the other values the same, & call the subroutine, we can clear the screen. Another way of using this, would be to load the A register with 080H, load the DE register pair with the value of line 5, then load BC with the value of 0140H, (320) to blank out the middle of the screen.

Going back to the main program, after we've printed a separator line across the screen at line13, we load the BC register pair with the value of Line1, & as we did in the last article, we save the value in CURLOC, point the HL register pair to the location of message line 1 (MLINE1), then call DSPMSG. After doing this, we load BC with the value of line3, (to leave a blank line) & do the whole thing again.

```
        LD    BC,LINE1        ; GOTO START OF DISPLAY
        LD    (CURLOC),BC     ; STORE IT
        LD    HL,MLINE1       ;GOTO START OF MENU
        CALL  DSPMSG          ;PRINT IT
        LD    BC,LINE3        ;ETC.
        LD    (CURLOC),BC
        LD    HL,MLINE2
        CALL  DSPMSG
```

This is repeated for all the messages we want to display. (see the screen printout at the bottom of the article.) The messages are stored in the same way as in my previous article.

The next part of the program is the GETINF subroutine.

```
GETINF  CALL  INPUT           ;GET A SINGLE CHAR
        RES   5,A             ;FORCE UPPERCASE
        CP    'R'             ;IS IT RIGHT?
        JP    Z,RIGHT
        CP    'L'
        JP    Z,LEFT
```

```
        CP    'F'
        JP    Z,FWARD
        CP    'B'
        JP    Z,BACK
        CP    'X'
        JP    Z,QUIT
        JP    ERRMSG
        JR    GETINF
```

I've explained the main principles of how GETINF works, at the start of the article, so there is no need to repeat it here.

The next part of the program to be considered, is the set of instructions to be executed when a valid menu item is entered. In our example, we are simply printing a reply to a correctly inputted character, & an error message if an incorrect one is printed. It should be noted, that these responses could be further sub-menus, with commands to store values. An example might be to set the various parameters in a terminal program.

```
RIGHT   LD    BC,LINE15          ;LOAD BC WITH POSITION OF LINE 15
        LD    (CURLOC),BC        ;STORE VALUE IN BC IN CURLOC
        LD    HL,MRHT            ;LOAD HL WITH LOCATION OF MESSAGE
        CALL  DSPMSG             ;DISPLAY MESSAGE
        JP    GETINF             ;GET NEXT CHARACTER
```

This routine works in a similar manner to the part that printed our choices to the screen. Except that we have pointed DSPMSG to line 15 to print the response. The only line that is different is the last one, where we go back to get another character by using a JP to the GETINF part of the program. The MRHT message is in our standard format, & needs no further explanation.

CORRECTIONS.

Unfortunately, in my last article, an error crept into the program listing of the DSPMSG subroutine.

The printed code is shown below;

```
        LD    (CURLOC),BC    ;SAVE POINTER
        CALL  DSP01          ;CONTINUE
DSP02   POP   HL             ;RESTORE REGISTERS
```

The correct version is;

```
        LD    (CURLOC),BC    ;SAVE POINTER
Modify ----> JR   DSP01      ;CONTINUE
DSP02   POP   HL             ;RESTORE REGISTERS
```

Apologies for this. Blame it on a 26 year old telex printout. (These where not known for their storage properties!)

IF ANYONE HAS ANY TOPICS THEY WANT ME TO COVER,
THEN PLEASE LET ME KNOW BY EMAILING ME AT
PETER_STONE45@HOTMAIL.COM.

TRS8BIT



IT'S CHRISTMAS AGAIN (HOW THE YEAR HAS FLOWN!) SO BEFORE I GO ANY FURTHER, LET DEE AND MYSELF WISH YOU ALL A

VERY MERRY CHRISTMAS AND A HAPPY AND HEALTHY 2010

I'M PLEASED TO REPORT THAT THE WEB SITE HAS JUST CLOCKED-UP 4000 HITS. QUITE A PLEASING MILE-STONE FOR ME, SO THANKS TO ALL YOU VISITORS FOR YOUR TIME, INTEREST AND ENTHUSIASM YOU'VE SHOWN FOR MY MEAGRE EFFORTS.

JUST IN CASE YOU HAVEN'T NOTICED, AT LONG LAST, I'VE PRODUCED AN INDEX, FOR THE FIRST SIX VOLUMES, OF 80-US. AS PER USUAL, IT'S AVAILABLE, AS A 'ZIP' FILE, FOR DOWNLOADING AS EITHER A M/S WORKS DATABASE FORMAT (WDB) OR A COMMA SEPARATED VALUES FILE (CSV).
PLEASE LET ME KNOW OF ANY ERRORS, MISSING ITEMS OR TYPO'S YOU FIND AND I'LL GET THEM CORRECTED ASAP. WHILE I'M ON THE SUBJECT OF INDEXES, I'VE UPGRADED THE NATGUG NEWS INDEX WITH A NUMBER OF MINOR CORRECTIONS, AND TYPO'S.

I'VE JUST HAD A STRANGE BIT OF LUCK. I WAS WATCHING AN EBAY SALE FOR A BOOK ENTITLED "THE TELEPRINTER HANDBOOK" AND HAD LEFT A £14 BID. IT WAS STUCK AT 99P FOR THE FIRST 9 DAYS, 23 HRS ETC ETC AND, TRUE TO FORM, I WAS DULY SNIPED AT £14.50 WITH 5 SECONDS TO GO. I REGRETTED NOT LEAVING A LARGER AMOUNT, AS YOU DO, BUT WELL, AS DEE SAYS, 'YOU CAN'T HAVE IT ALL'. ANYWAY, A COUPLE OF SUNDAYS AGO, I WAS AT A LOCAL CAR-BOOT SALE, AND GUESS WHAT! THERE, ON THE GRASS WITH A LOAD OF OTHER BITS AND BOBS WAS THE TELEPRINTER HANDBOOK. "HOW MUCH ?" I ENQUIRED. THE CHAP SHRUGGED HIS SHOULDERS; "ANY GOOD FOR YOU AT 50P?" "THAT'LL DO FINE" I SAID. I'LL HAVE A QUICK 'SKIM' THROUGH IT AND PLACE IT ON THE LENDING LIBRARY SHELF. IF ANYONE NEEDS TO BORROW IT, PLEASE LET ME KNOW.

TRS-80'S ARE STILL IN ABUNDANCE ON EBAY (UK). ALL MODELS ARE MAKING REGULAR APPEARANCES WITH EVEN SOME SOFTWARE FOR THE MODEL 2! THERE'S QUITE A LOT OF 'DEALER' ACTIVITY WITH CO-CO'S (AT NO-HOPER PRICES! I.E. £90+). OTHER PRICES DO FLUCTUATE SOMEWHAT, WITH MODEL 1'S WITH E/I'S FETCHING BETWEEN £12 AND £58! THANKS TO THE GENEROSITY OF VARIOUS CONTACTS, I'VE QUITE A SUPPLY OF MODEL 1 BITS, SO IF YOU'RE IN DESPERATE NEED OF SOMETHING, TRY AN EMAIL AND I MIGHT BE ABLE TO HELP.

I ALSO TRIED AN EXPERIMENT BY ADVERTISING A COPY OF TRS8BIT FOR SALE ON EBAY. (AT 99P INCLUSIVE OF POSTAGE, BUT CLEARLY STATING THAT IT WAS

AVAILABLE AS A FREE DOWN LOAD FROM THE WEB SITE!) I HOPED THAT IT WOULD SERVE AS A BIT OF PUBLICITY WHILE THE ADVERT RAN. AT 5P FOR THE FIRST INSERTION, EBAY OFFERED ME ANOTHER GO AS IT HADN'T SOLD, SO I HAD TWO GOES AT IT (LAST OF THE BIG SPENDERS EH!). I'VE NOTICED A MODEST INCREASE IN THE DOWNLOADS OF TRS8BIT DURING OCTOBER. I'LL PERHAPS HAVE ANOTHER GO WITH THE XMAS ISSUE, HOPING TO ATTRACT ANY PEOPLE NEW TO THE TRS-80 SCENE WITH THEIR RECENT PURCHASES FROM EBAY.

IF YOU GET A CHANCE, PLEASE CHECK-OUT YET ANOTHER UPDATE FROM MIGUEL DUTRA ON HIS VIRTUAL DISK READER. (V0.5). IT'S QUITE AMAZING. THERE'S A LINK ON THE WEBSITE FOR YOU TO USE.



IN THIS ISSUE, THERE'S PART 3 OF PETER STONES SUPER RE-LEARNING Z80 ASSEMBLER SERIES. DON'T FORGET, PETER HAS KINDLY OFFERED TO ASSIST ANYONE WITH A Z80 ASSEMBLER PROBLEM. HIS EMAIL ADDRESS IS ON THE WEBSITE. UNBELIEVABLE BUT TRUE EH; 'ON-LINE' SUPPORT FOR 30 YEAR OLD SYSTEMS; SURELY THIS MUST BE A U.K. FIRST? (I UNDERSTAND THAT PETER'S A WIZARD WITH HARDWARE PROBLEMS TOO!)

THERE'S A STANDARD BUT NONETHELESS VERY USEFUL BIT OF ADVICE FROM TANDY, TO ENABLE YOU TO USE TRS-80 PRINTERS FROM WITHIN WINDOWS. I JUST LOVE THE LAST PARAGRAPH-TANDY AT ITS BEST!

TALKING OF USING PRINTERS; ONE OF THE MOST ANNOYING PROBLEMS WHEN PRINTING NAME AND ADDRESS LABELS ON A LAZER PRINTER IS THE FACT YOU CAN'T PRINT SINGLE ONES, THEY NEED TO BE ON AN A4 SHEET. WELL THANKS TO WOODY POPE, THERE'S A GREAT ONE-LINER, TO ENABLE YOU TO PRINT INDIVIDUAL LABELS ON YOUR OLD DOT MATRIX PRINTER TO YOUR HEARTS CONTENT. I HAD GREAT FUN TRYING IT ON MY PC, USING MATTHEWS MODEL 1 EMULATOR. IT REALLY IS A MOST USEFUL UTILITY.
I DID ENCOUNTERED A PROBLEM WITH THE MICROLINE M80 FREEZING UP AFTER ONE PRINT JOB, WHICH I HAVEN'T SORTED AS YET. I'D JUST CONTACTED MATTHEW FOR HELP WHEN MY HARD DISK WENT WEST AND I'VE NOT HAD A CHANCE YET TO CONTINUE WITH IT.

AS PROMISED IN THE LAST EDITION, I MANAGED TO TRACE THE ARTICLE BY DON BANNISTER EXPLAINING HOW TO PRINT CONTROL CHARACTERS AND CHR$(0) ON AN EPSON PRINTER. HE MENTIONS USING ND80'S 'ROUTING' COMMAND, SOMETHING I'VE NEVER DONE!

AS THIS IS THE XMAS ISSUE, I FELT WE NEEDED THE USUAL XMAS PRINT-OUT, SO THIS YEAR, IT'S FEATURING STEVE MCQUEEN WITH AN ICONIC 1960'S POSE.
YOU'LL NEED TO SQUINT A BIT THOUGH!! I'VE INCLUDED A

CONTINUED ON PAGE 6

TRS-DOS PRINTERS AND WINDOWS

THIS DOCUMENT CONCERNS THE USE OF THE PRINTERS LISTED BELOW ON PC COMPATIBLE COMPUTERS USING MICROSOFT WINDOWS.

PRINTER	CATALOGUE NO.
DWP-2	(260-1158)*
DWP-210	(260-1257)
DWP-220	(260-1278)
DWP-410	(260-1250)
DWP-510	(260-1270)
LINE PRINTER 1	(260-1150)
	(260-1152)
LINE PRINTER 2	(260-1154)
LINE PRINTER 3	(260-1156)
LINE PRINTER 4	(260-1159)
LINE PRINTER 5	(260-1165)
LINE PRINTER 6	(260-1166)
LINE PRINTER 7	(260-1167)
LINE PRINTER 8	(260-1168)
DMP-100	(260-1253)
DMP-105	(260-1276)
DMP-110	(260-1271)
DMP-120	(260-1255)
DMP-200	(260-1254)
DMP-400	(260-1251)
DMP-420	(260-1267)
DMP-500	(260-1252)
DMP-2100	(260-1256)

THE PRINTERS LISTED ABOVE ONLY ACCEPT TANDY CONTROL CODES, AND WERE ORIGINALLY NOT DESIGNED TO BE PC COMPATIBLE. WINDOWS 95/98 WAS DESIGNED SPECIFICALLY AS A GRAPHICS USER INTERFACE, AND DOES NOT HAVE DRIVERS FOR NON-GRAPHICS TYPE PRINTERS, SUCH AS DAISY WHEELS. IN ADDITION, AS NONE OF THESE PRINTER EMULATE (ACCEPT THE SAME CONTROL CODES AS) ANY PRINTERS THAT ARE SPECIFICALLY SUPPORTED BY MICROSOFT WINDOWS, YOU MUST CHOOSE THE DRIVER "GENERIC TEXT/ONLY" WHEN USING THEM WITH WINDOWS AND WINDOWS BASED SOFTWARE. THIS PRINTER DRIVER WILL ALLOW YOU PRINT TEXT, BUT NOT BE ABLE TO PRINT ANY SPECIAL FONTS, BOLD-FACE, UNDERLINE, SUPER OR SUBSCRIPTS OR AND OTHER SPECIAL CHARACTERS. IN ADDITION, YOU WILL HAVE VERY LIMITED PAGE SETUP CAPABILITIES, AND WILL BE LIMITED TO A STANDARD 8.5" X 11" PAGE FORMAT.

AS THESE PRINTERS WERE DESIGNED TO BE USED ONLY ON COMPUTERS USING EITHER THE TRS-DOS, AND XENIX OPERATING SYSTEMS, THEY ALL ARE DESIGNED TO INSERT A LINE-FEED BETWEEN EACH PRINTED LINE, AS THESE OPERATING SYSTEMS DO NOT LINE FEED AUTOMATICALLY.

THE MICROSOFT OPERATING SYSTEM DOES AUTOMATICALLY LINE FEED FOR EACH PRINTED LINE. THEREFORE, WHEN ATTEMPTING TO USE THESE PRINTERS WITH MICROSOFT WINDOWS 95 AND 98, THE PRINTED INFORMATION WILL BE DOUBLE-SPACED.

SOME OF THESE PRINTERS, SUCH AS THE TANDY DMP-400, HAVE A SWITCH THAT CAN BE USED TO ELIMINATE THIS EXTRA LINE FEED. OTHERS, SUCH AS THE DWP-210 DO NOT. THEREFORE, ON THE PRINTERS THAT DO NOT HAVE THIS SWITCH, YOU MUST SEND A CONTROL CODE TO THE PRINTER TO ELIMINATE THE EXTRA LINE FEED.

FOLLOW THE DIRECTIONS LISTED BELOW TO ACCOMPLISH THIS TASK:

1. FROM THE WINDOWS DESKTOP, CLICK ON START, THEN SETTINGS, THEN DOUBLE CLICK ON PRINTERS.

2. THE PRINTERS FOLDER

SHOULD APPEAR. IF THE GENERIC / TEXT ONLY PRINTER DRIVER HAS NOT YET BEEN INSTALLED, DOUBLE CLICK ON THE ICON LABELLED "ADD PRINTER". IF IT HAS, PROCEED TO STEP 10.

3. THE ADD PRINTER WIZARD DIALOG BOX SHOULD APPEAR. CLICK ON NEXT.

4. A LIST OF PRINTER MANUFACTURERS AND MODEL NUMBERS SHOULD APPEAR. IN THE LIST OF MANUFACTURERS, SELECT GENERIC. IN THE LIST OF PRINTERS, SELECT GENERIC / TEXT ONLY. CLICK NEXT.

5. A DIALOG BOX LISTING PRINTER PORTS SHOULD APPEAR. SELECT LPT1:, THEN CLICK ON NEXT.

6. A DIALOG BOX ASKING FOR A PRINTER NAME SHOULD APPEAR. IF YOU WISH TO LIST THE PRINTER BY NAME, YOU MAY TYPE IT IN NOW. IF NOT, YOU CAN LEAVE THE NAME GENERIC / TEXT ONLY AS THE PRINTER DRIVER NAME. YOU WILL ALSO SEE A CHECK BOX ASKING IF YOU WISH THIS PRINTER TO BE YOUR DEFAULT PRINTER. SELECT YES, THEN CLICK ON NEXT.

7. YOU WILL BE ASKED IF YOU WISH TO PRINT A TEST PAGE. SELECT YES, THEN CLICK ON FINISH.

8. WINDOWS WILL BEGIN COPYING THE FILES NEEDED FOR THE GENERIC / TEXT ONLY PRINTER DRIVER. IF YOUR WINDOWS 95/98 CAB FILES ARE ONE YOUR HARD DRIVE, IT SHOULD FIND THEM AUTOMATICALLY. IF THEY ARE NOT, YOU MAY NEED TO INSERT YOUR WINDOWS CD.

9. ONCE THE FILES ARE COPIED, THE TEST PAGE SHOULD PRINT. YOU SHOULD NOTE THAT THERE IS A LINE FEED BETWEEN EACH LINE. THAT IS BECAUSE THE CONTROL CODE TO PREVENT THE EXTRA LINE FEED HAS NOT YET BEEN ADDED. A DIALOG BOX SHOULD APPEAR, ASKING IF THE TEST PAGE PRINTED CORRECTLY. CLICK ON YES. YOU SHOULD BE RETURNED TO THE PRINTERS FOLDER.

10. RIGHT CLICK ON THE ICON FOR THE GENERIC / TEXT ONLY PRINTER DRIVER. IN THE MENU THAT APPEARS, CLICK ON PROPERTIES.

11. THE GENERIC / TEXT ONLY PROPERTIES DIALOG BOX SHOULD APPEAR. CLICK ON THE DEVICE OPTIONS TAB.

12. CLICK THE MOUSE IN THE DATA FIELD LABELLED BEGIN PRINT JOB, SO THAT THE CURSOR IS IN THAT FIELD.

13. PRESS THE <ESC> KEY ON YOUR KEYBOARD. "<ESC>" SHOULD APPEAR IN THE DATA FIELD.

14. HOLD DOWN THE <CTRL> KEY, AND PRESS THE LETTER <U> ON YOUR KEYBOARD. "↑U" SHOULD APPEAR, DIRECTLY AFTER THE "<ESC>". THE FIELD SHOULD NOW LOOK LIKE THIS

BEGIN PRINT JOB
<ESC>↑U

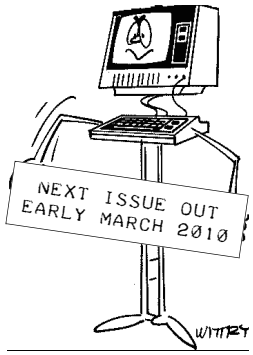
CLICK ON APPLY.

15. NOW, CLICK ON THE GENERAL TAB, AND CLICK ON THE BUTTON LABELLED "PRINT TEST PAGE". A NEW TEST PAGE SHOULD PRINT. THE LINE SPACING ON THE TEST PAGE SHOULD NOW BE

CORRECT.

* NOTE: FOR THIS TO WORK ON THE DWP-2 (260-1158), THE PRINTER MUST HAVE A SERIAL NUMBER LATER THAN 13010163. AT ONE TIME THERE WAS A BIOS ROM UPGRADE AVAILABLE FOR THIS PRINTER THAT WOULD ALLOW THIS PROCEDURE TO WORK, HOWEVER, THE UPGRADE IS NO LONGER AVAILABLE. (REF: TB I/O 32)

THIS DOCUMENTATION IS PROVIDED AS A COURTESY TO OUR CUSTOMERS, HOWEVER, TANDY/RADIO SHACK DOES NOT SPECIFICALLY SUPPORT NOR DO WE RECOMMEND USING THESE OLDER PRINTERS WITH ANY VERSION OF MICROSOFT WINDOWS.



PRINTING CHR$(0)

DON BANNISTER

AS PROMISED IN THE LAST EDITION, I'VE LOOKED OUT AN ARTICLE BY DON BANNISTER, WHICH FIRST APPEARED IN NATGUG NEWS, ABOUT LPRINTING CHR$(0). ROY BARBER, IN A PREVIOUS EDITION, COMMENTED THAT HE WAS HAVING PROBLEMS WITH HIS EPSON MX80 F/T WHEN PRINTING CHR$(0), SO DON TOOK UP THE CHALLENGE!

I TOO HAVE AN EPSON AND WENT THROUGH THE SAME TRAUMA!

THE PROBLEM, I THINK, IS THAT THE TRS-80 CAME ALONG BEFORE INTELLIGENT PRINTERS WERE READILY AVAILABLE AND TANDY PUT SOME CODE INTO THEIR PRINTER DRIVER TO SIMULATE FORM FEEDS, AND TO ALLOW VARIABLE LENGTH PAGES (REMEMBER POKE 16424,X). THE DRIVER TRAPS CHR$(N) WHERE N=0,11, OR 12. IN THE CASE OF 0, IT SIMPLY DOES NOTHING, AND HOW 11 AND 12 ARE DEALT WITH, DEPENDS ON THE CURRENT LINE COUNT AND LINES/PAGE SETTING.

THE PRINT DRIVER RESIDES IN ROM AT 058DH, AND DISASSEMBLING IT SHOWS THAT THE CHARACTER TO BE PRINTED IS PUT IN THE C REGISTER. IF THE CHARACTER IS 11 OR 12, THE PRINTER DCB (AT 4025H) IS EXAMINED TO SEE HOW MANY LINE FEEDS HAVE BEEN SENT TO SIMULATE A FORM FEED. IN THIS CASE OR WITH ANY OTHER NON-NULL CHARACTER, IF THE PRINTER IS FREE THE RELEVANT CHARACTER(S) ARE TRANSMITTED,

TO BYPASS THIS PART, ALL THAT IS NECESSARY IS TO LOAD THE ACCUMULATOR WITH THE VALUE IN THE C REGISTER (THIS REPLACES THE FIRST INSTRUCTION IN THE ORIGINAL) AND JUMP TO 05B4H, OMITTING THE TESTS FOR 0,11 AND 12. THIS NEW ROUTINE CAN BE PUT ANYWHERE IN MEMORY, AND THE ADDRESS IN THE DCB (4026/7H) AMENDED TO POINT TO IT.

THE ONLY PROBLEM IS TO FIND SOMEWHERE TO PUT THE NECESSARY 4 BYTES! IN NON-DISK LEVEL 2, SOME BYTES WHICH I BELIEVE ARE UNUSED ARE THOSE WHICH KEEP THE TIME AND DATE IN DISK

SYSTEMS. 4043H (16451 DECIMAL) SEEMS A SUITABLE PLACE, ONE ADVANTAGE BEING THAT THERE IS NO NEED TO PROTECT THESE LOCATIONS (AS WITH HIGH MEMORY). FOR MACHINE CODE PROGRAMMERS, THE NECESSARY (RATHER TRIVIAL) CODE IS SHOWN BELOW, BUT THE RELEVANT POKE'S ARE AS FOLLOWS :-

```
THIS IS THE NEW CODE -
POKE 16451,121:
POKE 16452,195:
POKE 16453,180:
POKE 16454,5
THIS REDIRECTS THE DCB TO
THE NEW DRIVER
POKE 16422,67:
POKE 16423,64
```

IF THE CODE IS PUT SOMEWHERE ELSE, THE LAST 2 POKES ARE WORKED OUT IN THE SAME WAY AS IS MENTIONED IN THE LEVEL 2 MANUAL FOR WORKING OUT THE ENTRY ADDRESS TO THE USR ROUTINE.

INCIDENTALLY, TO REVERT TO NORMAL, ONLY THESE LAST 2 NEED BE ALTERED, VIZ :-

```
POKE 16422,141:
POKE 16423,5
```

FOR DISK USERS, THERE ARE MORE OPTIONS. THE ONE I USE MYSELF IS TO USE ND80'S ROUTE COMMAND. THIS HAS THE ADDED ADVANTAGE THAT THE ROUTING CAN BE QUICKLY CHANGED OR DISSOLVED AT ANY TIME. ANOTHER ROUTINE USING THIS TECHNIQUE IS ONE TO PAGINATE PROGRAM LISTINGS.

FROM PAGE 2

COPY OF THE USUAL BASIC PROGRAM I USED, JUST TO SHOW YOU THE NEW DATA STATEMENTS. IT WORKS FINE FOR DISK AND 16K CASSETTE SYSTEMS ON MY M1 EMULATOR. ALL YOU NEED IS A PRINTER. IF ANYONE TRIES IT OUT ON A MODEL 3 OR 4, PLEASE LET US KNOW HOW YOU GET ON.

THERE'S BEEN ANOTHER HARDWARE ADDITION TO THE MODEL 100. AVAILABLE EXCLUSIVELY THROUGH THE CLUB100 WEBSITE, IT'S THE NEW 'REX' 32K ROM/RAM ADD ON. IT'S A MIND-BLOWING PIECE OF KIT. IF YOU MANAGE TO GET YOUR HANDS ON ONE, PLEASE LET US ALL KNOW HOW YOU GET ON WITH IT.
WITH THE REX AND THE NADS BOX, THE M100 IS A SERIOUSLY USABLE BIT OF KIT, EVEN IN THE 21ST CENTURY.

WELL, I THINK THAT ABOUT ROUNDS UP THIS, OUR 12TH ISSUE. IT'S BEEN GREAT TO HEAR FROM EVERYONE, WITH ALL YOUR KIND COMMENTS AND WORDS OF ENCOURAGEMENT, OVER THE LAST 3 YEARS. I HOPE WE CAN CARRY ON FOR ANOTHER 3!

SO, UNTIL THE NEXT ISSUE, DUE OUT MID MARCH 2010 -



TAKE CARE

DUSTY

```
                    ;EPSON-PRINT CHR$(0)
       ORG  4026H   ;IN PRINTER DCB
       DEFW ENTRY   ;SET DCB POINTER TO NEW DRIVER
       ORG  XXXXH   ;ANY SUITABLE FREE SPACE
ENTRY  LD   A,C     ;REPLACE INSTRUCTIONS IN ROM
       JP   05B4H   ;BACK TO ROM DRIVER
       END
```



FOR THOSE OF YOU WHO ACTUALLY RECOGNISE THIS ICONIC POSE, THE PROPER IMAGE IS ON PAGE 8. IN ALL FAIRNESS THOUGH, IF YOU STAND WELL ENOUGH BACK AND REALLY SQUINT HARD IT AINT TOO BAD. WELL, THAT'S MY OPINION; ANYWAY-HAPPY CHRISTMAS EVERYONE-IT IS THE THOUGHT THAT COUNTS EH?

```
10 CLEAR500
20M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMER
RYXMASMERRYXMAS"
30 GOSUB 200
40 READ S
50 IF S=0 GOSUB 150
60 READ L
70 T$=MID$(M$,S,L)
80      X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256
90 REM  ** THE NEXT LINE IS NOT NEEDED ON 16K SYSTEMS **
100      IF X>32767 THEN X=X-65536
110          FOR Y=1TOL
120          POKE X+S+Y,ASC(MID$(T$,Y,1))
130          NEXT Y
140 GOTO 40
150 LPRINT P$
160 GOSUB200
170 READ S
180 IF S=99 THEN END
190 RETURN
200 P$=STRING$(63,32)
210 RETURN
300 DATA 7,6,0,6,5,13,3,19,1,0,5,4,9,7,19,3,60,3,0
310 DATA 4,20,59,4,0,3,19,54,4,59,4,0,3,19,54,9,0
320 DATA 3,21,55,8,0,2,13,21,5,27,2,55,8,0,1,6,9,3,23,6,55,8,0
330 DATA 1,5,23,6,56,7,0,1,5,23,7,56,7,0,2,4,24,7,56,7,0
340 DATA 3,3,10,12,25,7,57,6,0,3,3,8,16,24,8,58,5,0
350 DATA 3,21,24,6,59,4,0,4,18,25,6,60,3,0,4,9,14,6,25,5,0
360 DATA 5,8,17,7,26,3,0,5,7,17,2,21,3,27,3,0
370 DATA 5,7,12,8,21,3,29,3,0,7,15,25,4,29,3,61,2,0
380 DATA 8,10,19,3,24,4,29,4,60,3,0,8,15,25,10,54,9,0
390 DATA 9,14,24,39,0,9,54,0,11,53,0,12,52,0,13,51,0,10,54,0
400 DATA 9,54,0,7,56,0,7,56,0,6,29,36,27,0,6,29,36,27,0
410 DATA 6,29,37,26,0,6,30,37,26,0,6,29,37,26,0,6,29,37,25,0
420 DATA 6,29,37,24,0,6,30,37,23,0,6,51,0,6,51,0,6,31,38,21,0
430 DATA 6,31,38,22,0,4,33,38,22,0,4,53,0,4,53,0,3,54,0,3,54,0
440 DATA 99
```



REX FOR THE M100/02

REX ... THE ULTIMATE OPTION ROM/RAM ADD-ON. UNTIL NOW, MODEL 100/102 OWNERS HAD ONE BANK OF 32K RAM AND COULD ONLY USE ONE OPTION ROM AT A TIME. THOSE DAYS ARE OVER. REX PLUGS INTO THE OPTION ROM SOCKET, AND VIA ITS USER FRIENDLY MANAGEMENT OS, YOU MAY SET UP 16 OPTION ROM BANKS TO HOLD ALL AVAILABLE OPTION ROM CODE, AND HAVE 16 BANKS OF 32K EACH. IT'S LIKE HAVING 16 MODEL "T"S IN ONE PACKAGE. IN A WORD: HEAVEN!

REX IS AVAILABLE EXCLUSIVELY THROUGH CLUB100.

REX IS COMPLETELY PLUG AND PLAY FLASH MEMORY UPGRADE. REX PLUGS INTO THE OPTION ROM SLOT, WITH NO SOLDERING OR NECESSITY TO OPEN THE LAPTOP CASE.
REX GETS IT'S NAME BY BEING A CROSS BETWEEN MY OTHER FLASH MEMORY DEVICE, REMEM, AND THE ORIGINAL OPTION ROM (RAM) MEMORY CARD, EXTRAM.

THE PRIMARY FEATURES OF REX ARE:
REX ALLOWS FOR QUICK AND EASY SWITCHING BETWEEN OPTION ROM IMAGES. ONCE THE BINARY SOFTWARE IMAGES ARE LOADED INTO REX, THE USER CAN EASILY JUMP BACK AND FORTH BETWEEN DIFFERENT APPLICATIONS. 16 OPTION ROMS MAY BE LOADED, STORED AND USED.

REX PROVIDES THE ABILITY TO BACKUP AND SWAP WORKING RAM IMAGES. AT THE PRESS OF A KEY, YOUR CURRENT RAM CONTENTS ARE SAVED IN REX. FURTHERMORE, YOU CAN RELOAD ANY SAVED RAM IMAGE FOR USE. THIS CREATES A LIBRARY OF RAM IMAGES, WHICH HAS A NUMBER OF USES. THIS OFFERS AN EASY WAY TO QUICKLY BACKUP ALL RAM TO FLASH, OR SWITCH BETWEEN MULTIPLE SNAPSHOTS OF RAM. THIS IS SIMILAR TO THE "FAST SWITCH ENVIRONMENTS" ON REMEM: IT IS LIKE HAVING MULTIPLE LAPTOPS IN THE SAME CASE! 16 RAM IMAGES MAY BE STORED.

REX INCLUDES AN INTERFACE TO TPDD DEVICES (LIKE TPDD, NADSBOX, LADDIECON ETC) FOR LOADING AND SAVING BINARY IMAGES TO EXTERNAL STORAGE.

IN ADDITION, REX IS IN-CIRCUIT, SOFTWARE REPROGRAMMABLE. THAT MEANS YOU CAN CHANGE, PATCH, OR REPLACE AN OPTION ROM IMAGE WITHOUT ANY SPECIAL HARDWARE OR PHYSICALLY UNPLUGGING REX.

FOR ADVANCED USERS, REX PROVIDES SOME ADDITIONAL FEATURES:

REX DISTINGUISHES BETWEEN TRADITIONAL OPTION ROM SOFTWARE, AND ALTERNATIVE OS SOFTWARE WHICH RUNS FROM THE OPTION ROM MEMORY LOCATION. THIS ALLOWS FOR SOME INTERESTING FUTURE CAPABILITIES LIKE CP/M OR A FORTH BASED SYSTEM.

REX ALSO SUPPORTS "MAIN ROM REPLACEMENT". THIS MEANS THAT, WITH THE ADDITION OF A SINGLE WIRE, THE MAIN ROM ON THE LAPTOP MAY BE REPLACED BY A VERSION STORED IN REX. THIS ALLOWS FOR FLASH-UPGRADEABLE MAIN ROM IMAGES, TO SUPPORT PATCHES, CUSTOM MENU AND CHARACTER SETS, LOMEM PATCH, Y2K PATCH ETC.

REX MAY BE INSTALLED IN MODEL 100 AND TANDY 102 CURRENTLY. TANDY 200 SUPPORT IS NOT YET AVAILABLE BUT IS IN DEVELOPMENT.

TINY TYPER

WOODY POPE

THIS PROGRAM BEGAN ACCIDENTALLY WHEN MY FRIEND, JAMES TALLEY, MADE A COMMENT TO ME ONE SATURDAY AT THE DALTRUG (DALLAS TRS-80 USERS GROUP) MEETING. WE WERE TALKING ABOUT WORD PROCESSOR PROGRAMS AND JAMES SUGGESTED I WRITE A ROUTINE THAT TURNS THE KEYBOARD AND LINE PRINTER INTO A TYPEWRITER. LATER, AT HOME, I WAS PLAYING AROUND AT THE KEYBOARD AND DECIDED TO GIVES JAMES'S IDEA A TRY.

FIRST CAME A 12 LINE PROGRAM WHICH PRINTED A LINE ACROSS THE TOP OF THE SCREEN AND WHEN <ENTER> WAS PRESSED, THE LINE WAS SENT TO THE PRINTER. OKAY, BUT THEM I GOT THE IDEA THAT MAYBE THE CODE COULD BE COMPRESSED INTO A SMALLER NUMBER OF LINES; IT REDUCED TO 5. I DECIDED TO TRY TO PUT IT ALL INTO 1 LINE, SIMILAR TO THE 'ONE LINERS' IN OTHER MAGAZINES. THIS TOOK AWHILE, BUT I FINALLY SUCCEEDED.

EVERYTHING WAS FINE UNTIL I DISCOVERED THAT EACH TIME THE ENTER KEY WAS PRESSED, THE CURSOR WOULD JUMP DOWN A LINE AND THE PRINTER DID A LINEFEED. CUTE I THOUGHT. I ENJOYED DOING THIS SO MUCH THAT I KEPT DOING IT UNTIL THE CURSOR WAS ON THE BOTTOM LINE OF THE SCREEN. THE NEXT ENTER BOMBED THE PROGRAM BECAUSE I HAD USED PRINT@+64. WHEN THIS NUMBER EXCEEDED 1023, THE PROGRAM BOMBED! SOMETHING HAD TO BE CHANGED.

WHY NOT PRINT THE LINE AT THE BOTTOM OF THE SCREEN AND LET IT SCROLL UP EACH TIME ENTER IS PRESSED? THEN THE TOP LINE WOULD BE SHIFTED UP AND OFF THE TOP JUST LIKE A REAL TYPEWRITER. A REWRITE WITH THIS FEATURE FIXED THE PROBLEM.

NEXT I DISCOVERED THAT A SPACE FOR THE BEGINNING CHARACTER ON A LINE CAUSED THE PROGRAM TO BOMB. A FIX WAS DISCOVERED, BUT ALL OF THE NECESSARY INFORMATION WOULDN'T FIT ONTO ONE LINE. A COMMA COULD BE SUBSTITUTED FOR 'THEN'. AND SINCE THE BASIC INTERPRETER COUNT CHARACTERS PER LINE, MORE ROOM WAS OBTAINED.

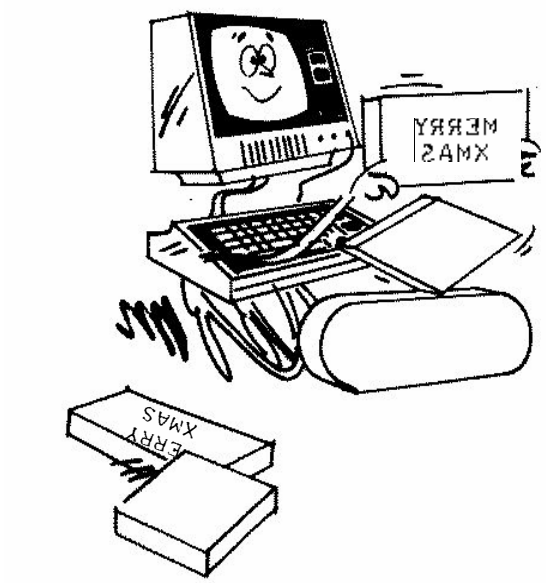
THE PROGRAM ALLOWS TYPING UP TO 63 CHARACTERS PER LINE BUT WILL NOT ACCEPT THE 64TH, FORCING THE OPERATOR TO PRESS ENTER. AN ASTERISK IS PLACED AT THE END OF THE LINE TO SHOW THE 64TH CHARACTER POSITION SO THAT THE OPERATOR WILL KNOW HOW MUCH SPACE IS LEFT ON THE LINE. THE BACKSPACE/ ERASE KEY IS ACTIVE AND ALL EDITING MUST BE DONE USING IT BEFORE ENTERING THE LINE. THE LINE WILL NOT GO TO THE PRINTER UNTIL ENTER IS PRESSED. IF DOUBLE SPACING IS DESIRED, THE OPERATOR JUST NEEDS TO PRESS THE ENTER AN EXTRA TIME BETWEEN LINES.

THE PROGRAM WORKS AS FOLLOWS :-

POKE 16382,42 PUTS AN ASTERISK AT THE LOWER LEFT OF THE SCREEN. IF F=0:CLS:CLEAR999:DEFSTR A-B:PRINT@960,CHR$(14);:F=1:GOTO 1 IS THE INITIAL SETUP. THE FIRST TIME THROUGH, F WILL BE ZERO, SINCE ALL VALUES ARE ZEROED AND STRINGS NULLED, WHEN YOU RUN A PROGRAM. NOTICE THAT

F=1:GOTO1 SETS THE F FLAG TO NONZERO THEN RETURNS TO THE BEGINNING OF THE LINE. THE ASTERISK WILL BE PRINTED BUT THE SETUP WILL BE SKIPPED. THE SCREEN IS CLEARED DURING SETUP AND STRING SPACE IS ASSIGNED. LETTERS A AND B ARE DEFINED AS STRINGS SO THAT THEY MAY BE WRITTEN IN THE LINE WITHOUT A $ AND THEREFORE, SAVE CHARACTER SPACE. PRINT@960,CHR$(14); TURNS ON THE CURSOR AT THE BOTTOM LEFT OF THE SCREEN.

A=INKEY$:IFA="",1ELSE D=ASC(A) FORMS THE KEYBOARD SCAN LOOP. IF NO KEY IS PRESSED, A WILL BE NULL AND THE PROGRAM GOES TO THE BEGINNING OF THE LINE (COMMAS REPLACE THEN'S), PRINTS OVER THE ASTERISK AND KEYSCANS AGAIN. WHEN A KEY IS PRESSED, THE ASCII VALUE IS FOUND AND ASSIGNED TO VARIABLE D.



IF D=13,PRINT A;:PRINTB:B="":C=0:GOTO1. IF THE ENTER KEY IS PRESSED, A SCROLL OCCURS, THEN THE STRING B IS PRINTED AND NULLED. THE CHARACTER COUNT IS ZEROED AND THE PROGRAM RETURNS TO THE BEGINNING OF THE LINE. A NEW ASTERISK IS PRINTED AND THE KEYBOARD WILL BE SCANNED AGAIN.

ELSE IF C=63,1: IF THE CHARACTER COUNT IS 63, DON'T PRINT, JUST RETURN TO THE BEGINNING OF THE LINE.

IF D=8,IF C=0, 1 ELSE B=LEFT$(B,LEN(B)-1):C=C-1:PRINT A;:GOTO1: IF THE BACKSPACE/ERASE KEY IS PRESSED AND THE CHARACTER COUNT IS ZERO, RETURN TO THE BEGINNING OF THE LINE. IF THE CHARACTER COUNT IS BETWEEN 1 AND 63, THEN CHOP OFF THE LAST CHARACTER OF THE STRING B, DECREMENT THE CHARACTER COUNT, PRINT THE BACKSPACE TO THE SCREEN AND RETURN TO THE BEGINNING OF THE LINE.

ELSE PRINT A;:B=B+A:C=C+1:GOTO1: PRINT ANY OTHER KEY NOT PREVIOUSLY DEFERRED TO THE SCREEN, ADD IT TO THE B STRING, INCREMENT THE CHARACTER COUNT AND RETURN TO THE BEGINNING OF THE LINE.

WHEN ENTERING THE PROGRAM LINE IN BASIC, YOU WILL NEED TO USE THE 'X' COMMAND WITHIN THE LINE EDITOR TO COMPLETE IT.

```
1 POKE16383,42:IFF=0,CLS:CLEAR999:DEFSTRA-B:PRINT@960,CHR$(14);:
F=1:GOTO1ELSEA=INKEY$:IFA="",1ELSED=ASC(A):IFD=13,PRINTA;:LPRINT
B:B="":C=0:GOTO1ELSEIFC=63,1ELSEIFD=8,IFC=0,1ELSEB=LEFT$(B,LEN(B
)-1):C=C-1:PRINTA;:GOTO1ELSEPRINTA;:B=B+A:C=C+1:GOTO1
```

ASSEMBLY LANGUAGE MADE EASY?

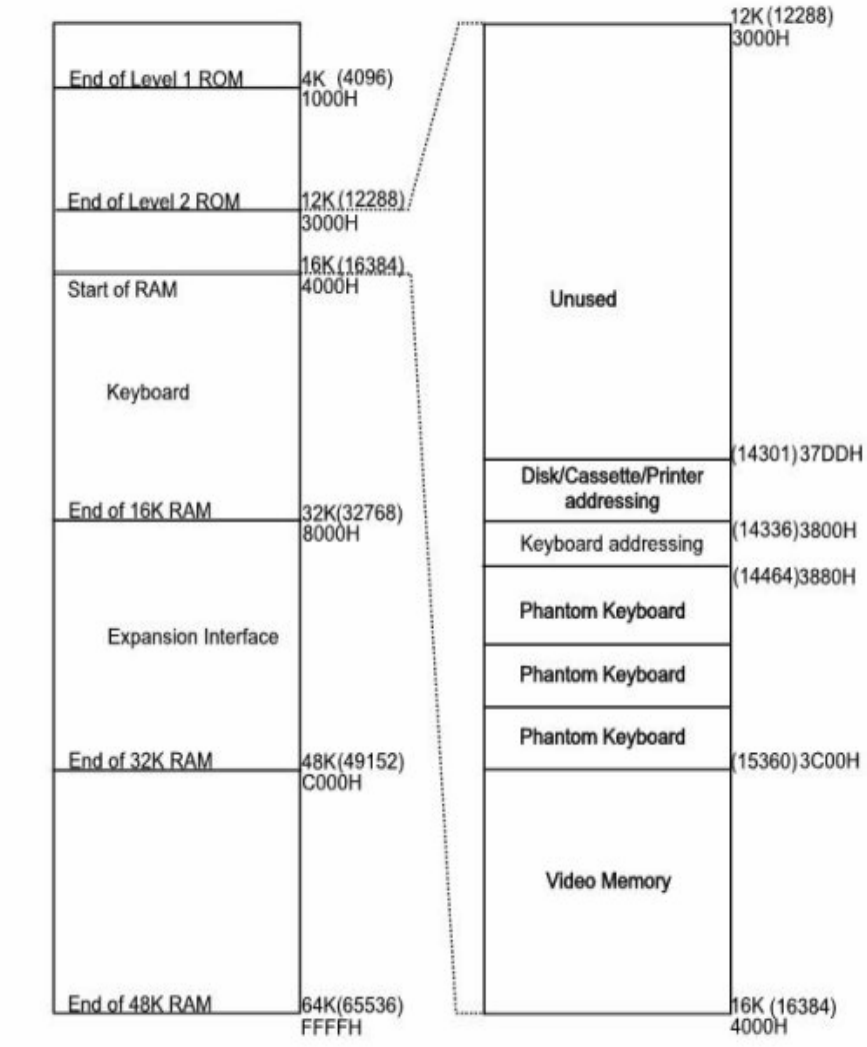
Peter Stone

Welcome to the third instalment in my series aiming to refresh your assembly programming skills, or alternatively if you've never programmed in assembler, tempt you to try it.

The focus of this instalment is going to be a discussion of the TRS80's memory organisation. This is partly due to a catastrophic & puzzling software crash on my system, which took a lot of time to sort out, & partly because I firmly believe that information is best shared & I'm concerned about the amount of information that has been either 'lost' or hoarded, & so want to get some of it back 'out there'.

Before I proceed with the article, some terms need to be defined. In all my articles, 1K equals 1024 bytes. Any number with a H after it is a hexadecimal number; if there is nothing after it, then it should be considered to be base10; if there is a B then it's binary; & an O after the number is octal.

The first thing I want to look at, is the TRS80's memory map, shown below. Just to refresh your memory, the Z80 has 16 address lines which means that it can address 64K of memory.



The 'tiny' BASIC that is level 1, fits into a 4k ROM & occupies the address space shown. When a keyboard has level 2 BASIC installed, it fits into three 4K ROMS, & occupies the address space up to 3000H. The block of memory from 3000H to 4000H is not physically connected to any memory, & is used as addresses for memory mapped devices. The memory from 4000H to 7FFFH is dynamic RAM, & is located in the keyboard unit. The dynamic RAM from 8000H to FFFFH is located in the expansion interface.

The area of memory from 3000H to 4000H is worth examining in greater detail, & is shown to the right of the diagram above. The area from 3000H to 37DDH is not used. In a physical TRS80, an EPROM could be wired in this space, giving you 2000 bytes which could contain drivers or other software. A commercial example of this, was the operating system for both the Exatron & Aculab Stringy Floppies. These used a value of 3039H (12345) as the entry point.

The area from 37DDH to 3800H contain a number of addresses used by the disk, cassette & printer devices. These are as follows.

37E0H - Drive Select. This address selects the required drive. i.e. Writing 01H will select drive 0, writing 02H will select drive 1, writing 04 will select drive 2, & writing 08H will select drive 3. The upper 4 bits are unused.

37E4H - Cassette Select. This address selects which of the two cassette drives to use. Writing 00H will select cassette 1 (default) & writing 01H will select cassette 2.

37E8H - Line Printer Data. Doing a read on this address will get the status of the printer. the values are 80H indicates that the printer is busy, 40H indicates that the printer is out of paper, 20H indicates that the printer isn't selected, & 10H indicates there is no fault. Doing a write to this address will cause the ASCII character associated with the byte to be printed. As a very simple example, the following code will print a character to the line printer.

```
;Load the ASCII character into the C register & then call prtdvr.
;
PRTDVR  LD   A,(37E8H)      ;CHECK STATUS
        BIT   7,A           ;BUSY BIT SET?
        JP   NZ,PRTDVR      ;LOOP WHILE BUSY
        LD   A,C            ;GET CHARACTER
        LD   (37E8H),A      ;SEND IT TO THE PRINTER
        RET
```

37ECH - Disc Controller Chip. Doing a read on this address, will get the status of the disk controller. Doing a write on this address, will issue commands to the controller. An example would be:

```
;Issue a reset to the disk controller chip (the value is D0H)
;
        LD   HL, 37ECH      ;LOAD HL WITH THE ADDRESS OF THE COMMAND REG
        LD   A,0DH          ;A CONTAINS THE VALUE OF THE RESET COMMAND
        LD   (HL),A         ;ISSUE COMMAND
        LD   C,(HL)         ;READ STATUS INRO REG C
```

37EDH - Track Select. A read on this address will get the track number of the current position of the read/write head from the Track Register. This address can also be written to.

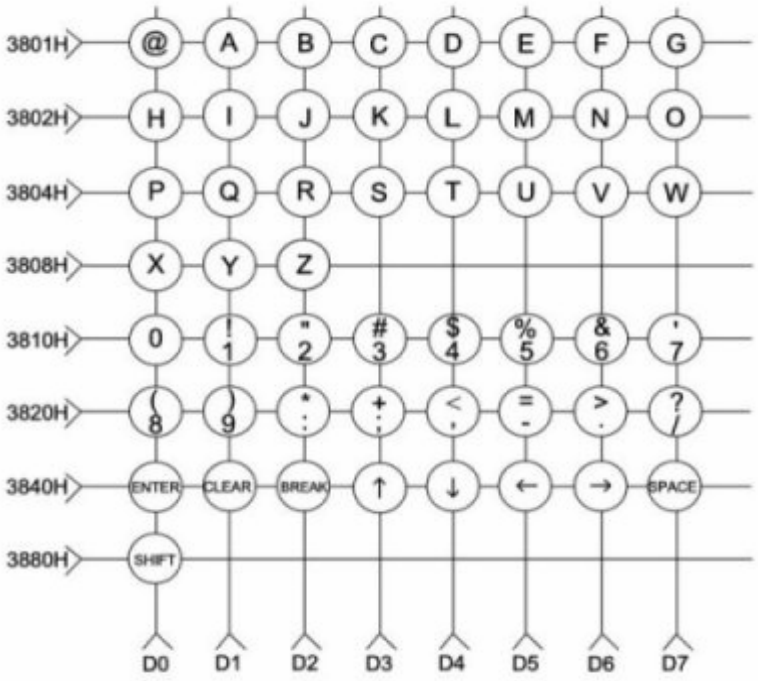
37EEH - Sector Select. A write to this address will load the desired sector to be read or written into the Sector Register. A read on this address will get the value contained in the Sector Register.

37EFH - Disk Data Address. This address contains the data byte to be written to the disk when a write command is issued, & the data byte read from the disk when a read command is issued.

I'm planning an article on how to program the disk controller for a future instalment, so I'll leave the more detailed explanations till then.

The area of memory from 3800H to 3880H is used to map the keyboard, (see diagram below). The more eagle eyed amongst you, will have noticed that the detailed memory map contains the legend phantom keyboard repeated 3 times. This is due to the fact that when the TRS80 keyboard circuitry was designed, the addresses were only partly decoded. An interesting side effect of this, is that any of the values shown, could be used to perform a keyboard read. (Though it is hard to think of an application when this might be required.)

For a reason why we might want to scan the keyboard directly rather than use the valid calls to the ROM, we only have to think of a space invaders type game. You use the Z & X keys to move left & right, & the space bar to fire. It we look at the diagram below, we would only have to use addresses 3808H & check if we get either 04H returned to move left, or 01H returned to move right, & use address 3840H & see if a value of 80H is returned to fire. Again, a more detailed explanation & usage will form the basis of a later article.



AS can be seen, the keyboard is made up of an 8 x 8 matrix, though not with every position filled. For example, if we use address 3802H & press the K key, we get the byte 08H returned, which can be decoded to the letter K. If we were to get the byte 1CH returned, then this would indicate that the J, K, & L keys had been pressed simultaneously, which could be used to activate a built-in DOS function. (i.e. minidos in Newdos80.) Before I finish with the keyboard, one thing that I must mention is key bounce. For those of us with long memories, we can recall typing hello & getting hhheeellllllooo. This is caused because the keyboard can be scanned in around 200 microseconds, whereas a fast press of a key might take 50 milliseconds, (for actual key closure), which means that the key could be read 50 times, while it's closed. To get around this a 'debounce' subroutine would be used. A further note, in the diagram above, only one shift key is shown. this is because the right & left shift keys were wired together in parallel.

The area from 3C00H to 4000H is the video memory. In actual component terms, the video memory consisted of seven 2102 1K x1bit static RAM chips. These stored bits 0, 1, 2, 3, 4, 5, & 7 of the characters displayed on the screen. Omitting bit 6 meant that both upper & lower case letters were displayed as uppercase, though hardware kits became available to rectify this problem. In this issue of TRS8BIT there is a blank video worksheet, which will prove invaluable for following the explanation that follows. The 1K memory is organised to provide a display of 16 rows of 64 characters, & under level 2 BASIC, can be addressed in a number of ways. The first way is to use the PRINT @ N, command, where n is a value between 0 & 1023. An example would be;

10 PRINT @ 0,"A" This would print the character A at the first position on the screen.

The next way is to use the POKE command. The syntax of this is POKE x,y. 3C00H converts to 15360 & 4000H converts to 16384. So an example would be;

10 POKE 15360,"A" Again, this would print A at the first position on the screen.

The first two articles I wrote have shown how to achieve the same thing in machine code. There is a third set of commands in BASIC to manipulate the video memory. These are SET & RESET, & are for compatibility with level 1 BASIC. If you examine the video worksheet you will see that each of the memory locations shown, is subdivided into two columns & three rows. This gives a grid of 128 (0-127) columns & 48 (0-47) rows. We can use SET (x,y) & RESET (x,y) to control each of these individual elements.

10 SET (3,6) would turn on pixel 3,6 (see video worksheet), while

10 RESET (3,6) would turn off pixel 3.6.

Before I go any further, I need to explain the ASCII character map & how it is implemented on the TRS80. The codes & their values are shown below.

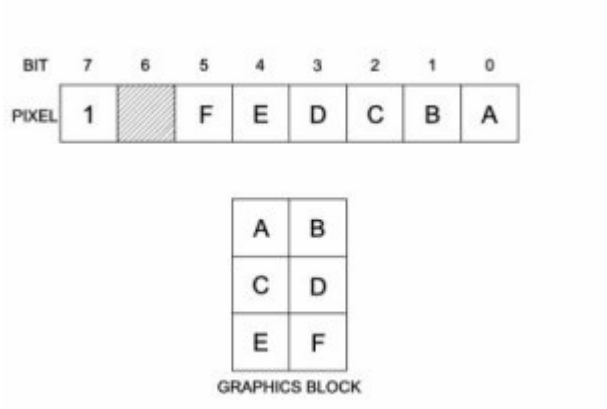
Code	Code	Code						
Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	NULL	32	20	SPACE	64	40	@
1	01	BREAK	33	21	!	65	41	A
2	02	STX	34	22	“	66	42	B
3	03	ETX	35	23	#	67	43	C
4	04	EOT	36	24	$	68	44	D
5	05	ENQ	37	25	%	69	45	E
6	06	ACK	38	26	&	70	46	F
7	07	BEL	39	27	‘	71	47	G
8	08	BKSP	40	28	(	72	48	H
9	09	HT	41	29	)	73	49	I
10	0A	LF	42	2A	*	74	4A	J
11	0B	VT	43	2B	+	75	4B	K
12	0C	FF	44	2C	,	76	4C	L
13	0D	CR	45	2D	-	77	4D	M
14	0E	CURON	46	2E	.	78	4E	N
15	0F	COROFF	47	2F	/	79	4F	O
16	10	DLE	48	30	0	80	50	P
17	11	DC1	49	31	1	81	51	Q
18	12	DC2	50	32	2	82	52	R
19	13	DC3	51	33	3	83	53	S
20	14	DC4	52	34	4	84	54	T
21	15	NAK	53	35	5	85	55	U
22	16	SYN	54	36	6	86	56	V
23	17	ETB	55	37	7	87	57	W
24	18	CAN	56	38	8	88	58	X
25	19	EM	57	39	9	89	59	Y
26	1A	SUB	58	3A	:	90	5A	Z
27	1B	ESC	59	3B	:	91	5B	Up arrow
28	1C	HOME	60	3C	<	92	5C	Down arrow
29	1D	BOL	61	3D	=	93	5D	Left arrow
30	1E	EREOL	62	3E	>	94	5E	Right arrow
31	1F	EREOF	63	3F	?	95	5F	DEL

Although all the possible control codes (0-31) are shown, only certain ones are actually used, these are shown below, along with what they actually do on a TRS80.

Dec Code	Hex Code	Function
0-7	00-07	None
8	08	Backspace & erases current character
9	09	None
10	0A	Line feed with carriage return
11	0B	Move carriage (cursor) to top of form (page)
12	0C	Move carriage (cursor) to top of form (page)
13	0D	Line feed with carriage return
14	0E	Turns on cursor
15	0F	Turns off cursor
16-22	10-16	None
23	17	Converts to 32 character/line mode. (Certain features work differently in 32 cpl. CHR$(25) spaces only one half space to the right.)
24	18	Backspace □ cursor
25	19	Advance □ cursor
26	1A	Downward □ linefeed (not applicable to the line printer)
27	1B	Upward □ linefeed (not applicable to the line printer)
28	1C	Home, return cursor to display position (0,0). Also sets display to 64 cpl.
29	1D	Move cursor to beginning of line
30	1E	Erase to end of line
31	1F	Erase to end of frame

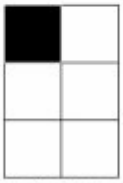
The actual displayable characters start at character 32. The chart only shows characters up to character 95, as characters 96-127 (60-7F hex,) are simply the lowercase equivalents to 64-95 (40-5F hex.)

The characters from 128-191 are the graphics characters. The relationship between the six elements of the graphics block & the 6 lower order bits of a memory byte is shown below.



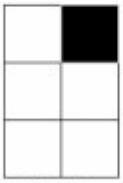
Quite simply if a bit is a 1, then the corresponding element in the graphics block is turned on. For example, if you were to use the BASIC command PRINT CHR$(128) a blank character would be displayed. Some examples of various PRINT CHR$(x) commands are shown below.

BIT	7	6	5	4	3	2	1	0
PIXEL	1		0	0	0	0	0	1



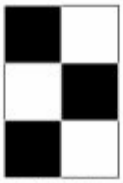
CHR$ 129

BIT	7	6	5	4	3	2	1	0
PIXEL	1		0	0	0	0	1	0



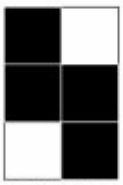
CHR$ 129

BIT	7	6	5	4	3	2	1	0
PIXEL	1		0	1	1	0	0	1



CHR$ 153

BIT	7	6	5	4	3	2	1	0
PIXEL	1		1	0	1	1	0	1



CHR$ 173

The final codes 192-255, (C0-FF Hex,) are space compression codes. These provide a compact means of representing a series of spaces from 0 to 63. For example, executing the BASIC command

10 PRINT CHR$ (192) would print zero spaces, while

10 PRINT CHR$(193) would print one space, &

10 PRINT CHR$(194) would print two spaces, up to

10 PRINT CHR$(255) would print 63 spaces.

Returning to our examination of the TRS80 memory map, we now arrive at the start of the RAM. However, we don't start programming at location 4000H. When a TRS80 is either switched on, or the reset button is pressed, an initialisation procedure is carried out, & the first 744 bytes of RAM are setup as an area known as the Communications Region. This is used to store various values required by the operating system, such as video cursor location, along with entry & exit points for various OS calls. I was going to provide a detailed breakdown of this area as part of this article, but I've decided to leave it for a later one. Because of this, the first available location used in a non-disk system to store a BASIC program is 42E9H. However, pressing the reset button causes material to be written into locations 4330H to 4348H, making 4349H the first free location for assembly language programs.

If a disk system is used, then things are different. As the TRS80 initialises, code in the level 2 ROM at location 0696H to 06C9H is executed. This detects if an Expansion Interface & disk drive is fitted to the system. If so, it reads the 256 bytes located at track 0 sector 0, & loads them into memory at 4200H to 42FFH. The code located in track 0 sector 0 has, like all other programs on the disk, a name & is called BOOT/SYS. However, it is unique from other programs in that it is an absolute core image file, created by writing a section of memory to the sector, allowing it to be read into memory & executed. Programs produced by an editor assembler package, have embedded codes indicating the load address, length of string to be loaded & an execution address. When BOOT/SYS has been loaded, then control is passed to it, & it seeks the existence of the SYS0/SYS file & if the file exists, loads it into memory at locations 4300H to 5200H. SYS0/SYS is the nucleus of the OS. This loads/overlays other SYS files into memory to provide other functions. The first free location for programs, not used by either the OS or the disk BASIC utility is 7000H.

At a future date, I'll explain the Communications Region in more detail.

Finally, if anyone has any topics they want me to cover, then please let me know by emailing me at **peter_stone45@hotmail.com**.

---== oØo ==---

E/Video Display Worksheet

E/1

TRS8BIT



WELCOME TO THE NEW DECADE AND THE FIRST EDITION OF 2010. THANKS FOR YOUR CONTINUED INTEREST AND SUPPORT.

A LITTLE BIT OF SAD NEW TO START OFF WITH. IT WAS WITH REGRET WE ALL HEARD THE NEWS OF THE DEATH OF STAN SLATER. HE WAS ONE OF THE ORIGINAL TEAM WHICH STARTED COMPUTER NEWS 80. I'M SURE BOTH HE AND HIS ONLINE SHOP, WILL BE SADLY MISSED BY ALL HIS COLLEAGUES AND FRIENDS.

I'VE HAD A GO AT TIDYING THE WEBSITE TO MAKE IT EASIER TO NAVIGATE THROUGH. THERE'S NOW A SEPARATE PAGE FOR DOWNLOADS AND I'VE STARTED A HELP, FOR SALE AND WANTS SECTION, SO IF THERE IS ANYTHING YOU'RE LOOKING FOR IN PARTICULAR OR ANYTHING YOU HAVE FOR SALE, JUST EMAIL ME WITH THE DETAILS AND I'LL GET IT PUT ON ASAP. NEEDLESS TO SAY, THERE IS NO CHARGE FOR THIS.

PETER STONE HAD COME UP WITH A COUPLE OF VERY INTERESTING IDEAS. HE'S SUGGESTED AN INDEX FOR THE FIRST 3 YEARS WORTH OF TRS8BITS. BELIEVE IT OR NOT, THIS HAD NEVER OCCURRED TO ME! I SHALL GET ON TO IT DURING MARCH AND HAVE IT AVAILABLE FOR DOWN LOADING AS QUICKLY AS I CAN. PETER'S ALSO ASKED ME TO POINT OUT THAT ANY ARTICLE, NO MATTER HOW SIMPLE, IS OF INTEREST TO THE GROUP. PLEASE DON'T LET WHAT YOU MIGHT THINK OF AS TRIVIAL, STOP YOU FROM LETTING US KNOW ANY IDEAS FOR ARTICLES OR REQUESTS FOR HELP YOU MAY HAVE.

EBAY, ONCE AGAIN HAS BEEN QUITE BUSY WITH ALLSORTS OF TANDY ITEMS OVER THE LAST COUPLE OF MONTHS. THE MOST NOTICEABLE WAS A M3 DISK UPGRADE AND A M3 POWER SUPPLY BOARD. I QUITE FANCIED THE DISK UPGRADE BUT IT WAS TOO MUCH FOR ME! IF I REMEMBER CORRECTLY, IT WENT FOR JUST UNDER £70. STILL NOT A BAD BUY THOUGH. IF THE LUCKY CHAP WHO BOUGHT IT READS THIS, LET US KNOW HOW YOU GET ON WITH IT.

MATTHEW REED HAS RELEASED A SUPER LITTLE UTILITY CALLED 'TRSTOOLS' (NOW AT V1.02). WITH THIS, YOU CAN NOW ACCESS AND AMEND 'DSK' FILE DIRECTLY FROM WINDOWS. THERE IS ALSO ANOTHER UPDATE TO HIS TRS32 EMULATOR AVAILABLE FROM HIS WEB SITE.

I NOTICED ON THE 'CLUB100' WEBSITE, THEY HAVE BEEN RUNNING A 29TH ANNIVERSARY PROGRAMMING COMPETITION. DID ANYONE IN THE UK HAVE A GO? (I STILL LOVE MY ONE-LINERS FOR THE M1). WAS THERE A CRAZE FOR THEM ON A M100? I NEVER MANAGED TO OWN ONE IN THE 80'S. TOO EXPENSIVE!

ANYWAY, DOWN TO THIS ISSUE. I HOPE YOU, AT LEAST, APPRECIATE THE SIZE OF IT, THIS IS THE LARGEST ONE YET! I WAS ASKED IF I HAD ANY EXPERIENCE OF CP/M. NOT A LOT, (IT ALL SEEMED BACK TO FRONT TO ME), BUT I MANAGED TO FIND AN OLD COMPARISON TABLE FOR MODEL 2 SYSTEMS WHICH I HOPE WILL BE OF INTEREST.

THERE'S THE USUAL BITS AND BOBS IN AT THE READY PROMPT, MAINLY DETAILS OF HOW TO TELL THE M1 AND M3 APART.

THERE'S A SMALL BUT USEFUL BASIC PROGRAM TO GIVE A SORTED DIR FOR ND80 V2 USERS.

THERE'S A COUPLE OF SNIPPETS ABOUT EARLY EMAIL AND SPEEDING UP A GENIE AND A ONE-LINER TO HELP NAME AND DATE A DISK UNDER ND80.

AN ARTICLE BY LEON HELLER, WHICH FIRST APPEARED IN NATGUG NEWS, TO ENABLE PASSING OF DOS PARAMETERS TO OTHER PROGRAMS. I CAME ACROSS THIS WHEN WORKING ON MY ARTICLE ABOUT GETTING THE MOST OUT OF YOUR LASER PRINTER. I THOUGHT IT WAS WELL-WORTH A REPRISE.

WHILE I'M ON ABOUT REPRISES, THE GERMAN CODE BREAKING MACHINE, ENIGMA, WAS THE SUBJECT OF AN ARTICLE BY CHRIS FARA AND BECAUSE I'D BEEN ASKED ABOUT ENCRYPTION OF TRANSMISSIONS TO THE OLD BULLETIN BOARD SYSTEMS, I THOUGHT YOU MIGHT FIND IT OF INTEREST. PETER STONE HAS OFFERED TO INVESTIGATE THIS AREA FURTHER IF ANYONE IS INTERESTED. (I'VE SAID, YES PLEASE, ON EVERYONE'S BEHALF!).

TALKING OF PETER, HE'S MANAGED TO EXCEL HIMSELF WITH PART 4 OF HIS RE-LEARN ASSEMBLER SERIES. HE'S DONE A MARVELLOUS JOB, I ONLY HOPE THIS LITTLE TOME IS WORTHY OF HIS EFFORTS.

CONTINUED ON PAGE 23 ...

AT THE READY> PROMPT



IN A PREVIOUS ISSUE, I MENTIONED A WAY TO TEST FROM WITHIN A PROGRAM AND CHECK IF YOU'RE RUNNING IN LEVEL 2 OR DISK BASIC (V01 NO.4 PAGE 2). I'VE BEEN ASKED IF IT'S POSSIBLE TO DETERMINE IF YOU'RE RUNNING ON A MODEL 1 OR A MODEL 3. WELL, I'M HAPPY TO REPORT THAT THE ANSWER IS YES! I UNDERSTAND THAT THE 'OFFICIAL' RADIO SHACK CHECKPOINT IS AT MEMORY LOCATION 293 DECIMAL (0125H). IF YOU PEEK AT THAT LOCATION AND IT'S 73 DECIMAL (0049H) YOUR PROGRAM IS RUNNING ON A MODEL 3. IF IT'S ANY OTHER NUMBER, YOUR PROGRAM IS RUNNING ON A MODEL 1. THE BASIC COMMAND IS X=PEEK(293):IF X=73 GOTO MODEL3 ROUTINE ELSE MODEL1 ROUTINE.

IF YOU'RE WRITING AN ASSEMBLY LANGUAGE PROGRAM THAT WILL RUN ON BOTH THE M1 AND/OR THE M3 AND WISH TO AVAIL YOURSELF OF MACHINE SPECIFIC ROUTINES, FROM THE ROM FOR INSTANCE, THEN THE FOLLOWING 7 BYTE ROUTINE WILL BE OF INTEREST (I HOPE!).

```
LD A,(1025H)    ;GET TEST CHAR
CP 49H          ;IS IT A M3?
JR NZ,MOD1      ;JUMP TO THE M1
                ;ROUTINE
                ;M3 ROUTINE
                ;START HERE
```

ANDY LEVINSON, ALSO POINTED OUT, IN AN ARTICLE IN 80-MICRO, A MORE EFFICIENT LOCATION TO CHECK (IF YOU'RE RUNNING A MACHINE LANGUAGE PROGRAM) IS 84 DECIMAL (54 HEX). IN A MODEL 1, THAT LOCATION ALWAYS CONTAINS A 1, BUT IT'S NEVER A 1 ON A MODEL 3.

BY TESTING THIS LOCATION, YOUR MACHINE CODE CAN BE ONE BYTE SHORTER! HERE'S THE CODE HE SUPPLIED TO CHECK THE TYPE OF MACHINE USING ONLY 6 BYTES.

```
LD A,(0054H)    ;GET
TEST            ;CHAR
DEC A           ;SEE IF ITS ONE
                  BY SUBTRACTING
                  ONE
JR Z,MOD1       ;GOTO M1
                  ROUTINE IF
                  ZERO
```

ANOTHER LITTLE SNIPPET I CAME ACROSS QUITE RECENTLY IN AN ADVERT FOR THE EXATRON. MOST PEOPLE ARE AWARE THAT EXATRON STRINGY FLOPPY WAFERS AND THOSE FROM THE ACULAB FLOPPY TAPE ARE COMPATABLE (SUBJECT TO THE POSITION OF THE WRITE-PROTECT REFLECTOR), BUT DID YOU KNOW THAT BOTH WERE COMPATABLE WITH THE OLIVETTI MTU-231, ROYAL AND ADLER TAPE DRIVES, THE STRUCTURE DESIGN SD-2024 PAL PROGRAMMER, AND WAFERS FOR THE TIMEX 1000 AND 1500?

AND FINALLY THIS MONTH, A LITTLE 'FACTOID' THAT WAS NEW TO ME (AFTER NEARLY 30 YEARS EH!). IF YOU HAVE A M1 L2 WITH THE NEW ROM, I.E. ON POWER UP IT ASKS 'MEM SIZE?' INSTEAD OF "MEMORY SIZE?' IT IS POSSIBLE TO PERFORM MULTIPLE PRINT@'S USING A SINGLE PRINT STATEMENT. FOR EXAMPLE;

```
PRINT@10,"A",@74,"B",@130,"C"
```

--=== o0o ===--

SORTED DIRECTORY

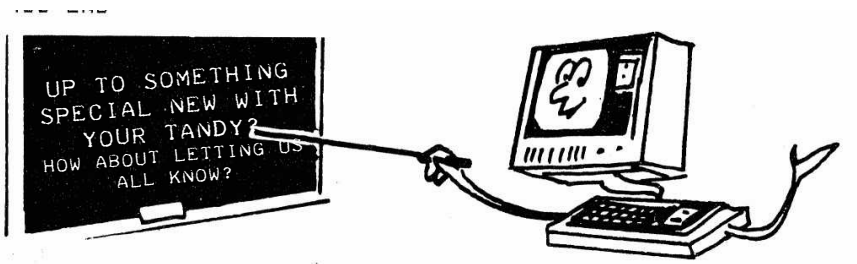
FROM AN ORIGINAL IDEA BY LAURIE SHIELDS

HERE'S A LITTLE PROGRAM FOR ND80 V2 USER WHO ARE PUZZLING ABOUT CMD"O". IT CALLS THE DIRECTORY, LOOK AT THE SCREEN TO SET THE FILENAMES INTO AN ARRAY, SORT THE ARRAY WITH CMD"O" AND THEN PRINTS THE DIRECTORY IN SORTED ORDER. IF YOU OVER-FILL THE SCREEN WITH FILENAMES, I DON'T KNOW WHAT HAPPENS!!

THE TECHNIQUE OF 'POKE'ING THE POINTERS OF A DUMMY VARIABLE TO PICK UP A STRING OF CHARACTERS FROM THE VIDEO IS MUCH FASTER THEN 'PEEK'ING AT EACH OF THE BYTES OF THE VIDEO AND BUILDING UP STRINGS BY CONCATENATION.

```
10 REM SDIR/BAS
15 CLEAR1000:AD=&H3C00+128:DIMDI$(56)
20 DX$="FILENAME/EXT":CMD"DIR :1":N=0
25 POKE VARPTR(DX$)+1,AD-256*(INT(AD/256))
30 POKE VARPTR(DX$)+2,INT(AD/256)
35 IFLEFT$(DX$,1)=" "THEN55
40 N=N+1:DI$(N)=DX$
45 AD=AD+15:IFN=4*INT(N/4)THENAD=AD+4
50 GOTO25
55 CMD"O",N,DI$(I):PRINT@128,CHR$(31);
60 FORJ=1TON:PRINTDI$(J),:NEXTJ
65 PRINT:END
```

--=== o0o ===--



'ELECTRONIC' MAIL

I THOUGHT THIS WAS AN INTERESTING LITTLE PIECE OF INFORMATION. THE FIRST TIME LEON HELLER, WHILST EDITOR OF NATGUG NEWS, RECEIVED AN ARTICLE SENT BY ELECTRONIC MAIL, FOR THE NEWSLETTER WAS IN JUNE 1981!
JUST FOR THE RECORD, IT WAS AN ARTICLE BY JOHN NEWGAS, REVIEWING THE HOLMES ENGINEERING 48K IN KEYBOARD EXPANSION. IT WAS SENT VIA VICTOR SALEH'S FORUM-80 COMPUTERISED BULLETIN BOARD SYSTEM. NOW, THAT'S NEARLY 29 YEARS AGO! JUST THINK HOW WE TAKE EMAIL FOR GRANTED THESE DAYS.
THIS, MOST PROBABLY, WASN'T A U.K. FIRST, BUT IT WAS THE START OF MANY WONDERFUL THINGS TO COME.

--== o0o ==--



HERE'S A LITTLE BASIC ONE LINER FROM AN ORIGINAL IDEA BY CL ROBERTSON, FOR ND80 USERS. IT SETS THE DISK NAME ON DRIVE 1, TO THE SUPPLIED NAME AND SETS THE DATE TO THE CURRENT SYSTEM ONE. IT WILL, OF COURSE, WORK FOR ANY DRIVE, JUST CHANGE THE NUMBER 1 TO WHICH EVER YOU NEED.

SPEED UP YOUR GENIE

THERE IS A VERY SIMPLE MODIFICATION WHICH WILL ENABLE VIDEO GENIE OWNERS TO USE THE 'KANSAS' 1 TO 3 SPEED-UP PROGRAM, ALLOWING THREE TIMES FASTER CASSETTE LOADING.
MANY GENIE OWNERS HAVE COMPLAINED THAT THOUGH IT IS POSSIBLE TO USE THE SPEED-UP PROGRAM TO GIVE 1500 BAUD RATE FOR THE TANDY, THE INCLUSION OF A FILTER ON THE GENIE MADE IT SO THAT THAT SYSTEM WOULD NOT ACCEPT THE HIGHER FREQUENCY OF THE 1500 BAUD RATE SIGNAL.

HERE'S HOW YOU GET ROUND IT ...

UNSCREW THE SCREWS UNDERNEATH THE GENIE, THEN TAKE OFF THE COVER.

ON THE INTERFACE BOARD, BY THE CASSETTE, IS A SOCKET AND PLUG WITH FOUR WIRES WHICH GO TO THE CASSETTE. JUST BELOW, RUNNING PARALLEL WITH IT IS A RESISTOR, NUMBER R28. SIMPLY CONNECT THE LOWER END OF THIS TO EARTH. THAT'S ALL THERE IS TO IT!

AS YOU WILL BE WANTING TO USE BOTH THE NORMAL 500 BAUD AND THE FAST 1500 BAUD RATES, JUST PUT A SWITCH IN CIRCUIT, SO THAT EITHER CAN BE SWITCHED IN.

WITH THIS VERY SIMPLE MODIFICATION, THE 1 TO 3 SPEED-UP PROGRAM WILL WORK PERFECTLY, AS IT DOES ON THE TANDY.

```
10 INPUT"DISK NAME ";N$:D$=LEFT$(TIME$,8):CMD"PROT,1,NAME="+N$:C
MD"PROT,1,DATE="+D$
```

PASSING PARAMETERS FROM DOS

LEON HELLER

THIS ARTICLE FIRST APPEARED IN NATGUG NEWS

A LITTLE USED FEATURE OF TRSDOS IS THE ABILITY TO PASS PARAMETERS TO A PROGRAM FROM DOS. THE TRSDOS MANUAL STATES THAT LOCATION 4318H IS THE START ADDRESS OF A 64 BYTE BUFFER CONTAINING THE LAST TRSDOS COMMAND (USUALLY A FILENAME) THAT WAS ENTERED. THE FOLLOWING ROUTINE PLACES THE WORD FOLLOWING THE FILENAME INTO A SECOND BUFFER. THE SECOND WORD COULD BE THE NAME OF A FILE TO BE INPUT TO THE MAIN PROGRAM, AND THE SECOND BUFFER COULD BE A DCB. FOR INSTANCE, COUNT FRED/DOC WOULD LOAD A WORD COUNT PROGRAM WHICH WOULD COUNT THE WORDS IN THE FILE FRED/DOC. THE ROUTINE CAN EASILY BE EXTENDED FOR ADDITIONAL PARAMETERS.

SOME COMMERCIAL PROGRAMS SUCH AS MACRO-80 AND LINK-80 USE THIS TECHNIQUE; M80 FRED=FRED WILL LOAD MACRO-80 AND ASSEMBLE THE SOURCE FILE FRED/MAC, PRODUCING THE RELOCATABLE OBJECT FILE FRED/REL. IT WOULD BE NICE IS THIS FEATURE WERE ADDED TO SCRIPSIT - SCRIP FRED/DOC WOULD LOAD SCRIPSIT AND A TEXT FILE CALLED FRED/DOC, SAVING A FEW KEYSTROKES. (HAS ANYONE TRIED THIS YET?)

THE TECHNIQUES BELOW CAN BE USED WHEREVER THERE IS A REQUIREMENT TO EXTRACT WORDS FROM A SENTENCE, IN ADVENTURE TYPE GAMES AND SPELLING CHECKERS FOR INSTANCE, OR TOKENS FROM STATEMENTS WHEN WRITING ASSEMBLERS AND COMPILERS.

```
GETPARAM:
    LD   HL,BUFF1     ;POINT TO START OF BUFFER
                      ;(4318H)
G1: LD   A,(HL)       ;GET CHAR IN A
    CP   CR           ;CARRIAGE RETURN?
    JR   Z,DONE       ;IF CR - DONE
    CP   SPACE        ;SPACE?
    JR   Z,G2         ;IF SPACE - JUMP
    INC  HL           ;NOT SPACE - GET NEXT CHAR
    JR   G1
G2: INC  HL           ;SKIP SPACE(S)
    LD   A(HL)
    CP   SPACE
    JR   Z,G2

;
;HL NOW AT START OF SECOND WORD - MOVE TO BUFF2
;HL=SOURCE
;

    LD   DE,BUFF2     ;DE=DESTINATION
G3: LD   A(HL)        ;MOVE WORD UNTIL CR OR SPACE
    LDI
    CP   CR           ;DONE IF CR OR SPACE
    JR   Z,DONE
```

--== o0o ==--

A QUICK LOOK-UP TABLE FOR M2 CP/M USERS. I THINK THIS FIRST APPEARED IN 80-US.

TABLE 1

Comparison of TRSDOS and CP/M

TRSDOS	Cybernetics	Lifeboat	P & T
AGAIN	***	***	***
ANALYZE	STAT	STAT	STAT
APPEND	PIP	PIP	PIP
ATTRIB	STAT	STAT	STAT
AUTO	***	***	***
BUILD	ED	ED	ED
CLEAR	***	***	***
CLOCK	***	***	***
CLS	***	***	***
COPY	PIP	PIP/COPY	PIP
CREATE	***	***	***
DATE	***	***	***
DEBUG	DDT	DDT	DDT
DIR	DIR	DIR	DIR
DO	SUBMIT	SUBMIT	SUBMIT
DUAL	(CTRL-P)	(CTRL-P)	(CTRL-P)
DUMP	SAVE	SAVE	SAVE
ECHO	(CTRL-P)	(CTRL-P)	(CTRL-P)
ERROR	***	***	***
FORMS	SET	CONFIG	SETMISC
FREE	STAT	STAT	STAT
HELP	***	***	***
HOST	***	***	***
I	(CTRL-C)	(CTRL-C)	(CTRL-C)
KILL	ERA	ERA	ERA
LIB	***	***	***
LIST	TYPE/DUMP	TYPE/DUMP	TYPE/DUMP
LOAD	***	***	***
MOVE	PIP	PIP	PIP
PAUSE	***	***	***
PRINT	TYPE/PIP	TYPE/PIP	TYPE/PIP
PROT	STAT	STAT	STAT
PURGE	ERA	ERA	ERA
RECEIVE	PIP	PIP	PIP
RENAME	REN	REN	REN
RESET	(CTRL-C)	(CTRL-C)	(CTRL-C)
SCREEN	***	***	***
SETCOM	SET	CONFIG	SETUP
SPOOL	***	***	***
STATUS	STAT	STAT	STAT
TIME	***	***	TIME
VERIFY	***	***	***

NOTE:

1. CP/M commands are often multi-purpose and very flexible so they often appear in more than one place.
2. This table was produced by simply listing the closest commands, transient or built-in, that could perform the task done by the TRSDOS command.
3. This table is based on the capabilities of the TRSDOS 2.0.

PRINT IT ALL!

NOW THIS STARTED AS A MINOR INCONVENIENCE, DUE MAINLY TO MY MEMORY OR RATHER THE LACK OF IT AND IT GREW, JUST LIKE TOPSIE! (WELL, IN USE-ABILITY IF NOT SIZE).
MY MAIN PRINTER IS AN HP LASERJET 2200DTN. I THINK IT'S THE BEST THING SINCE SLICED-BREAD. I PAID THE PRINCLEY SUM OF THREE POUNDS FOR IT, NEARLY 5 YEARS AGO AT A LOCAL CAR BOOT AND IT HAS PERFORMED FAULTLESLY SINCE DAY ONE. IT'S ONLY EVER REQUIRED TONER AND EVEN HERE, EBAY HAS COME TO THE RESCUE WITH NEW PACKS AT UNDER TWENTY POUNDS, ON THE FEW OCCASIONS I'VE NEEDED SOME.

HOWEVER, THE ONE SMALL IDEOSINCRACY I'VE NOTICED IS THAT, WHEN IN M1 BASIC, RUNNING WITHIN MATTHEW'S EMULATOR, ANY LPRINTING THAT HAS BEEN PERFORMED IS SENT TO THE PRINTER BUT NOT ACTUALLY PRINTED OUT UNTIL A FULL PAGE OF SAY, 66 LINES, IS PRESENT. THE EASIEST WAY ROUND THIS, WHILST IN BASIC, IS TO LPRINTCHR$(12) AND ALL IS WELL. WELL THAT IS, UNTIL YOU'VE FORGOTTEN AND ARE BACK IN DOS. SO BEING FED-UP OF RETURNING TO BASIC JUST TO COMPLETE THE PRINT OUT, I THOUGHT OF THE WONDERFUL IDEA OF ISSUING A SIMILAR COMMAND FROM THE DOS LINE. WITH PETER STONE'S EXCELLENT ARTICLES ON ASSEMBLER MEANT IT WOULD HAVE TO BE IN MACHINE CODE, JUST FOR THE PRACTICE.

'SIMPLES' I CAN HERE YOU ALL YOU M/C HACKERS MUTTER. IN FACT, IT WASN'T SO MUCH AS HOW TO DO IT, BUT WHICH, OF THE MULTITUDE OF ROM CALL AND PRINT CHARACTERS, TO USE!

FROM THE OUTSET, I DECIDED THAT IT WAS TO BE AS SIMPLE AND AS SMALL AS POSSIBLE, AS IT WAS MY INITIAL INTENTION FOR THE PROGRAM TO BE USED ONLY ONCE AT THE ND80 DOS PROMPT.
I ASSUMED THAT AN ORG OF 8000H SHOULD DO THE TRICK, THEN LOAD A FF CHARACTER INTO THE ACCUMULATOR AND USE A ROM CALL TO SEND IT TO THE PRINTER.

THIS WORKED FINE AND FOR QUITE A WHILE I WAS CONTENT. IN FACT, IT BECAME STANDARD PROCEDURE TO RUN FF/CMD EVERY TIME I EXITED FROM BASIC. THEN I GOT TO THINKING, (ALWAYS DANGEROUS), WHAT IF I EVER NEEDED TO RUN THE PROGRAM FROM WITHIN BASIC, OR I HAD TO RE-ENTER BASIC USING THE ND80 BASIC* COMMAND. THAT ORG AT 8000H DIDN'T LOOK SO HANDY. BUT, OF COURSE, ONCE YOU WANT THINGS TO RUN IN BASIC, THEY NEED TO BE IN MEMORY SOMEWHERE WHERE YOUR BASIC PROGRAM (AND MANY OTHER THINGS TOO) ISN'T! THE MOST OBVIOUS WAY WOULD BE TO RESERVE SOME MEMORY AT BASIC LOADING TIME BUT, WHAT A DRAG EH! I MEAN, IT'S ONLY 6 BYTES LONG, SURELY THERE'S SOME LITTLE 'HOLE' WHERE I CAN PLACE IT FOR THE ODD TIMES I'LL NEED IT.

THEN I REMEMBERED AN ARTICLE BY LEON HELLER, WHICH FIRST APPEARED IN NATGUG NEWS), FOR A PROGRAM TO PASS PARAMETERS TO PROGRAMS FROM THE DOS PROMPT. (I'VE MANAGED TO FIND THE ARTICLE AND IT'S REPRINTED ON PAGE 5).
ON PAGE 6-11 OF THE TRSDOS DISK REFERENCE MANUAL IT SAYS THAT THERE IS A 64-BYTE BUFFER THAT CONTAINS THE LAST TRSDOS COMMAND THAT WAS ENTERED. THIS SEEMED AN IDEAL PLACE TO PUT THE NECESSARY CODE AND SO THIS IS WHERE FF/CMD NOW RESIDES. NOT ONLY CAN IT BE RUN FROM THE ND80 DOS PROMPT, BUT ALSO, SHOULD YOU EVER NEED TO, FROM ND80 BASIC WITH THE COMMAND CMD"FF.

I DETAIL THE SOURCE PROGRAM BELOW, WHICH REALLY IS VERY TRIVIAL, BUT A WORD OF WARNING!! A LITTLE KNOWLEDGE CAN BE A DANGEROUS THING, AND BELIEVE ME, I'VE A VERY LITTLE KNOWLEDGE. SO DO TAKE CARE AND CHECK THAT THE CODE WILL NOT CLASH WITH ANY OTHER PROGRAM(S) YOU HAVE RUNNING ON YOUR SYSTEM.

LINE 3 SETS UP A CALL TO THE ROM ROUTINE AT 039CH TO SEND THE CHARACTER IN THE ACCUMULATOR TO THE PRINTER. WHY THIS CALL? WELL, AFTER CONSULTING ONE OF MY FAVOURITE AND MOST USEFUL BOOKS, LEVEL 2 ROM - ASSEMBLY LANGUAGE TOOLKIT BY EDWIN PAAY, IT SEEMS THAT THIS CALL DOES ALL THE HARD WORK FOR US IN THAT ALL THE REGISTERS ARE SAVED BEFORE USING IT. (ANYTHING FOR AN EASIER LIFE YOU KNOW!)

LINES 4,5 & 6 INFORM THE ASSEMBLER PROGRAM (WHICH, IN THIS CASE, WAS ZEN) WHERE THE 6 BYTES ARE TO WORK.
LINE 7 LOAD THE ACCUMULATOR WITH 0CH (13 DECIMAL), A FORM FEED.
LINE 8 CALL THE ROM ROUTINE TO SEND THIS TO THE PRINTER.
LINE 9 RETURN BACK TO THE CALLING ENVIRONMENT.

```
 1  ;FF PRINT
 2  ;
 3  PRINTER:  EQU  039CH
 4  ORG  4318H
 5  LOAD 4318H
 6  EXEC 4318H
 7  LD   A,0CH
 8  CALL PRINTER
 9  RET
10  END
```

--== o0o ==--

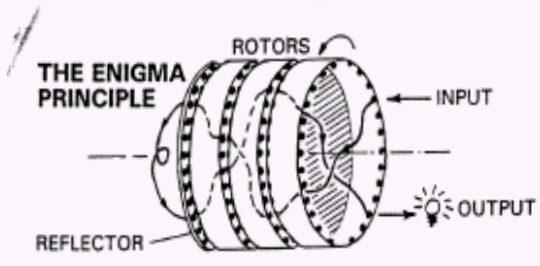
BASIC ENIGMA MACHINE

(C) 1992 - CHRIS FARA, MICRODEX
THIS ARTICLE FIRST APPEARED IN TRSTIMES



One of the most intriguing figures in the history of computers was the British mathematician Alan Turing (1912-1954). At the age of 24 he published a paper on some famous issue of algebra. But that issue seems less relevant today than the way Turing went about solving it. To prove his point, he described what essentially was a digital computer. The features of his "universal computing machine" (since then dubbed the "Turing machine", often mis-spelled "Touring") are taken for granted today, but were quite novel at that time: binary computing using only zeros and ones, conditional branching of program flow, sharing of computer memory by programs and data. He imagined "memory" as an impossibly long paper "ticker" tape. The implementation of his machine had to wait until electronics made it practical. All our computers today are "Turing machines".

Turing was (and still is) relatively little known, except in the rarefied circles of higher science. Even less known is his crucial role during World War II, in breaking the system of the "Enigma" machine used by Germans to encode military messages. That machine was a sort of 26-letter electric typewriter. When a key was pressed, current went to one of 26 input pins located on one rim of a drum-like wheel (rotor). Inside the rotor, the input pins were wired at random to 26 output pins on the opposite rim. These output pins passed the current to the input pins of the next wheel, and so forth. The original Enigma had 3 rotors, later expanded to 4 and even 5. The output from the last (inner) rotor went to a pin on a stationary plate called "reflector" where it was wired to another pin on that plate and thus was "reflected" as input to the backside of the inner rotor. From there it went back to the first (outer) rotor, and finally to one of 26 light bulbs that indicated the encoded letter.



After one letter was encoded, the outer rotor would make 1/26-th of the full turn, automatically changing the relation of pins. After the outer rotor made one full turn, the next rotor would advance by one pin, and so on, like an odometer. Thus each letter would be encoded by a different wiring path, and this made the Enigma so hard to crack. All breaking of encrypted messages is based on analyzing typical patterns of letters. It didn't work with the Enigma. Even with just 3 rotors, identical pattern could be only guaranteed after exactly 26*26*26 = 17,576 characters! The encoding also depended on the initial setting of the rotors in respect to each other at the beginning of a message. There were some other minor complexities. It all added up to millions of combinations.

Enigma was considered unbreakable, and Germans trusted it 100%. Yet the code was broken already before the war by a couple of Polish mathematicians who later passed the secret to the Allies. But when Germans added the 4th rotor, the original manual decrypting procedure proved too time consuming. An automated solution was the only hope. This fell right in line with Turing's vision of computing machines, and he eventually led a team that developed the "Colossus", arguably the first-ever modern computer, even though its mission was limited to the deciphering of the Enigma code.

So much for history. The Enigma machine was a bulky clanker stuffed with a jungle of wiring. The operator had to watch the light bulbs and jot down the letters one by one. Today, the Enigma can be easily implemented on any desktop computer (our "universal Turing machine") with a simple BASIC program. To see how it works, let's assume a simplified 6-letter alphabet symbolized by numbers 0-5. Let's also assume for now that our Enigma machine has only 2 rotors. The wirings can be written in a table:

	0	1	2	2'
0	4	1	5	0
1	2	3	0	1
2	1	4	1	4
3	5	5	4	2
4	0	2	2	3
5	3	0	3	5

Rows represent the 6 pins. Column 0 represents the wiring of the "reflector". For example row 3 has value 5 which means that the wire from pin 3 goes to pin 5. The wiring is "self-inversing": row 5 has value 3. Column 1 represents the random wiring of the "inner" rotor, column

2 is the "outer" wheel. Column 2-prime will be discussed in a moment. Now suppose we want to encode the digit 1. Starting from the outer rotor (column 2) in row 1 we find 0. Therefore output from the outer rotor goes to pin 0 (row 0) on the inner rotor, where we find 1. In the "reflector" row 1 routes it to pin 2. Now going back, we look for the "output pin" 2 in column 1. This is found in row 4. In the outer column we look for 4 and find it in row 3. Hence 1 has been encoded into 3.

After the first character has been transmitted, the outer rotor turns by one pin. Its new, shifted "wiring" position is shown in column 2-prime. If we now try to encode 1, then the result will be 4, not 3 (the sequence is 1-1-3-5-3-4). In the same manner we can trace the reverse encoding of 4 into 1 (4-3-5-3-1-1). This symmetry is the result of the self-inversing "reflector". Thus decoding of a secret message is the reverse of encoding, with the rotors set in the same starting position in both cases.

This entire process can be written in just a few lines of BASIC. Let's call our "wiring" table the array R(2,5). In addition, we'll need a small array S(2) to keep track of the setting of each wheel. The number to be encoded is passed to the routine in the variable N.

```
10 for x=2 to 1 step -1
11 N = R(x, ( N + S(x) ) mod 6)
12 next

13 N = R(0, N)

14 for x=1 to 2: for y=0 to 5
15 if N=R(x,(y+S(x)) mod 6) then N=y:y=5
16 next y, x

17 for x=2 to 1 step -1
18 S(x) = (S(x)+1) mod 6: if S(x) then x=1
19 next
```

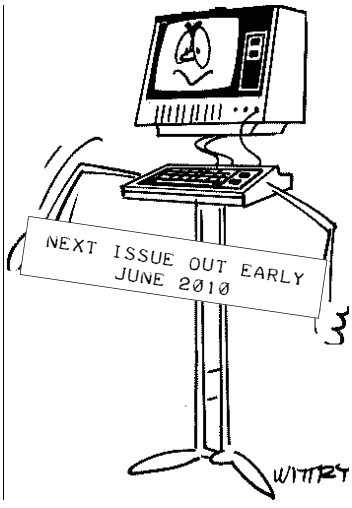
The first 3 lines encode the original value of N, starting from the outer rotor. Notice how the encoding depends on the setting S(x) for each rotor. Line 13 is the "reflector". Lines 14-16 pass the code back to the outer rotor, and yield the final output N. In the last 3 lines the "odometer" advances the rotor settings. That's it.

Actually, the "physical" Enigma behaved slightly differently from our computerized model. A rotation of any wheel not only shifted the pins in respect to the input, but obviously also shifted the output pins in respect to the next rotor. But that was an accidental property of the mechanical implementation of the system, irrelevant to the coding process, because the wiring of each rotor is random anyway. So we can skip that detail.

To make a complete working Enigma, DIMension the arrays for the desired number of rotors (columns) and characters (rows), fill in all cells with proper "wiring" numbers, adjust loop counters and "modulo" values, and provide for input and output. Or if you don't want to bother with programming, send me 5 bucks for a "Basic Enigma on the disk" (c/o Microdex, 1212 N.Sawtelle, Tucson AZ 85716; specify Mod-III TRSDOS or LDOS, Mod-4, or MSDOS). It's a neat game for kids and for war history buffs. Great for secret love letters, too.

After the war, Turing went on to build a general-purpose computer, and then turned to the mathematical principles of genetics and evolution. But the memory of Alan Turing often comes to mind today for reasons other than computing. He was "gay" and, unlike his contemporaries, insisted on being open about it when homosexuality was still illegal in England. It got him into trouble with the courts, which in turn may have been a cause of his suicide when he was just reaching the peak of his career. How much more might his genius have achieved, had he lived in our more tolerant age? How much more can ANY person achieve in an environment that accepts harmless deviations from arbitrary rules?

Author's note: This essay was written on June 23, 1992, on the 80-th anniversary of Turing's birthday, as a small "thank you" for our wonderful "Turing machines".



Soldering on to a perfect finish

The ability to use a soldering iron is of major importance in any electronic projects. David Elbee gives a simple guide on how to make it seem easy.

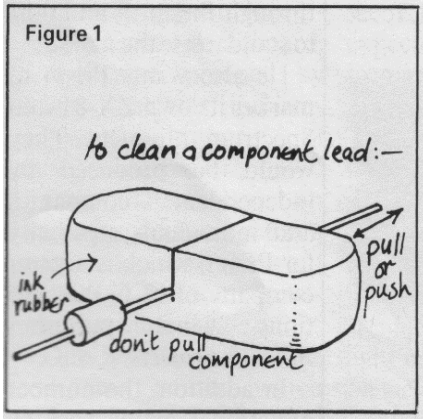
SOLDERING is easy, isn't it? Well, it is once you can do it but at first it seems almost impossible. The best way to learn to solder is to watch somebody who does it well and then to try it yourself while they see where you are going wrong.

Nevertheless, I will try to tell you how to make good soldered joints without 'frying' your components, resistors, transistors—and fingers.

Of the things you need, the most important tool obviously is the soldering iron. For working on printed circuit boards, Veroboard and the kind of circuitry we are concerned about. I suggest you use a small lightweight iron of no more than 15 watts power. I use both an Autex C-240 with 1/32 in., 3/32 in. and 5/32 in. bits; and a low-voltage Oryx 6V 6-watt miniature soldering iron for really small jobs. Other manufacturers produce equally suitable small lightweight irons. I also suggest you buy a soldering iron stand as well.

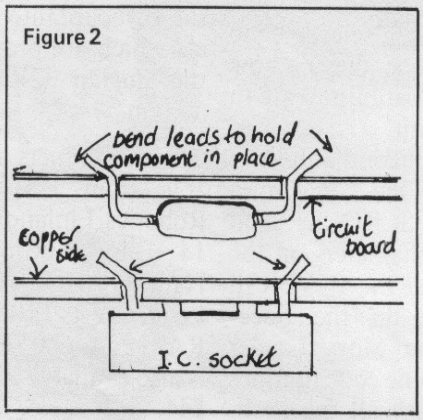
You many wonder why I use three

Figure 1



different sizes of bit. The answer is that the bit is a store of heat and the bigger the bit the more heat it stores, so if I am making very small joints I use the 1/32in. bit and for larger ones on switches or terminals I use the 5/32in. one. If I use the 1/32in. one on

Figure 2



large joints all the heat flows out of the bit into the joint and there is not sufficient to heat the joint and bit to a high enough temperature for the solder to flow. I know the soldering iron bit is being heated by the electric element in the iron but that is being done only slowly and we need fast heat transfer to the joint, or we can over-heat the components being soldered. On the other hand, if I am making very small joints the 5/32in. bit is often too big to reach the joint without touching something I do not want it to touch.

For soldering electronic components we need to use a good quality resin-cored solder made specially for electronics work. The resin core, when heated by the soldering iron, melts into a flux which helps heat transfer to the joint and mildly cleans and protects the components from the air while they are being soldered, and because when the joint has cooled the flux has solidified in all the nooks and crannies it is vital that it does not contain harmful chemicals which must be washed away.

The cored solder which you can buy from a local ironmonger may or may not be satisfactory but I wouldn't use it. I use Ersine Multicore five-core solder in a 0.71mm. size but any cored solder you buy from a reputable component supply shop or radio/TV repair shop should be suitable, provided it is a small diameter size.

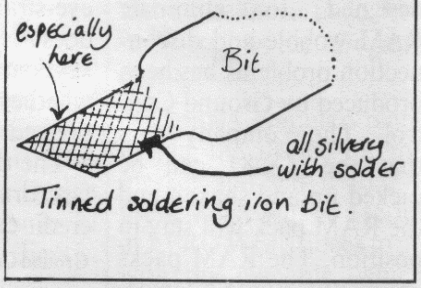
A damp paper towel is needed to clean the hot bit before starting to solder and to clean the flux from the bit after each joint before you return the iron to its stand. Soldering iron stands usually have a piece of sponge but I prefer the damp towel.

I never use heat shunts; most components will tolerate their leads being at the temperature of molten solder for five to 10 seconds and that is ample time to make a soldered joint.

Some people would say you need a small pair of round-nosed pliers or tweezers for bending component leads. They are useful to have but most times I just bend the leads with my fingers.

An ink rubber is useful. If you

Figure 3



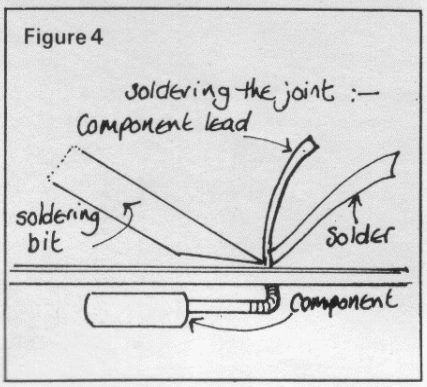
make a slit in one and draw a component lead through it—figure one—it will clean-off all the dirt, grime and grease from the lead. Normally that is unnecessary but sometimes a batch of resistors can be difficult to solder. You can also use it to rub over the copper strips on the circuit board to make them shine; that makes them easier to solder and will not damage them.

Blu-tack is very usful for holding components in place in a circuit board until they are soldered and saves burnt fingers.

Now we have discussed what you need, we can proceed to the process.

To start, though, I would have a few integrated circuit sockets and a small piece of Veroboard and practise soldering the sockets in place. That way it does not matter if you make a mess of the first few joints,

Figure 4

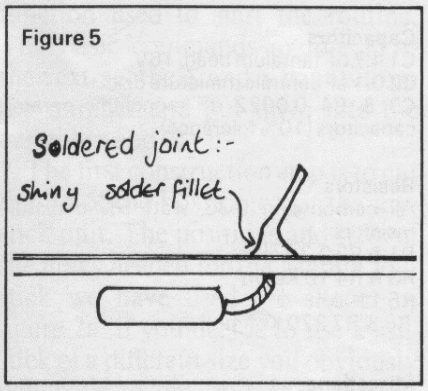


because at the end you can just throw away the assembly and put the small cost down to experience.

Before you begin, prepare a space to work on. Do not work on a polished table; components leave scratches and hot solder makes little hard burnt patches. Protect the surface with two newspapers or a sheet of hardboard or cardboard.

Switch on the iron and leave it for five minutes to warm-up and stabilise at the proper temperature. Put a component in the circuit board and bend the leads a little where they protrude the other side, to hold it in place—figure two. Blu-tack also

Figure 5



helps but do not use it on the copper side of the board—it leaves something behind which makes soldering difficult. Make sure the circuit board cannot move when you touch it with the iron—stick it down with Blu-tack.

Then wipe the soldering iron bit on the damp towel to clean it; it should be a silvery colour all over the part which will be in contact with the joint—figure three.

Wait for at least 10 seconds for the tip of the bit to return to temperature and touch the end of the bit with the end of the solder to put some molten flux on the bit. Then touch the joint with the bit and immediately afterwards touch the joint and the end of the bit with solder and feed in the solder as it melts—figure four.

Almost immediately the flux should flow over the joint, to be followed quickly by the molten solder which should form a fillet at the joint—figure five. As soon as the molten solder has formed a fillet, stop feeding-in the solder and take away the iron. From the time you touched the bit to the joint to the time you took it away should be no more than about four seconds. Then wipe the bit on the damp towel to clean off the flux and excess solder.

Do not disturb the component until the joint has set. That will take from two to four seconds. If you disturb it while it is cooling you will break the solder connection in the joint and have what is known as a dry joint, which does not conduct electricity very well. If you make a dry joint re-melt the solder at the joint and let it set again.

Then cut off the component lead about 1/16in. from the circuit board. Do not bother to cut off the pins of integrated circuit sockets, even though they may be fractionally longer—figure six.

Inspect the joint carefully; it should look like figure five and there should be no solder splashed around the joint or solder bridges to other components.

When you have finished soldering-in all the components inspect the board carefully for joints you have missed, solder bridges, solder splashes which may short-out something. If there is a quantity of flux on the board, and to ease inspection, you can clean-off the flux with methylated spirit and an old toothbrush but be careful—the spirit may dissolve some components. Now to look at what can go wrong:

—The bit may be too small for the joint and the joint does not heat up in time. In that event, try a bigger bit.

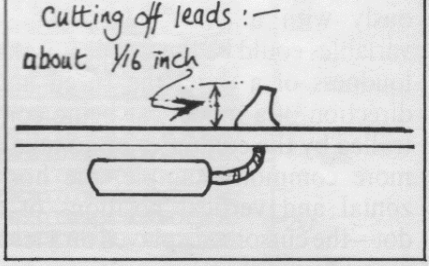
—The iron has not been switched on long enough and has not reached temperature. Wait a short time.

—The bit is dirty and/or it is not shiny with solder—a bit which is shiny with solder is technically called "tinned". Clean it on the damp towel and apply some solder.

—The component is dirty. Try the ink rubber on its leads. As a last resort try scraping the leads gently with a knife.

Some old plastic-covered wire gives off some chemical which seems to prevent the solder flowing over the

Figure 6



joint. That is generally green plastic for some reason. Do not use it.

Get the joint too hot by keeping the bit on it too long and the component will burn and/or the copper will come off the circuit board. If that is happening, you need more practice.

Solder bridges to other components caused by too much solder or too big a bit can be carefully removed by stroking with the bit.

Solder splashes happen to everybody sometimes. They should brush off; if not, try stroking gently with the bit.

Let me repeat, soldering is not difficult providing you do it the correct way. What you are trying to do is to get the molten solder to wet the surface of the things to be joined so as to form a good electrical contact. Consequently things must be clean and not moved while they are setting.

That said, soldering really is easy. Good luck.

ASSEMBLY LANGUAGE MADE EASY? PART 4

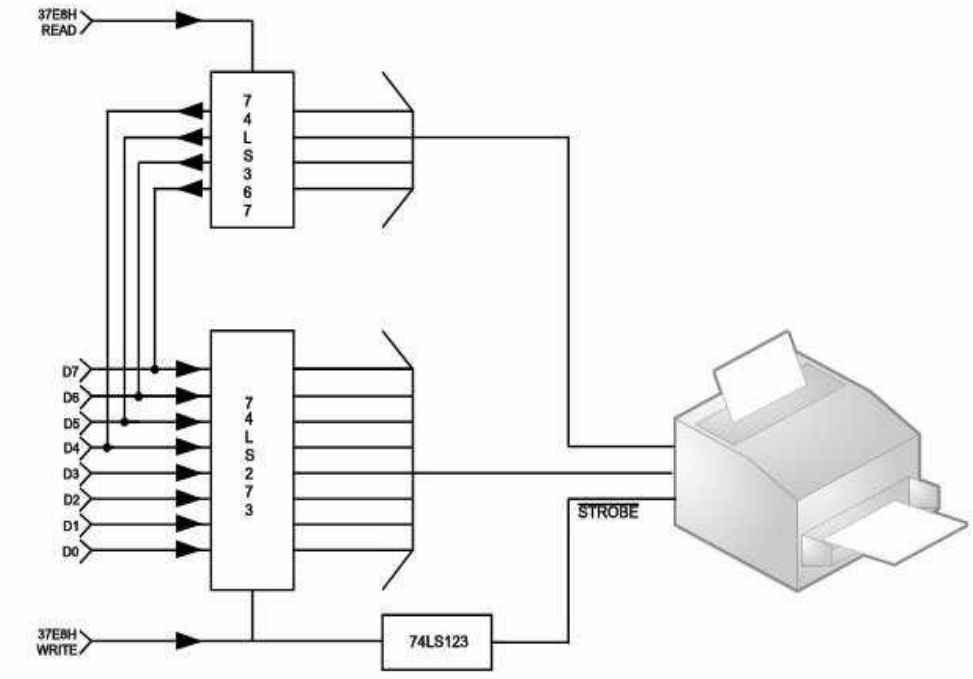
PETER STONE

WELCOME TO THE FOURTH INSTALMENT IN MY SERIES AIMING TO REFRESH YOUR ASSEMBLY PROGRAMMING SKILLS. I'M TAKING A BREAK FROM PROGRAMMING FOR THE VIDEO DISPLAY & MOVE ONTO SOME HARDWARE PROGRAMMING FOR THE PARALLEL & SERIAL PORTS. ONE THING I SHOULD MENTION BEFORE I START, & THAT IS TO GET THE MOST OUT OF THE SERIAL PART OF THE ARTICLES, YOU WILL NEED A SECOND COMPUTER, A COMMUNICATIONS PACKAGE FOR THAT COMPUTER, & A NULL MODEM LEAD WHICH YOU CAN EITHER CONSTRUCT OR BUY. YOU MAY ALSO WISH TO HAVE A COPY OF LAST ISSUES' ARTICLE OPEN AS YOU READ THIS ONE FOR EASY REFERENCE.

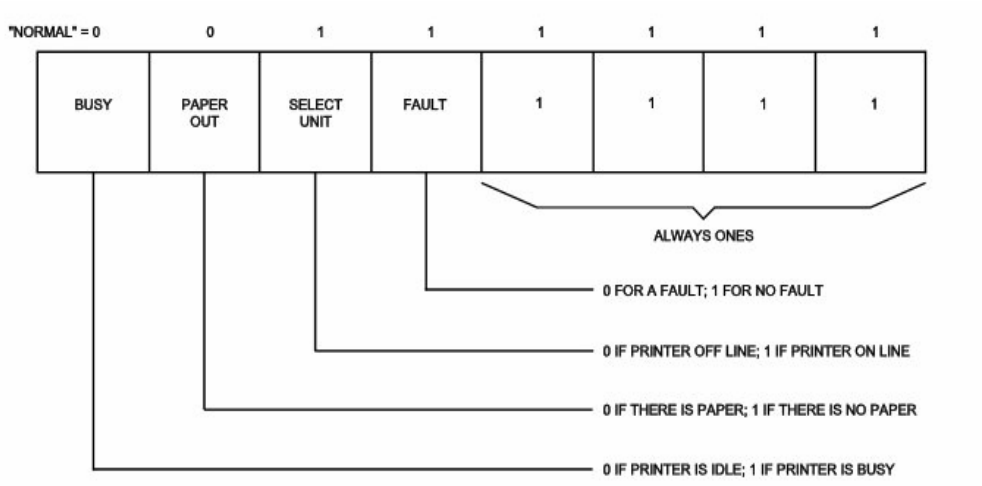
THE PARALLEL PORT

IF YOU REFER TO LAST ISSUES' ARTICLE, YOU WILL SEE THAT THE PRINTER PORT IS MAPPED TO MEMORY LOCATION 37E8H. THIS PORT SERVES TWO FUNCTIONS. WRITING A BYTE TO THIS ADDRESS WILL CAUSE THE ASCII CHARACTER ASSOCIATED WITH THE BYTE TO BE PRINTED. DOING A READ ON THIS ADDRESS WILL GET THE STATUS OF THE PRINTER.

A SIMPLIFIED SCHEMATIC DIAGRAM OF THE PRINTER PORT CIRCUITRY IS SHOWN BELOW.



EXECUTING A READ COMMAND ON 37E8H CAUSES THE PRINTER STATUS TO BE READ. HOW THIS BYTE IS MADE UP IS SHOWN BELOW.



A SIMPLE MACHINE CODE SUBROUTINE TO CHECK THIS BYTE WOULD LOOK LIKE THIS.

```
LPSTAT  PUSH    AF          ;SAVE REGISTER PAIR
LPSØ1   LD      A,(37E8H)   ;GET PRINTER STATUS
        AND     ØFØH        ;MASK UNUSED BITS
        CP      3ØH         ;TEST FOR BUSY
        JP      NZ,LPSØ1    ;LOOP IF BUSY
        POP     AF          ;RESTORE REGISTER PAIR
        RET
```

THE PROGRAM FLOW FOR THIS SUBROUTINE IS FAIRLY SIMPLE. WE GET THE PRINTER STATUS BY A LD A,(37E8H), THE STATUS BITS ARE ANDED WITH ØFØH TO MASK OUT THE UNUSED LOWER BITS. FROM THE DRAWING ABOVE WE CAN SEE THAT THE ONLY BITS THAT ARE NORMALLY AT A ONE ARE BITS 5 & 4, SO THE CP 3ØH TAKES CARE OF THIS. IF EITHER THE PRINTER IS BUSY, OR THERE IS A FAULT, THEN THIS WILL CAUSE THE CP INSTRUCTION TO FAIL & THE PROGRAM WILL LOOP UNTIL EITHER THE PRINTER ISN'T BUSY OR THE FAULT IS RECTIFIED & THE SUBROUTINE WILL END.

WHILE THE ABOVE SUBROUTINE WILL DO A SIMPLE CHECK, IT WOULD BE MORE ADVANTAGEOUS FROM A USER POINT OF VIEW IF THE PRINTER ACTUALLY TOLD YOU WHAT THE ERROR IS. A PRINTER SUBROUTINE TO DO THIS & ALSO PRINT THE CHARACTER WOULD LOOK SOMETHING LIKE THIS.

```
; LINE PRINTER STATUS SUBROUTINE
;
LPSTAT  PUSH    AF          ;SAVE REGISTER PAIR
LPSØ1   LD  A,(37E8H)       ;GET PRINTER STATUS
        AND ØFØH            ;MASK UNUSED BITS
        CP  8ØH             ;TEST FOR BUSY
        JP  Z,LPSØ1         ;LOOP IF BUSY
LPSØ2   CP  4ØH             ;CHECK FOR PRINTER OUT OF PAPER
        JP  NZ,LPSØ3        ;IF PRINTER HAS PAPER, NEXT CHECK
        CALL PAPERR         ;PRINT 'PRINTER OUT OF PAPER' ERROR
                            ON SCREEN
        JP  LPSØ2           ;LOOP UNTIL PRINTER HAS PAPER
LPSØ3   CP  2ØH             ;CHECK PRINTER IS ONLINE
        JP  Z,LSPØ4         ;IF PRINTER  IS ONLINE, NEXT CHECK
```

```
        CALL    OFFLNE  ;PRINT 'PRINTER IS OFFLINE' ON SCREEN
        JP  LPSØ3       ;LOOP UNTIL PRINTER IS ONLINE
LPSØ4   CP  1ØH         ;CHECK FOR PRINTER FAULT
        JP  Z,FIN1      ;NO PRINTER FAULT, FINISH
        CALL PRFLT      ;PRINT 'THERE IS A PRINTER FAULT' ON
                        SCREEN
        JP  LPSØ4       ;LOOP UNTIL FAULT CONDITION IS
                        CLEARED
FIN1    POP AF          ;RESTORE REGISTER PAIR
        LD  A,C         ;GET CHARACTER
        LD  (37E8H),A   ;SEND IT TO THE PRINTER
        RET             ;DONE
```

IF THE SUBROUTINE IS VIEWED AS A SERIES OF LOOPS, THEN IT IS EASY TO ANALYSE. THE CODE NEEDED TO PRINT THE ERROR MESSAGES ON SCREEN IS, IN THE WORDS OF SOME OF THE TEXTBOOKS I HAVE, 'LEFT AS AN EXERCISE FOR THE STUDENT'. IF YOU STUDY THE ARTICLE I WROTE IN TRS8BITØ3Ø3 THEN THIS SHOULD HELP YOU WITH WRITING THE CODE.

WRITING TO 37E8H RESULTS IN THE BYTE ON THE DATA BUS BEING STORED IN THE 74LS273, & AFTER A 13ØMS DELAY CAUSED BY THE 74LS123, IS THEN PRINTED.

THE SERIAL (RS232) PORT

THE SERIAL PORT ISN'T A MEMORY MAPPED DEVICE. INSTEAD IT USES FOUR I/O (INPUT/OUTPUT) PORTS & THE IN & OUT COMMANDS TO PROGRAM & TRANSMIT & RECEIVE DATA. THERE ARE A POSSIBLE 255 8 BIT I/O PORTS NATIVELY AVAILABLE TO THE Z8Ø PROCESSOR. THIS IS DUE TO THE FACT THAT THE LOWER ADDRESS LINES, (AØ - A7), TO ADDRESS THE DESIRED PORTS. OF THESE, ONE (PORT 255,) IS USED FOR CONTROLLING THE CASSETTE RECORDER/S & VIDEO DISPLAY. THE PORTS USED BY THE RS232 INTERFACE ARE LISTED IN THE TABLE BELOW, ALONG WITH THEIR FUNCTIONS.

AD-DRESS	OUT (I/O PORT WRITE)	IN (I/O PORT READ)
E8H (232)	MASTER RESET (ANY DATA)	MODEM STATUS REGISTER
E9H (233)	BAUD RATE SELECT	CONFIGURATION SENSE SWITCHES
EAH (234)	UART CONTROL REGISTER & HANDSHAKE LATCH	UART STATUS REGISTER
EBH (235)	TRANSMIT DATA REGISTER	RECEIVED DATA REGISTER

PORT E8H

PERFORMING THE Z8Ø INSTRUCTION OUT (ØE8H),A CAUSES THE UART INTO A KNOWN STATE, & SHOULD BE PERFORMED ONCE BEFORE ANY UART PROGRAMMING IS PERFORMED, USUALLY DONE WHEN THE PROGRAM IN INITIALISING. IN THIS INSTANCE, THE ACTUAL DATA HELD IN REGISTER A IS UNIMPORTANT AS IT HAS NO RELATION TO THE RESULT OF THE OPERATION.

PERFORMING THE Z80 INSTRUCTION IN A,(0E8H) WILL LOAD THE CONTENTS OF THE MODEM STATUS REGISTER INTO A. THIS INFORMATION IS NOW RELEVANT TO THE STATUS OF ANY EXTERNAL EQUIPMENT CONNECTED TO THE RS 232 INTERFACE. THE BIT ALLOCATION FOR THE DATA HELD IN A IS AS FOLLOWS

D7	D6	D5	D4	D3	D2	D1	D0
Clear To Send Pin 8 on D9 (CTS)	Data Set Ready Pin 6 on D9 (DSR)	Carrier Detect Pin 1 on D9 (CD or DCD)	Ring Indicator Pin 9 on D9 (RI)	Unused	Unused	Unused	Reciever Input UART P20

THE CTS, DSR, CD & RI INPUTS ALL WORK THE SAME WAY. A POSITIVE VOLTAGE INDICATES THAT THE PARTICULAR SIGNAL IS ACTIVE. FOR EXAMPLE, A POSITIVE VOLTAGE ON THE CARRIER DETECT PIN WOULD INDICATE THAT THERE IS A CARRIER BEING RECEIVED BY THE MODEM. BECAUSE THE RS232 VOLTAGE LEVELS ARE +12 VOLTS TO -12 VOLTS, ANY ATTEMPT TO APPLY THESE VOLTAGES DIRECTLY TO THE COMPUTER CIRCUITRY WOULD RESULT IN A SPECTACULAR CIRCUIT FAILURE. THEREFORE THERE ARE INTEGRATED CIRCUITS USED TO TRANSFORM THE HIGHER VOLTAGES INTO THE RANGE +5 VOLTS (A LOGIC 1) TO 0 VOLTS (A LOGIC 0), WHICH THE TTL LOGIC WITHIN THE COMPUTER CAN HANDLE. THESE INTEGRATED CIRCUITS, BESIDES REDUCING THE VOLTAGE, INVERT THE OUTPUT, SO +12 VOLTS IN WILL RESULT IN 0 VOLTS OUT, & A -12 VOLTS IN WILL RESULT IN +5 VOLTS OUT. FROM THIS, WE CAN SEE THAT WHEN A CARRIER IS BEING RECEIVED THEN D6 WILL CONTAIN A ZERO, & WHEN THERE IS NO CARRIER THEN D6 WILL CONTAIN A 1. D6 IS USED TO PROVIDE A RAW SERIAL DATA STREAM FOR YOUR OWN USE, BYPASSING THE UART RECEIVER REGISTER. SOME EXAMPLE CODE TO CHECK THE STATUS OF THE CARRIER DETECT & ACT ON IT IS SHOWN ON THE NEXT PAGE;

```
MAIN    IN  A,(0E8H)     ;GET MODEM STATUS
        AND 32           ;CHECK CARRIER DETECT (BIT 6 - D5)
        JR  NZ,NOTONE    ;LOST THE CARRIER
```

THE AND INSTRUCTION CHECKS THE MODEM STATUS BIT D5, FOR A FULLER EXPLANATION, SEE MY FIRST ARTICLE IN TRS8BIT0302

PORT E9H

PERFORMING THE Z80 INSTRUCTION OUT (0E9H),A WILL PROGRAM THE BAUD RATE GENERATOR. THE BAUD RATE IS SIMPLY THE SPEED THAT THE UART SENDS & RECEIVES DATA AT. THE AVAILABLE BAUD RATES ARE SHOWN IN THE TABLE BELOW.

NIBBLE LOADED	BAUD RATE	NIBBLE LOADED	BAUD RATE
ØH	5Ø	8H	18ØØ
1H	75	9H	2ØØØ
2H	11Ø	AH	24ØØ
3H	134	BH	36ØØ
4H	15Ø	CH	48ØØ
5H	3ØØ	DH	72ØØ
6H	6ØØ	EH	96ØØ
7H	12ØØ	FH	192ØØ

THE LOW ORDER NIBBLE (DØ - D3), CONTROL THE RECEIVE SPEED, & THE HIGH ORDER NIBBLE (D4 - D7), CONTROL THE TRANSMIT SPEED. SO FOR EXAMPLE IF WE WISHED TO SEND & RECEIVE AT 3ØØ BAUD, THE INSTRUCTIONS WOULD BE:

```
        LD  A,55H       ;LOAD A WITH BYTE FOR 3ØØ BAUD TRANS
                        & RECEIVE
        OUT (ØE9H),A    ;LOAD BRG LATCH WITH VALUE
```

HOWEVER, THE ABILITY TO TRANSMIT & RECEIVE AT SEPARATE BAUD RATES WOULD ALLOW, FOR EXAMPLE, THE TRSØØ TO ACT AS A PRESTEL STYLE SERVER. THIS WOULD REQUIRE THE COMPUTER TO RECEIVE AT 75 BAUD & TRANSMIT AT 12ØØ BAUD. THEREFORE THE INSTRUCTIONS WOULD BE:

```
        LD  A,71H       ;LOAD A WITH BYTE FOR 12ØØ TRANS & 75
                        RECEIVE
        OUT (ØE9H),A    ;LOAD BRG LATCH WITH VALUE
```

THE BAUD RATE COULD BE CHANGED AT ANY POINT DURING COMMUNICATION. FOR EXAMPLE, IF YOU WERE SYNCHRONISING TO DIFFERENT SPEED MODEMS. YOU WOULD START AT THE HIGHEST FREQUENCY & WORK DOWN UNTIL A LOCK WAS ACHIEVED. OF COURSE, IF YOU'RE USING A TRSØØ EMULATOR, THEN YOU ARE TIED INTO THE PC'S HARDWARE BAUD RATES. THIS SUPPORTS A LIMITED RANGE, & DOESN'T ALLOW SPILT WORKING. I WILL DEAL WITH THIS LATER.

PERFORMING THE ZØØ INSTRUCTION, IN A,(ØE9H) READS THE CONFIGURATION SENSE SWITCHES. THESE WERE A BANK OF EIGHT SWITCHES MOUNTED ON A MODEL 1 RS232 INTERFACE BOARD WHICH ALLOWED A DEFAULT SETTING OF BAUD RATE, WORD LENGTH, PARITY & STOP BIT TO BE SETUP. THIS COULD BE READ BY A COMMUNICATIONS PACKAGE & BE USED AS A STARTING POINT FOR ITS COMMUNICATION PARAMETERS. HOWEVER THESE SETTINGS CAN BE OVER-RIDDEN. FOR COMPLETENESS, I INCLUDE THE BIT ALLOCATION FOR THE DATA HELD IN A.

D7 (S1)	D6 (S2)	D5 (S3)	D4 (S4)	D3 (S5)	D2 (S6)	D1 (S7)	D0 (S8)
Parity Select 1=even, 0=odd	Word Length Select 1	Word Length Select 2	Stop Bit Select 1=2 bits, 0=1 bit	Parity Inhibit 1 disbles parity	Baud Rate 2	Baud Rate 3	Baud Rate 1

WHAT EACH OF THE SWITCH COMBINATIONS DO, IS BEST ILLUSTRATED BY THE TABLE BELOW

	S8	S7	S6	S5	S4	S3	S2	S1	NOTES
BAUD RATE									
110	C	C	C						C=CLOSED, O=OPEN
150	O	C	C						C=CLOSED, O=OPEN
300	C	C	O						C=CLOSED, O=OPEN
600	O	C	O						C=CLOSED, O=OPEN
1200	C	O	C						C=CLOSED, O=OPEN
2400	O	O	C						C=CLOSED, O=OPEN
4800	C	O	O						C=CLOSED, O=OPEN
9600	O	O	O						C=CLOSED, O=OPEN
STOP BITS									
ONE STOP BIT				C					C=CLOSED, O=OPEN
TWO STOP BITS				O					C=CLOSED, O=OPEN
PARITY ENABLE									
PARITY ENABLED					C				C=CLOSED, O=OPEN
PARITY DISABLED					O				C=CLOSED, O=OPEN
WORD LENGTH									EXCLUDES PARITY BIT
5 BIT WORD						C	C		C=CLOSED, O=OPEN
6 BIT WORD						C	O		C=CLOSED, O=OPEN
7 BIT WORD						O	C		C=CLOSED, O=OPEN
8 BIT WORD						O	O		C=CLOSED, O=OPEN
PARITY SELECT									
ODD PAR-ITY								C	C=CLOSED, O=OPEN
EVEN PAR-ITY								O	C=CLOSED, O=OPEN

PORT EAH

PERFORMING THE Z80 INSTRUCTION OUT (0EAH),A LOADS THE INFORMATION IN THE A REGISTER INTO THE UART CONTROL REGISTER & HANDSHAKE LATCH. THESE TWO FUNCTIONS ARE SEPARATE & DISTINCT, WITH THE UPPER 5 BITS TELLING THE UART WHAT NUMBER OF DATA BITS, STOP BITS, WORD LENGTH & PARITY CONVENTION. WHILE THE LOWER THREE BITS ARE LATCHED CONTROL OUTPUTS.

D7	D6	D5	D4	D3	D2	D1	D0
Parity Enable 1=even, 0=odd	Word Length Select 1	Word Length Select 2	Stop Bit Select 1=2 bits, 0=1 bit	Parity Inhibit 1 disables parity	Break 0 Disables Transmit Data	Data Terminal Ready DTR Pin 4 on D9	Request to Send RTS Pin 7 on D9

COMPARING THIS DIAGRAM WITH THE ONE ABOVE, WE CAN SEE THAT THE FIRST FIVE BITS MIRROR EACH OTHER, WITH THE VALUES OF WORD LENGTH BEING SELECTED FROM THE TABLE ABOVE. THE LAST THREE BITS PROVIDE CONTROL BETWEEN THE RS232 INTERFACE & THE EXTERNAL EQUIPMENT. SETTING D2 TO 0, WILL CAUSE ALL TRANSMISSION FROM THE UART TO STOP. SETTING D1 TO 0, PROVIDES A SIGNAL TO TELL THE EXTERNAL EQUIPMENT THAT IT SHOULD CONNECT TO THE COMMUNICATIONS CHANNEL & MAINTAIN THE CONNECTION AS LONG AS THE 'ON' CONDITION IS PRESENT, WHILE SETTING D0 TO 1, TELLS THE EXTERNAL EQUIPMENT TO STAY IN TRANSMIT MODE, (FOR DUPLEX COMMUNICATION), WHILE TELLING THE EXTERNAL EQUIPMENT TO TRANSMIT, SETTING IT TO A 1 WILL TELL THE EXTERNAL EQUIPMENT TO RECEIVE, (IN HALF DUPLEX MODE).

THE FOLLOWING EXAMPLE SHOWS HOW THIS MIGHT BE DONE.

```
        IN  A,(0E9H)    ;READ SENSE SWITCHES
        AND 0F8H        ;ZERO LOWER 3 BITS
        OR  05H         ;SET REQUEST TO SEND, & BREAK,
                        ;RESET DATA TERM READY
        OUT (0EAH)      ;LOADS UART CONTROL REGISTER
                        ;& HANDSHAKE LATCH
        LD  (IMAGE),A   ;SAVES BIT PATTERN FOR UPDATES
```

REMEMBER THAT THE LABEL IMAGE POINTS TO A MEMORY LOCATION USED TO SAVE THE CURRENT STATE OF THE UART CONTROL REGISTER & HANDSHAKE LATCH. AS AN EXAMPLE, SUPPOSE THAT YOU WANT TO CHANGE THE LOGIC STATE OF REQUEST-TO-SEND, THE FOLLOWING EXAMPLE SHOWS HOW TO DO THIS WITHOUT CHANGING WITHOUT CHANGING THE UART CONTROL REGISTER OR THE OTHER HANDSHAKE BITS.

```
        LD  A,(IMAGE)   ;LOAD CURRENT STATE OF UART CONTROL
                        REGISTER
        RES 0,A         ;RESETS BIT 0 IN A (REQUEST TO SEND)
        OUT (0EAH)      ;LOADS NEW BIT PATTERN INTO UART CON
                        TROL REGISTER
        LD  (IMAGE),A   ;SAVES NEW PATTERN FOR UPDATES
```

PERFORMING THE Z80 INSTRUCTION, IN A,(0EAH) LOADS A WITH THE CONTENTS OF THE UART STATUS REGISTER. THIS PROVIDES A NUMBER OF FLAGS THAT CAN TELL US IF ANY ERRORS HAVE OCCURRED.

D7	D6	D5	D4	D3	D2	D1	D0
Data Received 1 = true	Transmitter HoldingRegister Empty 1 = true	Overrun Error 1 = true	Framing Error 1 = true	Parity Error 1 = true	Unused	Unused	Unused

D7 TELLS US WHEN SOME DATA HAS BEEN RECEIVED. D6 TELLS US WHEN WE CAN LOAD ANOTHER CHARACTER TO THE HOLDING REGISTER, READY TO BE SENT. D5, D4 & D3 TELL US IF ANY ERRORS HAVE OCCURRED IN THE DATA THAT HAS BEEN RECEIVED.

PORT EBH

PERFORMING THE Z80 INSTRUCTION OUT (0EBH),A LOADS THE BYTE IN THE A REGISTER INTO THE TRANSMIT DATA REGISTER. THIS SHOULD NOT BE DONE UNTIL THE HOLDING REGISTER IS EMPTY (AS SHOWN ABOVE). THE FOLLOWING EXAMPLE SHOWS HOW THIS IS DONE.

```
;OUTPUT A CHARACTER TO UART FOR TRANSMISSION
;A SHOULD CONTAIN THE CHARACTER TO BE TRANSMITTED

        PUSH    AF      ;SAVE CHARACTER TO BE TRANSMITTED
STATIN  IN  A,(0EAH)    ;LOAD UART STATUS REGISTER
        BIT 6,A         ;TEST TRANSMITTER HOLDING REGISTER
                        FOR A HIGH
        JR  NZ,STATIN   ;TRY AGAIN IF NOT
        POP AF          ;RESTORE CHARACTER TO BE TRANSMITTED
        OUT (0EBH),A    ;LOAD HOLDING REGSITER WITH CHARACTER
```

PERFORMING THE Z80 INSTRUCTION, IN A,(0EBH) WILL GET DATA FROM THE RECEIVED DATA REGISTER & STORE IT IN THE A REGISTER. THIS SHOULD NOT BE DONE UNTIL THE RECEIVED DATA BIT IN THE UART STATUS REGISTER IS SET, INDICATING THAT A COMPLETE CHARACTER HAS BEEN RECEIVED. THE FOLLOWING EXAMPLE SHOWS HOW THIS MIGHT BE DONE.

```
;INPUT A CHARACTER FROM THE UART
;CHARACTER RECEIVED WILL BE IN A

 INCHAR IN  A,(0EAH)    ;LOAD UART STATUS REGISTER
        BIT 7,A         ;TEST DATA RECEIVED FOR A 1 (HIGH)
        JR  Z,INCHAR    ;TRY AGAIN IF NOT
        IN  A,(0EBH)    ;LOAD RECEIVED CHARACTER
```

TRS80 MODEL 1 EMULATOR

ABOVE INFORMATION WILL BE USEFUL WHETHER YOU USE AN ACTUAL TRS80 MODEL 1, OR LIKE ME, RUN AN EMULATOR ON A PC. I DO

HAVE AN ACTUAL TRS80 MODEL1, BUT I KEEP THIS FOR SPECIAL OCCASIONS, & THE EMULATOR I USE IS MATTHEW REED'S. IT WILL BE OBVIOUS THAT WHEN USING AN EMULATOR, TRYING TO READ THE SENSE SWITCHES WON'T WORK. THE RESULT FROM INP (233) IN BASIC, FOR ME, IS 0.

NULL MODEMS

AT THE START OF THIS ARTICLE, I SAID THAT YOU NEEDED A SECOND PC TO GET THE BEST OUT OF THIS & THE FOLLOWING ARTICLES. THIS MACHINE CAN BE AS SIMPLE AS AN MSDOS MACHINE, SO LOANS AS YOU HAVE A SUITABLE COMMS PACKAGE, & A SUITABLE LEAD YOU WILL BE OK.

THE SERIAL CONNECTION ON THE BACK OF A PC THESE DAYS, IS A 9 WAY D TYPE PLUG. IF YOU HAVE A VERY UP TO DATE COMPUTER WITH ONLY USB PORTS, THEN YOU NEED TO PURCHASE A USB TO SERIAL ADAPTOR FOR EACH MACHINE. WHATEVER TYPE OF PC YOU HAVE, YOU NEED TO CONNECT THE SERIAL CONNECTORS TOGETHER. TO DO THIS YOU USE A NULL MODEM CABLE. THIS CAN EITHER BE BOUGHT, OR IF YOU HAVE REASONABLE SOLDERING SKILLS, MADE.

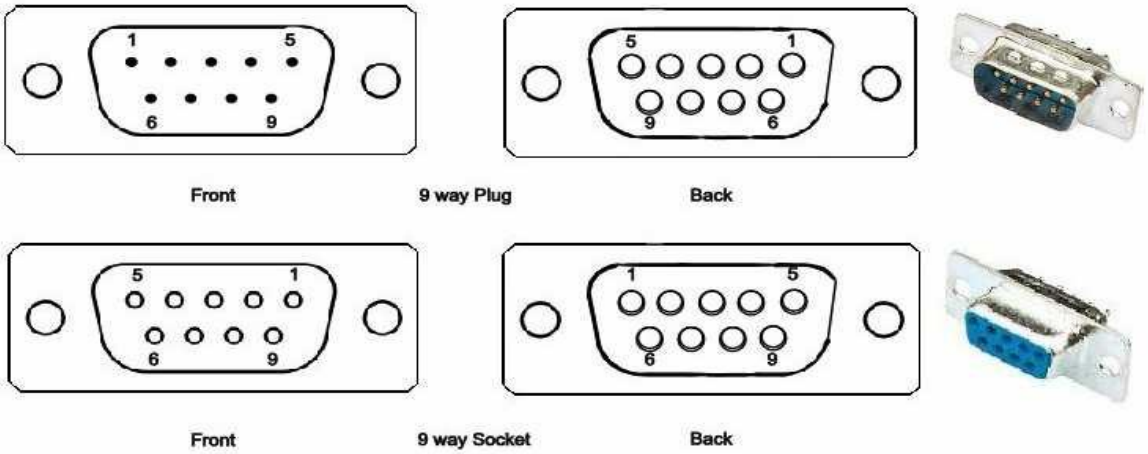
TO MAKE A NULL MODEM CABLE, YOU NEED THE FOLLOWING PARTS.

2 OFF 9 WAY D TYPE PLUGS.
2 OFF 9 WAY HOODS (PLASTIC ARE OK).
A SUITABLE LENGTH OF CABLE, (ALWAYS GET MORE THAN YOU NEED)

THE CABLE IDEALLY NEEDS TO HAVE 8 CORES, EVEN CAT5 CABLE CAN BE USED. IF YOU VISIT WWW.MAPLIN.CO.UK & DO A SEARCH FOR CATALOGUE NUMBER XR27E WILL GIVE YOU AN IDEA OF WHAT TO ASK FOR. I HAVE USED 6 WAY TELEPHONE EXTENSION CABLE (XS04E), FOR LEADS WITH SUCCESS.

GIVING A TUTORIAL ON SOLDERING IS BEYOND THE SCOPE OF THIS ARTICLE. (** SEE PAGES 10 & 11 ED. **) THE ONLY EXTRA ITEMS I'D SUGGEST YOU GET ARE 2 OFF 9 WAY D TYPE SOCKETS, AS THEY WILL BE USEFUL AFTER THE CABLE HAS BEEN MADE TO TEST THAT IT IS WIRED CORRECTLY. SIMPLY PLUG EACH END OF THE MADE UP CABLE INTO A SOCKET, & YOU WILL BE ABLE TO TEST THE WIRING USING A MULTIMETER, OR CIRCUIT TESTER.

THE NUMBERING CONVENTION OF THE 9 WAY CONNECTORS IS SHOWN BELOW.





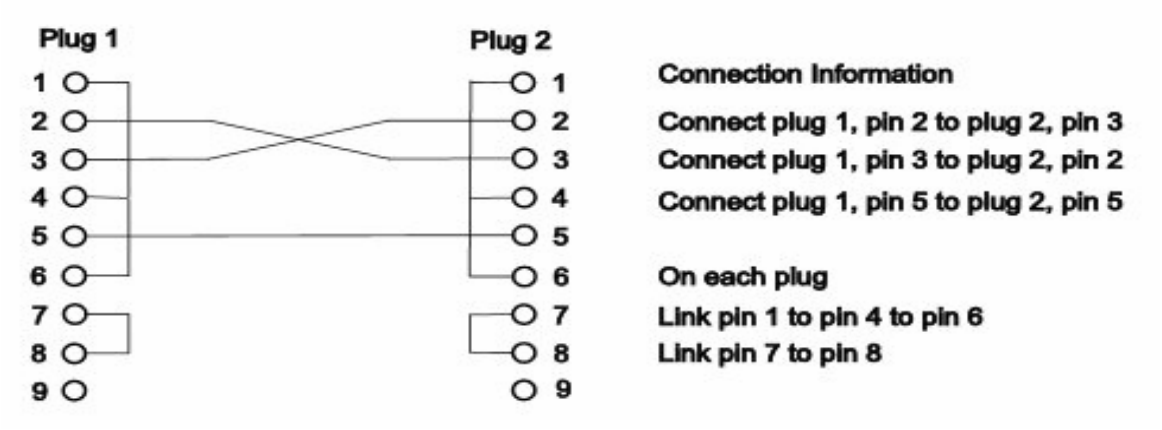
9 way Hood

THE FUNCTIONS OF EACH OF THE PINS ON THE 9 WAY PLUG ARE SHOWN IN THE TABLE BELOW.

D9 PIN NUMBER	ABBREVIATION	FULL NAME
1	CD	CARRIER DETECT
2	RD	RECEIVED DATA
3	TD	TRANSMITTED DATA
4	DTR	DATA TERMINAL READY
5	SG	SIGNAL GROUND
6	DSR	DATA SET READY
7	RTS	REQUEST TO SEND
8	CTS	CLEAR TO SEND
9	RI	RING INDICATOR

THERE ARE A NUMBER OF WAYS YOU CAN MAKE A NULL MODEM LEAD. THE WIRING DIAGRAMS ARE SHOWN NEXT.

THE FIRST IS KNOWN AS A 3 WIRE NULL MODEM, (FOR OBVIOUS REASONS), & IS THE SIMPLEST.

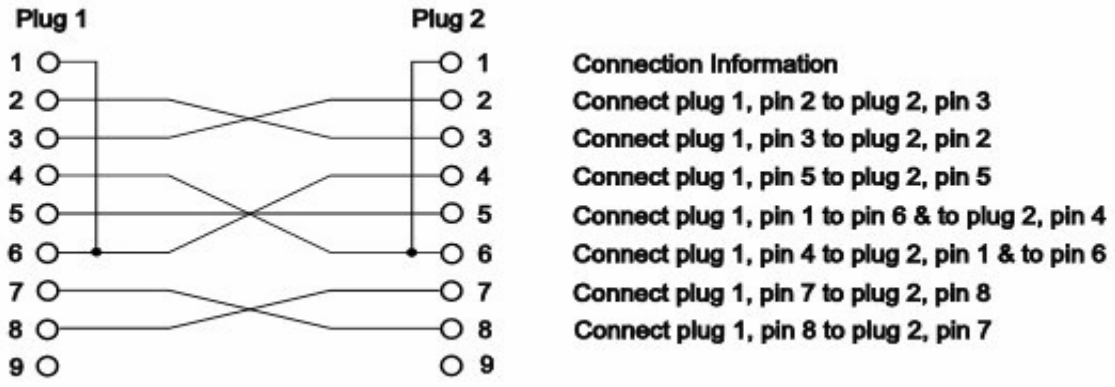


THE THEORY OF OPERATION IS REASONABLY EASY. THE AIM IS TO MAKE TO COMPUTER THINK IT IS TALKING TO A MODEM RATHER THAN ANOTHER COMPUTER. ANY DATA TRANSMITTED FROM THE FIRST COMPUTER MUST BE RECEIVED BY THE SECOND THUS TD IS CONNECTED TO RD. THE SECOND COMPUTER MUST HAVE THE SAME SET-UP THUS RD IS CONNECTED TO TD. SIGNAL GROUND (SG) MUST ALSO BE CONNECTED SO BOTH GROUNDS ARE COMMON TO EACH COMPUTER.

THE DATA TERMINAL READY IS LOOPED BACK TO DATA SET READY AND CARRIER DETECT ON BOTH COMPUTERS. WHEN THE DATA TERMINAL READY IS ASSERTED ACTIVE, THEN THE DATA SET READY AND CARRIER DETECT IMMEDIATELY BECOME ACTIVE. AT THIS POINT THE COMPUTER THINKS THE VIRTUAL MODEM TO WHICH IT IS CONNECTED IS READY AND HAS DETECTED THE CARRIER OF THE OTHER MODEM. ALL THAT IS LEFT TO WORRY ABOUT NOW IS THE REQUEST TO SEND AND CLEAR TO SEND. AS BOTH COMPUTERS COMMUNICATE TOGETHER AT THE SAME SPEED, FLOW CONTROL IS NOT NEEDED THUS THESE TWO LINES ARE ALSO LINKED TOGETHER ON EACH COMPUTER. WHEN THE COMPUTER WISHES TO SEND DATA, IT ASSERTS THE REQUEST TO SEND HIGH AND AS IT'S HOOKED TOGETHER WITH THE CLEAR TO SEND, IT IMMEDIATELY GETS A REPLY THAT IT IS OK TO SEND AND DOES SO.

NOTICE THAT THE RING INDICATOR IS NOT CONNECTED TO ANYTHING OF EACH END. THIS LINE IS ONLY USED TO TELL THE COMPUTER THAT THERE IS A RINGING SIGNAL ON THE PHONE LINE. AS WE DON'T HAVE A MODEM CONNECTED TO THE PHONE LINE THIS IS LEFT DISCONNECTED.

THE NEXT IS A MORE COMPLEX CONNECTOR.



LIKE THE PREVIOUS CONNECTOR, ANY DATA TRANSMITTED FROM THE FIRST COMPUTER MUST BE RECEIVED BY THE SECOND THUS TD IS CONNECTED TO RD. THE SECOND COMPUTER MUST HAVE THE SAME

SET-UP THUS RD IS CONNECTED TO TD. SIGNAL GROUND (SG) MUST ALSO BE CONNECTED SO BOTH GROUNDS ARE COMMON TO EACH COMPUTER.

THE DATA SET READY IS LOOPED BACK TO CARRIER DETECT ON BOTH COMPUTERS, BUT IS THEN CONNECTED TO DATA TERMINAL READY ON THE OTHER COMPUTER. AS BEFORE, WHEN THE DATA TERMINAL READY IS ASSERTED ACTIVE, THEN THE DATA SET READY AND CARRIER DETECT BECOME ACTIVE ON THE OTHER COMPUTER. AT THIS POINT THE COMPUTER THINKS THE VIRTUAL MODEM TO WHICH IT IS CONNECTED IS READY AND HAS DETECTED THE CARRIER OF THE OTHER MODEM. ALL LEFT TO WORRY ABOUT NOW IS THE REQUEST TO SEND AND CLEAR TO SEND. THE REQUEST TO SEND ON ONE COMPUTER IS LINKED TO THE CLEAR TO SEND ON THE OTHER, THIS ALLOWS FLOW CONTROL TO BE USED.

NOTICE THAT THE RING INDICATOR IS NOT CONNECTED TO ANYTHING OF EACH END. THIS LINE IS ONLY USED TO TELL THE COMPUTER THAT THERE IS A RINGING SIGNAL ON THE PHONE LINE. AS WE DON'T HAVE A MODEM CONNECTED TO THE PHONE LINE THIS IS LEFT DISCONNECTED.

THIS MORE COMPLEX CONNECTOR COULD BE USED TO TEST SOFTWARE UNDER DEVELOPMENT, BY USING SOME SORT OF EMULATION PROGRAM ON THE SECOND COMPUTER. TO BE TRUTHFUL, THE FIRST CONNECTOR IS THE MOST COMMON ONE, & ONE I USUALLY USE.

THE NEXT ARTICLE WILL BUILD ON THIS ONE & INVOLVE THE DEVELOPMENT OF A TERMINAL PROGRAM.

FINALLY, IF ANYONE HAS ANY TOPICS THEY WANT ME TO COVER, OR NEED ME TO CLARIFY ANYTHING IN THIS ARTICLE, THEN PLEASE LET ME KNOW BY EMAILING ME AT PETER_STONE45@HOTMAIL.COM.

FOR THE RECORD, ALL THE DIAGRAMS IN THIS & MY PREVIOUS ARTICLES WERE PREPARED USING SERIF DRAWPLUS 8

--== o0o ==--

FROM PAGE 2 ...

I CAME ACROSS AN ARTICLE (SORRY, BUT I CAN'T REMEMBER WHICH MAGAZINE IT'S FROM), GIVING ADVICE AND HANDY TIPS ON SOLDERING. SOMETHING WHICH I THINK MAY HELP WITH PETER'S ARTICLE. THIS CAME ABOUT BY SHEER CHANCE AND WAS SCHEDULED FOR INCLUSION IN THIS ISSUED WEEKS AGO! SORRY IT'S A FUNNY COLOUR, BUT IT'S REPRODUCED FROM THAT OLD FASHIONED COPIER PAPER!

WELL, I THINK THAT ABOUT WRAPS UP THIS EDITION. I HOPE YOU WILL FIND IT OF INTEREST AND PERHAPS ENCOURAGE YOU TO CONTRIBUTE AN ARTICLE. ANYWAY, I'M ALWAYS PLEASED TO HEAR FROM YOU, ANYTIME.

BYE FOR NOW

DUSTY



TRS8BIT



'FLAMING' JUNE AND SUMMER ARE WITH US AT LAST, SO I WELCOME YOU ALL TO OUR 14TH EDITION OF TRS8BIT. I HOPE I'VE MANAGED TO FIND, YET AGAIN, A FEW BITS OF INTEREST FOR YOU ALL TO PLAY WITH, DURING THE SUMMER MONTHS!

AS AN EXPERIMENT, I'VE PLACED A COPY OF TRS8BIT FOR SALE, ON EBAY. (CLEARLY STATING THAT IT'S AVAILABLE AS A FREE DOWNLOAD FROM THE WEBSITE). IT IS A 'SNEAKY' ATTEMPT AT SOME ADVERTISING FOR US. DOWNLOADS HAVE INCREASED SLIGHTLY, SO I'LL TRY AGAIN WHEN THIS ISSUE HITS THE STREETS.

OVER THE PAST FEW MONTHS, EBAY (UK) HAS BEEN DOMINATED BY CO-CO'S FOR SALE. WITH SOME OF THE PRICES OF 'BUY IT NOW'S' STARTING AT £90. I DON'T THINK MANY HAVE SOLD AS THEY KEEP RE-APPEARING!



THERE HAS BEEN VERY LITTLE SELLING ACTIVITY ON THE MODEL 1 FRONT. I'VE ONLY NOTICED A COUPLE RECENTLY. ONE E.I. WITH 32K RAM WAS OFFERED, WHICH DID SELL FOR £28.

HOWEVER, I WAS LUCKY ENOUGH, TO 'WIN' (FOR THE PRINCELY SUM OF 99P) A COPY OF THE Z80 INSTRUCTION HANDBOOK BY NAT WADSWORTH.
THIS, IN TURN, SET MY MIND WANDERING AND I RECALLED AN ARTICLE IN NATGUG NEWS BY PETER KNAGGS ABOUT ILLEGAL Z80 INSTRUCTIONS. I'VE MANAGED TO FIND THE ARTICLE AND HAVE INCLUDED A REPRINT OF IT IN THIS ISSUE.

A REALLY NICE LOOKING MODEL 2 WITH A FAIR AMOUNT OF SOFTWARE, REACHED THE AMAZING SUM OF £500! NOW THAT'S WHAT I CALL REAL MONEY! (I WONDER IF IT'S TAX DEDUCTIBLE).

A WORKING MODEL 102 FETCHED £26. QUITE A BARGAIN I THOUGHT, ESPECIALLY WITH ALL THE NEW HARDWARE THAT'S CURRENTLY AVAILABLE. I CAN QUITE SEE THAT THE M100'S ARE ABOUT TO MAKE A NOTICEABLE COME-BACK. (NOT THAT THEY EVER WENT AWAY ALL THAT MUCH). IF YOU DON'T BELIEVE ME, CHECK OUT THE LINK ON OUR WEBSITE TO THE MODEL 100 CLUB, OR PAGE 12 IF THIS ISSUE!

I ALSO MANAGED TO 'WIN', ON EBAY, ONE OR TWO OTHER GOODIES. A COUPLE OF

REPRINTS OF TANDY'S MICROCOMPUTER NEWS FROM 1981.
THERE WERE LOTS OF INTERESTING BIT AND BOBS, BUT, AS USUAL, I WAS OUT-BID! SURPRISINGLY, ALL THE ITEMS WERE ONLY 20 MILES OR SO UP THE ROAD!
A COPY OF COMPUTRONICS FROM 1983, ALSO CAME MY WAY (SEE PAGE 12). THEY MAKE FASCINATING READING FROM ALL THOSE YEARS AGO.

DUE TO (YET ANOTHER) VERY GENEROUS DONATION TO THE SITE, I'VE ACQUIRED A SECOND ACULAB FLOPPY-TAPE. AS DEE AND I WERE STAYING WITH FRIENDS NEAR TO MILTON KEYNES AT THE TIME, I TOOK THE LIBERTY OF VISITING THE INVENTOR OF THE FLOPPY TAPE, ALAN POUND, (AT ACULAB U.K. PLC H.Q.) AND DONATED THE FLOPPY TAPE TO HIM. HE SEEMED QUITE PLEASED ABOUT IT AND HOPED TO PUT IT ON DISPLAY.
I REMEMBERED HIM QUITE CLEARLY FROM THE 1980'S AND WE HAD A REALLY ENJOYABLE NATTER. ALAN STATED THAT HE WAS QUITE HAPPY FOR ALL INFORMATION RELATING TO THE FLOPPY TAPE TO BE PLACED IN THE PUBLIC DOMAIN.

MANY OF YOU KNOW I'M A BIG FAN OF ELECTRIC PENCIL, (@PENCIL AND 16K RULES!). WHILST BROWSING THROUGH AN ISSUE OF TRSTIMES, I CAME ACROSS A REVIEW OF 'ZORLOF THE MAGNIFICENT' WORD PROCESSOR. I'VE HAD A COUPLE OF 'PLAYS' WITH IT AND I'M QUITE IMPRESSED, SO I THOUGH THE REVIEW WAS WORTH A REPRISE. I ALSO FOUND AN ABRIDGED VERSION

(CONTINUED ON PAGE 20)

AT THE
READY>
PROMPT



WHEN EDITING IN BASIC DON'T FORGET THE "." ABBREVIATION MENTIONED ON PAGE 1/7 OF THE LEVEL 2 MANUAL. IF YOU'VE JUST EDITED LINE 10000, INSTEAD OF TYPING EDIT 10000, EDIT. WILL DO THE TRICK. IN ND80 TYPE "." AND PRESS ENTER, THIS WILL TELL YOU YOUR CURRENT LINE NUMBER. ANYTHING FOR AN EASY LIFE EH!

HERE'S A LITTLE GEM WHICH I CAME ACROSS THE OTHER WEEK. I HAD COMPLETELY FORGOTTEN IT. THE HIGHEST LINE NUMBERS YOU CAN USE DEPENDS ON THE AMOUNT OF RAM YOU HAVE AVAILABLE.
ON A:-
 4K MACHINE IT'S 20479
16K MACHINE IT'S 32767
32K MACHINE IT'S 49151 AND
48K MACHINE IT'S 65535

YOU'LL GET A SYNTAX ERROR IF YOU EXCEED ANY OF THE ABOVE.

FOR LEVEL 2 CASSETTE USERS OF ELECTRIC PENCIL, SHOULD YOUR SYSTEM 'FREEZE UP' ON YOU, ALL THAT IS NEEDED AFTER RESETTING BACK TO READY> IS TO TYPE SYSTEM AND PRESS ENTER. THEN TYPE /17318 AND PRESS ENTER. THIS TECHNIQUE APPLIES TO THE ACULAB FLOPPY PENCIL WITH THE DIFFERENCE BEING THAT THIS VERSION OF PENCIL IS 2048 BYTES HIGHER IN MEMORY, SO AFTER TYPING SYSTEM AND ENTER, TYPE IN /19366

--== o0o ==--

Zorlof Word Processing System

by Fred Blechman

ZORLOF The Magnificent Word Processing System
Anitek Software Products
P.O. Box 361136
Melbourne, FL 32936
(407)259-9397
(See UPDATE at end. IMPORTANT!!!)

"Zorlof The Magnificent Word Processing System" is the unlikely name for an incredibly versatile, easy to use, inexpensive word processor. It seems to have most of the features of much more expensive and complex word processors, yet few of the disadvantages. Although, as with all sophisticated microcomputer programs, it will take some time and effort to learn to use all the features, you can have Zorlof up and running in a short time (AFTER you read through the manual) doing the most common things. From then on you add more and more features as you need them. I suppose there are some esoteric functions it won't perform, but I haven't found any yet that I need!

The first section of the manual, "Getting Started", is included on the Zorlof disk as sample text. This allows you to compare the text and formatting commands with the actual printed results - a very nice touch. You actually learn by example and use rather than just reading the manual. You can modify the sample text at will, trying out all kinds of variations, just to see what happens.

Over 50 Printers Supported

Word processing has developed a vocabulary of its own, but bear with me as I try to cover some of Zorlof's features. For one thing, many word processors are blind to the printer you're using, not allowing you to use many of the printer features. Zorlof supports over fifty popular printers - parallel or serial - and most of their features.

For example, on those printers that offer proportional character spacing, underlining, bolding, subscripts or superscripts, by simply specifying your printer designation at the beginning of your text Zorlof will send the proper character codes to your printer. The same applies to condensed and wide lettering, or even Graftrax italics. You can specify alignment of left margin, right margin, both margins or centering of any line or group of lines. You can set tabs or indent text on both the left and right side, and you can specify different margins for odd and even pages! If you've ever produced a manual or book, you can appreciate how nice it is to be able to have the left and right pages formatted so the center margin is generous.

Page numbering is supported, of course, but you can also put titles and page numbers at the top or bottom (or both) of each sheet, and they can be different for odd and even pages. Each header or footer can run up to three lines. You can pause printing between pages when using single sheets, and you can merge a mailing list and data file with the text file to produce personalized documents. You can print all or any part or parts of a text with printer on-off commands in the text. Page lengths and line spacing are specified, as well as the placement of the first line on a page.

Although it may seem very complex to the casual reader, because of the large number of print formatting options, the built-in defaults give you plain-vanilla output without any commands at all! It's only when you wish something different than the normal default that you must specify. That's why it's so easy to get started with Zorlof.

Text Editing Functions

While the formatting capabilities of Zorlof are extensive, the majority of your effort will be in actually generating text on the screen. I'm happy to report that Zorlof is fast, logical and crash-proof. After using some other word processors that lost all text without warning, or dropped letters while word-wrapping to the next line, I was delighted to find neither of these problems with Zorlof. The key combinations used to delete characters, words or lines - or even blocks of text - were logical and simple. Inserting text or moving blocks of text is a breeze, and the cursor controls are rapid and efficient.

Some of the features available during text entry are outstanding. For example, you can get a disk directory - of any disk - at any time. By placing the cursor on any filename you can load that file at the text cursor position, making text merging a pleasure.

I use various letterheads as files, and load one in at the beginning of the text. If I'm writing a personal letter, a magazine article, correspondence regarding my Amway business, or a letter to a software customer, I use one of the four letterheads I've put on the Zorlof disk as separate files, and then go from there. The directory feature also allows me to kill obsolete files on any disk to make room for the current text file.

And, critically important if you have been using another word processor and already have a bunch of disk files in the "other" format, Zorlof loads any ASCII file into text! Boy, this was really important to me, since I had been using the Electric Pencil 2.0 for about a year and had numerous files I occasionally needed to update. They loaded perfectly into Zorlof and required virtually no

change to use with Zorlof. I now save them all as Zorlof files.

Actually, Zorlof can accommodate five different types of files - Zorlof, Apparat EDTASM, Radio Shack EDTASM, BASIC and ASCII. This means you should be able to use files from most other word processors, since most save their files in ASCII. Also, you can edit BASIC programs with word-processing convenience (if the file was saved in ASCII), and even have limited renumbering capability. Furthermore, you can save a Zorlof file in ASCII for use on other word processors! You can even display and edit any type of data or program file in "ZAP" (byte-hexidecimal) format. The possibilities are enormous.

I had serious problems with Electric Pencil keeping up with my hunt-and-peck typing during inserts or at the beginning of word-wrapped lines. I have yet to "outrun" Zorlof, even with bursts of high speed entry, and even in the insert mode, since it is written entirely in Z80 machine language with type-ahead key-stroke buffering.

Justification and word wrapping (moving a partial word to the next line when in won't fit on the current line) occur in real time on the screen. Reverse word-wrapping fills an incomplete line above the current line. At any time you can scroll the screen up or down, or you can go to the top or end screen of text without scrolling. You can dump the contents of the screen to the printer anytime. You can search text anytime for a string of up to 28 characters (including "wild-card" characters), or you can both search and replace a string with another string. Not only can you move blocks of text, but you can also copy blocks, or delete them.

Printer Preview

Perhaps one of the most convenient and frustration-saving features of Zorlof is its print-preview function. Instead of printing out the text only to find out halfway through that you omitted some kind of formatting command, you can preview the text on the screen. This will show headers, footers, page breaks, justification, line spacing and just about everything but type fonts, underlining, subscripting and superscripting (since the display can't show those). If you spot a problem, you can correct the text or printer commands BEFORE taking the time to print it out first.

Zorlof even allows access to special characters and printer codes that you may wish to use. Line width can be set from 5 to 128; if you specify more than the 64 characters the display can show on a single line, then two display lines will be used.

Two lines continuously resident at the top of the screen tell you the filename (which you can save to disk anytime), the line width (I use 62, for compatibility with my Electric Pencil files), the number of words (right now this reads 1250, more than the editor wanted!), number of lines, and the free memory size. (Free memory with a 48K Model III is 25708 before any text is entered). There are also spaces to enter a "search" and "replace" string. Having these at the top of the screen is not at all distracting, as one might expect.

I was surprised that multiple printer copies could not be commanded directly. A print queue allows up to 13 files to be printed one after another, and you could make multiple copies of a single file that way. Also, if you have a hardware spooler that allows multiple copies, you could pre-program any number of copies. Also, the version of Zorlof I have (2.19D) does not allow you to exit Zorlof and return to the text; the only exit is pressing the computer RESET button. Exit/return, I understand, may be provided in a forthcoming release.

The 96-page detailed manual is excellent! It's bound in a three-ring binder that has pockets front and back for the disk and any notes of your own. A cardboard card acts as both an index and command summary. It lists 59 editing functions and 39 printer commands, with summary of use and manual page number for detailed explanation. Furthermore, in preparation for this article, I had occasion to call Anitek to verify some facts. Peter Ray, Zorlof's creator, was extremely helpful, and schedules appointment times for registered customers to call with questions. It appears that Anitex not only has a great product, but they provide the necessary support as well.

If you have the impression that I like Zorlof, you are close. Actually, I love it!

UPDATE 12/11/90: Although the foregoing article was originally written in 1983, I STILL USE ZORLOF for all my writing. I've written almost 500 magazine articles and five books about microcomputers using Zorlof on my TRS-80 Model III or 4!

I'm over here at TRSTimes as I write this on Lance's Model 4P, updating this and writing a special story for the magazine about using TRS-80s with an HP LaserJet II printer.

Unfortunately, ZORLOF is no longer being sold by Anitek. It was superseded years ago with "LeScript" for both the TRS-80 and IBM PC. LeScript has all the features of Zorlof and more - but takes up memory in my Model III and 4 that I don't want to lose - so I've stayed with Zorlof. If you are not committed to a word processor, try LeScript. Call the above phone number for the latest information and prices.

Best Wishes!....Fred Blechman

ILLEGAL Z80 COMMANDS
AND HOW TO OBTAIN THEM

PETER KNAGGS

THIS ARTICLE FIRST APPEARED IN NATGUG NEWS.

THE Z80 HAS A CAPABILITY FOR A TOTAL OF 3756 INSTRUCTIONS, IF ALL THE BIT PATTERNS AVAILABLE ARE USED. HOWEVER, ZILOG ONLY INFORM US OF 691, THESE ARE KNOWN AS 'LEGAL COMMANDS'.
THIS MEANS THAT THERE ARE OVER 600 COMBINATIONS LEFT UN-ASSIGNED. MANY OF THESE COMBINATIONS ACTUALLY DO HAVE FUNCTIONS AND WORK ON MOST LATER VERSIONS OF THE Z80 MICROPROCESSOR.
MANY ASSEMBLERS WILL NOT ACCEPT THE FOLLOWING 'ILLEGAL' COMMANDS
THE SLL, SHIFT LOGICAL LEFT FAMILY. (THESE HOWEVER, ARE AVAILABLE UNDER THE SLS COMMAND WITH LATER VERSIONS OF THE ZEN EDITOR/ ASSEMBLER).

HEX	COMMAND
CB 37	SLL A
CB 30	SLL B
CB 31	SLL C
CB 32	SLL D
CB 33	SLL E
CB 34	SLL H
CB 35	SLL L
CB 36	SLL (HL)

THERE IS ALSO A METHOD OF ACCESSING THE HIGH AND LOW BYTES OF THE INDEX REGISTERS IX AND IY. AS THERE ARE SO MANY COMMANDS THAT COME UNDER THIS BANNER I WILL NOT LIST THEM, BUT HERE IS A QUICK METHOD OF DERIVING THEM.

IF YOU WISH TO ACCESS THE IX PAIR, THEN DEFINE THE PRE-BYTE OF DD, OR FD FOR THE IY PAIR. YOU CAN THEN SIMPLY TYPE IN THE COMMAND AS USUAL BUT USING THE H REGISTER FOR THE HIGH BYTE AND THE L REGISTER FOR THE LOW BYTE. OBVIOUSLY YOU CANNOT USE THE H OR L REGISTER IN THIS SUB-SET.
FOR EXAMPLE, TO SWAP THE HIGH AND LOW BYTES OF IY AND IX,

```
DD 7C LD A,HX   DEFB 0DDH
                   LD A,H
DD 65 LD HX,LX  DEFB 0DDH
                   LD H,L
DD 6F LD LX,A   DEFB 0DDH
                   LD L,A
FD 7C LD A,HY   DEFB 0FDH
                   LD A,H
FD 65 LD HY,LY  DEFB 0FDH
                   LD H,L
FD 6F LD LY,A   DEFB 0FDH
                   LD L,A
```

THE COMMANDS AVAILABLE ARE -
LD ADD ADC SUB SBC INC DEC AND OR XOR CP

THERE ARE TWO REMAINING COMMANDS WHICH ARE SUB-SOMMANDS OF THE FUNCTIONS OUTI, OUTD, OTDR AND INI, IND, INIR, INDR. THESE ARE IN (HL), (C) (INPUT OT LOCATION (HL) FROM PORT (C) AND OUT (C),(HL) (OUTPUT FROM LOCATION (HL) TO PORT (C).

HEX	COMMAND
ED 70	IN (HL),(C)
ED 71	OUT (C),(HL)

--== o0o ==--

USING ZORLOF

SORRY, BUT I'VE NO ACCREDITATION FOR THIS

THIS IS AN EXCERPT FROM THE ZORLOF MANUAL (PAGES 1-6) USED AS AN EXAMPLE TO HELP YOU GET STARTED. NOTHING FANCY IS GOING ON HERE, JUST SOME OF THE MORE BASIC COMMANDS. IF YOU WOULD LIKE TO SEE THIS PRINT OUT ON YOUR PRINTER JUST CHANGE THE "K7" COMMAND IN THE NEXT PRINTER COMMAND LINE TO THE ONE WHICH SPECIFIES YOUR PRINTER TYPE (SEE SECTION 5.1 IN THE MANUAL). ALSO IF YOUR PRINTER IS SET TO ACCEPT A LINE-FEED AFTER THE CARRIAGE-RETURN THEN INSERT "LF," AS A COMMAND ON THAT LINE SOMEWHERE BEFORE THE LAST COMMAND (POJB). ONCE THESE TWO CHANGES HAVE BEEN MADE, HOLD DOWN THE "CLEAR" KEY AND WHILE DOING SO HIT THE "P" KEY.

1. GETTING STARTED

1.2. READ THE MANUAL BEFORE YOU DO ANYTHING ELSE. A THOROUGH UNDERSTANDING OF ITS CONTENTS IS IMPORTANT. ONCE THE MANUAL IS READ AND UNDERSTOOD, REMOVE THE SUMMARY CARD FROM THE BACK OF THIS MANUAL AND REFER TO IT AS YOU USE ZORLOF TO HELP YOU REMEMBER WHAT YOU READ IN THE MANUAL.

1.3. COPY ZORLOF AND THE EXAMPLE TEXT FILES FROM THE MASTER DISK TO YOUR DOS SYSTEM DISK. THE ZORLOF MASTER DISK CONTAINS NO SYSTEM SOFTWARE AND THUS WILL NOT OPERATE BY ITSELF IN DRIVE Ø. THE FILE NAMES WHICH YOU NEED TO COPY ARE ZORLOF/CMD, SAMPLE/TXT, FORM/TXT, FORM/DAT. SOME DOS'S REQUIRE 2 DISK DRIVES TO MAKE THIS COPY. WE DON'T ATTEMPT TO EXPLAIN THE PROCEDURE FOR COPYING FILES FROM A TRSDOS FORMATTED DISK TO OTHER OPERATING SYSTEM DISKS, SINCE IT IS DIFFERENT WITH EACH DOS. YOU WILL HAVE TO CONSULT YOUR DOS OPERATOR'S MANUAL FOR THIS INFORMATION. ONCE THE FILES FROM THE ORIGINAL ARE COPIED, STORE THE MASTER DISK AWAY IN A SAFE PLACE AND DON'T EVER USE IT FOR ANYTHING EXCEPT MAKING MORE COPIES. (IF YOU OWN MULTIDOS YOU WILL HAVE TO CHANGE THE SYSTEM DISK DATE FROM XX-XX-XX TO XX/XX/XX, USING THE PROT COMMAND.)

1.4. BOOT UP YOUR SYSTEM WITH A DOS DISK THAT CONTAINS ZORLOF/CMD. TYPE "ZORLOF", THEN HIT THE ENTER KEY. AFTER A FEW SECONDS THE SCREEN WILL BLANK EXCEPT FOR TWO STATUS LINES AT THE TOP OF THE SCREEN (EXPLAINED IN SECTION 3) AND A BLINKING RECTANGLE AT THE START OF THE NEXT LINE DOWN (THIS IS THE CURSOR, THE POINT AT WHICH YOUR TYPING TAKES PLACE). ZORLOF IS NOW RUNNING.

1.5. NOW YOU CAN BEGIN TYPING A TEXT FILE. MOVE YOUR CURSOR UP TO THE NAME FIELD WITH CLEAR-=. TYPE A NAME FOR YOUR FILE. MOVE THE CURSOR BACK DOWN TO THE THIRD LINE USING THE DOWN ARROW. TYPE A PRINTER COMMAND LINE STARTING WITH A PRINTER COMMAND CHARACTER (CLEAR-;) AND ENDING WITH A RETURN

CHARACTER (ENTER). THIS FIRST COMMAND LINE WILL CONTAIN YOUR MAIN FORMATTING COMMANDS SUCH AS YOUR PRINTER TYPE (5.1), TAB POSITIONS (5.4), MARGINS (5.9), ETC. PAGE NUMBERING PRINTER COMMAND LINES SHOULD FOLLOW RIGHT AFTER THIS. (REVIEW SECTIONS 5.28-37 AND ALSO SEE HOW IT'S DONE IN THE SAMPLE/TXT FILE).

1.6. MOVE YOUR CURSOR DOWN TO THE NEXT LINE AND TYPE, JUST LIKE YOU WOULD ON A TYPE WRITER. THE WORD WRAP MAKES SURE WORDS DON'T SPLIT BETWEEN LINES. THE ENTER KEY IS USED TO TERMINATE PARAGRAPHS AND TO CREATE BLANK LINES. THE TAB KEY IS CLEAR-K FOR PARAGRAPH INDENTING. CLEAR-L WILL DELETE A LINE. CLEAR- WILL DELETE CHARACTERS. CLEAR-I WILL PUT YOU INTO INSERT MODE AND ONCE AGAIN WILL TAKE YOU OUT. BOLDING, UNDERLINING, ITALICS, CONDENSED PRINT, EXPANDED PRINT AND MANY MORE TEXT ENHANCING COMMANDS ARE EXPLAINED IN SECTIONS 4 AND 5 OF THIS MANUAL.

1.7. ONCE THE TEXT IS FINISHED, YOU MAY VIEW IT IN NEAR FINAL FORM WITH CLEAR-V. IN THE VIEW MODE THE DOWN ARROW SCROLLS THE TEXT AND CLEAR-Z WILL GET YOU BACK TO TEXT EDITING. CLEAR-P WILL CAUSE THE TEXT TO BE PRINTED. CLEAR-Z CAN BE USED TO ABORT PRINTING TOO. WHEN DONE, STORE THE TEXT ON THE DISK WITH CLEAR-F, AND CLEAR THE SCREEN CLEAR-Ø.

1.8. THE DIRECTORY CAN BE VIEWED WITH CLEAR-D AND THEN THE NUMBER KEY OF THE DESIRED DRIVE. PAGING THROUGH THE DIRECTORY IS DONE WITH THE SPACE BAR. FILES CAN BE LOADED INTO MEMORY FROM THE DIRECTORY WITH CLEAR-G. AND FILES CAN BE KILLED FROM THE DIRECTORY USING CLEAR-K. CLEAR-Z WILL GET YOU OUT OF THE DIRECTORY.

1.9. AT THIS POINT WE SUGGEST YOU TAKE A LOOK AT, AND TRY PRINTING THE SAMPLE TEXT FILES WHICH WERE SUPPLIED ON THE ZORLOF MASTER DISK. SAMPLE/TXT WILL DEMONSTRATE MANY OF THE PRINT FORMAT COMMANDS AND HOW THEY WORK (EXPLAINED IN SECTION 5). FORM/TXT AND FORM/DAT TOGETHER GIVE AN EXAMPLE OF HOW A FORM LETTER TEXT FILE AND FORM LETTER DATA FILE MIGHT BE SET UP (EXPLAINED IN SECTION 7).

1.1Ø. ONCE YOU UNDERSTAND THE MANUAL AND HAVE TRIED SOME OF THE EXAMPLES, WE DON'T EXPECT YOU TO HAVE ANY DIFFICULTIES USING ZORLOF. EVEN THOUGH ZORLOF IS AN EXTREMELY POWERFUL TEXT PROCESSING TOOL, A LOT OF CARE AND THOUGHT HAS GONE INTO MAKING IT AS EASY TO USE AS POSSIBLE. HOWEVER, SHOULD YOU NEED ANY ASSISTANCE OR HAVE A QUESTION CONCERNING THIS PRODUCT, FEEL FREE TO GIVE US A CALL.

WE WOULD LIKE TO PREFACE THIS OPEN ENDED INVITATION BY POINTING OUT THAT ALMOST EVERY CALL FOR HELP THAT WE RECEIVE HAS BEEN THE RESULT OF EITHER THE USER NOT UNDERSTANDING HOW TO DO A "COPY" OR SOME OTHER SIMILAR FUNCTION ON HIS DOS, OR BECAUSE HE HASN'T TAKEN THE TIME TO READ THROUGH THE ZORLOF MANUAL

AND STUDY THE SECTIONS WHICH APPLY TO HIS PROBLEM AREA. JUST ABOUT EVERYTHING ONE WOULD EVER NEED TO KNOW ABOUT ZORLOF IS ALREADY CONTAINED IN THIS MANUAL. SO, TO SAVE YOURSELF THE COST OF A TELEPHONE CALL, GIVE THE MANUAL ONE MORE CHANCE, AND USE THE EXAMPLE TEXT FILES SUPPLIED ON THE DISK AS A GUIDE. YOU WILL PROBABLY FIND THE ANSWERS YOU NEED.

2.
WORD-WRAP AND JUSTIFICATION

THE OPERATOR OF ZORLOF IS FREE TO TYPE CONTINUOUSLY, NEVER WORRYING ABOUT WHERE TO BREAK THE LINES, OR HOW THE SPACING BETWEEN THE WORDS SHOULD BE TO ACHIEVE PROPER LINE JUSTIFICATION. ZORLOF DOES THIS ALL FOR YOU AUTOMATICALLY, USING ITS ADVANCED WORD-WRAP AND LINE JUSTIFICATION CAPABILITIES WHICH MATCH AND EVEN SURPASS THOSE FOUND ON MANY OF THE HIGHER PRICED WORD PROCESSING SYSTEMS. BOTH OF THESE FEATURES WILL BECOME VERY POWERFUL TEXT EDITING TOOLS FOR YOU ONCE THEY ARE UNDERSTOOD

2.1.
WORD-WRAP

IF A WORD THAT YOU ARE TYPING WILL NOT FIT ON THE LINE WITHOUT SPILLING OVER THE END OF THE LINE (SEE 3.2 FOR LINE LENGTH SPECIFICATIONS), THEN ZORLOF WILL AUTOMATICALLY TAKE THAT WORD OFF THAT LINE AND PUT IT ON THE NEXT LINE. AT THE SAME TIME, THE LINE YOU WERE TYPING ON AUTOMATICALLY JUSTIFIES ON THE SCREEN.

ANOTHER PART OF THE WORD-WRAP CAPABILITIES OF ZORLOF IS REVERSE WORD-WRAP. ANY TIME YOU TYPE A SPACE ON A LINE, ZORLOF WILL CHECK TO SEE IF THE WORD(S) FROM THE BEGINNING OF THE LINE TO THAT SPACE CAN FIT ON THE PREVIOUS LINE. IF IT CAN THEN IT IS "WRAPPED BACKWARDS", SO TO SPEAK, AND PUT AT THE END OF THE PREVIOUS LINE. THE PREVIOUS LINE IS THEN RE-JUSTIFIED AND THE CURSOR IS MOVED BACK TO THE SECOND COLUMN OF THE CURRENT LINE (SECOND, BECAUSE THE SPACE IS IN THE FIRST). ZORLOF WILL CONTINUE TO MOVE WORDS UP TO THE END OF THE PREVIOUS LINE WITH EACH SPACE YOU TYPE UNTIL THE PREVIOUS LINE IS FULL.

WHILE SCROLLING UP, THE WORD-WRAP FUNCTION IN ZORLOF WILL ALWAYS PUT AS MANY WORDS ON A LINE AS WILL FIT. LINES CONTAINING A RETURN CHARACTER ()(4.47), HOWEVER, ARE THE EXCEPTION. NO TEXT WILL EVER BE PUT ON A LINE FOLLOWING A RETURN CHARACTER WHILE SCROLLING.

2.2.
LINE JUSTIFICATION

ZORLOF HAS VERY ADVANCED LINE JUSTIFICATION ROUTINES WHICH AUTOMATICALLY JUSTIFY THE LINES OF YOUR TEXT NOT ONLY AS THEY ARE PRINTED, GIVING YOU A VERY PROFESSIONAL-LOOKING COPY, BUT ALSO RIGHT ON THE SCREEN AS YOU TYPE, GIVING YOU THE OPPORTUNITY TO SEE HOW THE TEXT IS GOING TO LOOK BEFORE YOU EVER PRINT IT. YOU CAN FORMAT YOUR TEXT

USING ONE (OR A COMBINATION) OF FOUR DIFFERENT LINE JUSTIFICATION MODES. THEY ARE:

JUSTIFY LEFT
ALL TEXT LINES START IN THE FIRST CHARACTER COLUMN TO FORM A SMOOTH MARGIN ON THE LEFT SIDE OF THE SCREEN AND PAPER.

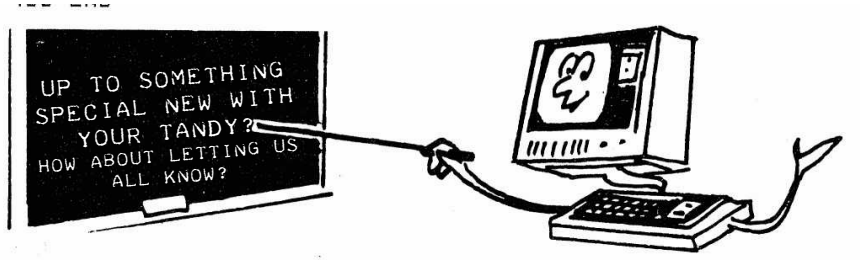
JUSTIFY RIGHT
ALL TEXT LINES END IN THE LAST CHARACTER COLUMN TO FORM A SMOOTH MARGIN ON THE RIGHT SIDE OF THE SCREEN AND PAPER.

JUSTIFY BOTH
ALL TEXT LINES START AND END IN THE FIRST AND LAST CHARACTER COLUMNS, RESPECTIVELY, TO FORM SMOOTH MARGINS ON BOTH SIDES OF THE SCREEN AND PAPER. THE SPACING BETWEEN WORDS IS DONE AUTOMATICALLY.

JUSTIFY CENTER
ALL TEXT LINES ARE CENTRED ON SCREEN AND ON PAPER. IF DESIRED, YOU CAN CHANGE THE LINE JUSTIFICATION MODE AT ANY PLACE IN YOUR TEXT--HAVING ONE PARAGRAPH SET TO JUSTIFY LEFT, THE NEXT TO JUSTIFY CENTER, ETC. THE DEFAULT JUSTIFICATION MODE IS JUSTIFY LEFT. REFER TO SECTION 5.11 FOR INSTRUCTIONS ON HOW TO SET THE LINE JUSTIFICATION MODE WITHIN YOUR TEXT.

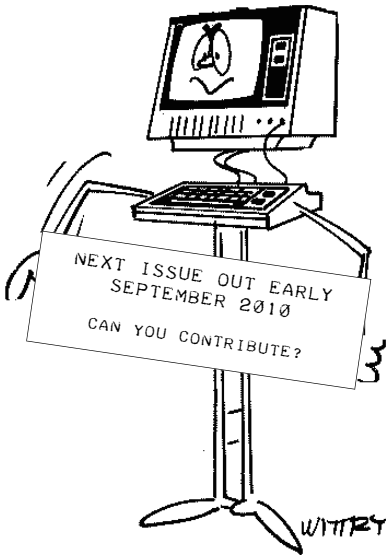
IF YOU HAPPEN TO OWN A PRINTER TYPE K3, K7, K8, K9, K11, K13, K14, K16, K20 (5.1), YOU WILL BE PLEASED TO KNOW THAT ZORLOF CAN TAKE ADVANTAGE OF THE FULL POWER OF THE PROPORTIONAL-SPACE TEXT-PRINTING FEATURE THAT THESE PRINTERS HAVE. EACH LINE WILL BE JUSTIFIED, AS WITH ANY PRINTER, BUT ALL THE BETWEEN-WORD SPACING WILL BE MADE EXACTLY EQUAL AND THE SPACES BETWEEN CHARACTERS WILL BE UNIFORMLY INCREASED TO AVOID EXCESSIVE BETWEEN-WORD SPACING, WHENEVER NECESSARY. THE RESULT IS A PRINTED PAGE THAT LOOKS ALMOST LIKE IT WAS RUN OFF ON AN EXPENSIVE TYPESETTER.

USING ZORLOF WITH ANY ONE OF THESE TYPES OF PRINTERS, YOU CAN ALSO SPECIFY ANY CHARACTER PRINT DENSITY FROM 6.0 TO 20.0 CHARACTERS PER LINE INCH. THE MAXIMUM PRINT DENSITY YOU WILL ACTUALLY BE ABLE TO ACHIEVE AND STILL HAVE LINES CORRECTLY JUSTIFY WILL DEPEND ON THE AVERAGE CHARACTER WIDTH FOR YOUR PRINTER. ABOUT 14 CHARACTERS PER INCH IS AS TIGHT AS YOU WILL NORMALLY BE ABLE TO PRINT ON A DOT MATRIX PRINTER, AND ABOUT 11 CHARACTERS PER INCH ON A DAISY WHEEL. THIS MANUAL WAS PRINTED IN PROPORTIONAL-SPACE MODE AT A CHARACTER DENSITY OF 11.0 CHARACTERS PER INCH USING NEC PC-8023A-C PRINTER. REFER TO SECTION 5.16 FOR INSTRUCTIONS ON HOW TO SPECIFY CHARACTER DENSITY WITHIN YOUR TEXT.



POKEING CHANGES TO A PROGRAM

THE ORIGINAL MODEL 1 I OWNED, BACK IN 1979, WAS ONE OF THE EARLY U.K. IMPORT TYPES WHICH DID NOT HAVE A SEPARATE NUMERIC KEYPAD ON THE RIGHT SIDE OF THE KEYBOARD. I HAVE OFTEN HEARD IT SAID THAT WHAT YOU'VE NEVER HAD, YOU DON'T MISS. THAT WAS TRUE UNTIL ALL THE MODEL 1 KEYBOARDS THAT BEGAN TO APPEAR AROUND ME (BACK IN THE EARLY 1980'S) HAD THE KEY PAD AND I COULD SEE WHAT A BOON THAT WAS FOR ALL NUMERIC INPUT.



THE MODEL 1 I NOW OWN HAS THE KEYPAD AND ALTHOUGH I TRY NOT TO USE THE COMPUTER TOO OFTEN, (I TEND TO KEEP IT FOR 'BEST' SO TO SPEAK), I DO NOTICE HOW USEFUL THE NUMERIC KEYPAD IS. THE SAME APPLIES TO MATTHEW REED'S EMULATOR. USING THE PC KEYBOARD, IT IS SECOND NATURE TO USE THE KEYPAD FOR ANY NUMERIC ENTRIES I PERFORM. THE ONLY DRAW BACK I ENCOUNTER, WHICH IS AS TRUE NOW AS IT WAS IN THE EARLY 1980'S IS THAT THE NUMERIC KEYPAD ONLY HAS A FULL STOP. THIS IS GREAT WHEN ENTERING NUMERICAL AMOUNTS, BUT WHEN ENTERING DATA STATEMENTS WITHIN A BASIC PROGRAM, YOU NEED A COMMA TO SEPARATE THEM, NOT A FULL-STOP.

THE NORMAL WAY I USUALLY ATTEMPTED TO GET AROUND THIS PROBLEM WAS TO USE THE LEVEL 2 EDITOR WITH THE (S) SEARCH AND (C) CHANGE SUB-COMMANDS. NOT IDEAL THOUGH, AS IT WAS TIME CONSUMING AND CERTAINLY NOT VERY ELEGANT. I REMEMBER, I HAD ON OCCASIONS USED ELECTRIC PENCIL, AND THE REPETITIVE CHANGE COMMAND, BUT WHILE THIS IS NOT TOO TIRESOME IN A DISK SYSTEM, IN LEVEL 2, MESSING ABOUT WITH THE PROGRAM CASSETTE AND THE INEVITABLE LOADING PROBLEMS, MADE IT A 'NO-NO'.

SO THE OTHER DAY I HAD THIS 'MAD' IDEA TO SEE IF I COULD FIND A SOLUTION TO THIS 25 YEAR OLD ANNOYANCE! LIKE MANY OF MY IDEAS, I WANTED IT TO BE SIMPLE! (A ONE-LINER EVEN!). I ASSUMED THAT THE EASIEST WAY WAS TO CHANGE THE OFFENDING FULL-STOPS TO A COMMA, WOULD BE BY PEEKING AND POKEING IN MEMORY. THE MAIN PROBLEMS, I THOUGHT, WAS WHERE DO YOU BEGIN SUCH A TASK AND ALSO, WHERE/WHEN DO YOU STOP. SO AFTER GETTING OUT MY TRUSTY COPY OF EDWIN PAAY'S BOOK, ASSEMBLY LANGUAGE TOOLKIT AND SEARCHING OUT THE START AND END OF A BASIC PROGRAMS ADDRESS, THE FOLLOWING BASIC PROGRAM, (WHICH IS USED PURELY AS AN EXAMPLE), TOOK SHAPE.

LINES 10-40 COULD EASILY BE CONDENSED DOWN TO ONE LINE, OR EVEN ENTERED DIRECTLY AT THE READY PROMPT. A NO-LINER IN EFFECT! (EVEN BETTER)

LINE 10 FINDS THE END OF THE CURRENT BASIC PROGRAM.

LINE 20 SETS UP A LOOP FROM THE START OF THE BASIC PROGRAM TO THE END. THE NUMBER 27172 (6A24H) IS THE START ADDRESS OF BASIC PROGRAMS WITHIN ND80 V2. IF YOU ARE RUNNING THIS IN LEVEL 2, YOU SHOULD REPLACE IT WITH 17129 (42E9H). I'M NOT SURE WHAT THE ADDRESS IS IN LDOS. COULD SOMEONE ENLIGHTEN US PLEASE?

LINE 30 TRAWLS THROUGH THE BASIC PROGRAM, LOOKING FOR A FULL-STOP, ASCII 46, AND REPLACES IT WITH A COMMA, (ASCII 44).

** PLEASE BE AWARE THAT ALL FULL-STOPS WITHIN THE PROGRAM WILL BE REPLACED. EVEN ONES PLACED WITHIN A STRING VARIABLE WHICH, PERHAPS, YOU DON'T NECESSARILY WANT CHANGING! **

LINE 40 COMPLETES THE FOR/ NEXT LOOP.

LINES 50 AND 60 ARE JUST EXAMPLES OF DATA STATEMENTS ENTERED WITH A FULL-STOP FROM THE NUMERIC KEYPAD AND WILL BE CHANGED TO A COMMA WHEN THE PROGRAM IS RUN.

```
10 J=PEEK(16633)+(256*PEEK(16634))
20    FOR X=27172 TO J
30    IF PEEK(X)=46THENPOKEX,44
40    NEXT X
50 DATA 23.25.65.45.78.59
60 DATA 78.123.589.4.14.58
```

--== o0o ==--

HERE'S A ONE-LINER, WHICH, IS ONCE AGAIN, COMPETING WITH DEE'S NINTENDO DS.

THE GRAND CHILDREN SEEMED IMPRESSED WITH THE IDEA OF REMEMBERING A SERIES OF NUMBERS GENERATED ON A COMPUTER. SO GRANDDAD SET TO ON HIS TANDY!
THE ORIGINAL PROGRAM WAS A SOMEWHAT LARGER AFFAIR, REMEMBERING NAMES AND HAVING A TIMER ETC., BUT AS I'VE NOT HAD ANY ONE-LINERS SENT IN FOR SOME CONSIDERABLE SOMETIME, I THOUGHT THIS WAS AN IDEAL CANDIDATE.
TO MAKE IT MORE DIFFICULT, (I.E. A LONGER LIST OF NUMBERS TO REMEMBER) JUST INCREASE THE VARIABLE 'A' TO A LARGER AMOUNT.
12 OR GREATER NUMBERS ARE QUITE DIFFICULT TO COPE WITH! YOU WILL NEED SOME PRACTICE!
I HOPE YOU GET A BIT OF FUN OUT OF IT.

```
0 DEFDBLC,F:INPUT"PRESS ENTER";G:CLS:A=RND(4)+3:FORB=1TOA:A$=A$+
CHR$(RND(10)+47):NEXT:C=VAL(A$):FORD=1TOB:PRINTMID$(A$,D,1);" ";
:FORE=1TOA*30:NEXTE:NEXTD:CLS:INPUT"ENTER THE NUMBER";F:IFF=CTHE
NPRINT"CORRECT":RUNELSEPRINT"OOPS IT WAS";C:RUN
```

REX2

NEWS OF AN UPDATE
(FROM THE TRS-80 FORUM)

REX2 IS A SUCCESSOR, A 'SEQUEL' IF YOU LIKE, OF REX. REX WAS A FLASH MEMORY CARD FOR THE MODEL 100 THAT 'EMULATED' RAM/OPTION ROMS IN 32K BANKS ON FLASH MEMORY. IT IS A SMALL DEVICE THAT GOES IN THE OPTROM SLOT, AND REQUIRES NO HARDWARE ENHANCEMENT. REX2 IS PHYSICALLY SIMILAR EXCEPT FOR A CONNECTED CARD THAT GOES ON THE EXPANSION BUS OF THE M100 TO PROVIDE SOME REQUIRED SIGNALS. AS OF NOW, REX2 IS A LIMITED-RELEASE NON-NON-PUBLICLY AVAILABLE PRODUCT MADE BY THE FANTASTIC STEPHEN ADOLPH.

MTCPM

HOW MANY OF YOU HAVE USED CP/M ON A VINTAGE PORTABLE THE SIZE OF A BINDER? OKAY, THAT ONE GUY IN THE BACK CAN PUT HIS HAND DOWN. MTCPM (MODEL T CONTROL PROGRAM FOR MICROCOMPUTERS) IS A PORT OF CP/M TO THE TRS-80 MODEL 100 SERIES OF PORTABLES. THIS IS MADE POSSIBLE BY THE NEW REX2, WHICH HAS SOME RESOURCES SET ASIDE ESPECIALLY FOR CP/M. AFAIK, MTCPM WILL BE SOFTWARE-COMPATIBLE TO 'MAINSTREAM' CP/M, AND YOU SHOULD BE ABLE TO RUN MOST (IF NOT ALL) COMMAND-LINE OR TEXT-BASED APPLICATIONS ON IT (GRAPHICS ARE IFFY ON AN 8X40 CHAR. SCREEN

--== o0o ==--

THE ULTIMATE IN SOFTWARE UNPROTECTION

FPS-3 IS A FRONT PANEL SIMULATOR FOR THE TANDY CORP. TRS80 MODEL III

JUST FLIP A SWITCH AND! ! !

THE PROGRAM IN MEMORY IS COPIED TO YOUR CHOICE OF DISK OR TAPE. TO RUN THE COPY SIMPLY BOOT THE DISK FROM RESET OR LOAD THE TAPE WITH THE SYSTEM COMMAND. YOU DO NOT NEED ANY TECHNICAL KNOWLEDGE TO USE THE FPS-3. ALL YOU NEED TO INSTALL THE FPS-3 ARE A HALF HOUR OF YOUR TIME AND A SCREWDRIVER. THE COST FOR A COMPLETE FPS-3 IS ONLY $50.

WHAT THE SOFTWARE GODS HAVE HIDDEN

THIS LITTLE SHALL REVEAL DEVIL

J.E.S. GRAPHICS, P.O. BOX 2752, TULSA, OK. 74101 CALL 918 742 7104.

M100 BAR CODES

(AND OTHER BITS!)

I'VE HEARD IT SAID ON MANY OCCASIONS THAT AS ONE DOOR CLOSES, ANOTHER OPENS AND SO IT WAS THE OTHER WEEK. I WAS HAPPILY BIDDING ON A BAR-CODE READER (COMPETE WITH SOFTWARE!), FOR MY OLD BUT TRUSTY M102, AND TRUE TO FORM, WAS 'SNIPED' IN THE LAST FEW SECONDS. AS DEE OFTEN REMARKS, 'YOU CAN'T HAVE IT ALL - THERE'S NO MORE ROOM FOR YOUR CLUTTER ANYWAY!'. AH WELL, I THOUGHT, I'LL JUST HAVE TO BE SATISFIED WITH THE JUNE 1983 EDITION OF 'COMPUTRONICS' I (SECRETLY) 'WON', THANKS TO EBAY. I RAPIDLY SENT VIA PAYPAL, THE FABULOUS SUM OF 99P, AND WITHIN 48 HOURS I WAS THUMBING MY WAY THROUGH IT.

THE ARTICLE WHICH HAD CAUGHT MY EYE WAS A REVIEW OF THE THEN, JUST RELEASED, M100. IT ACTUALLY MENTIONS THE BAR-CODE READER PORT, AND SOMETHING WHICH I WASN'T AWARE OF, THAT IT CAN BE USED WITH 'SEVERAL AVAILABLE BAR-CODE WANDS'. THE ONLY ONE MENTIONED IS THE HEWLETT-PACKARD HEDS-3000. ANOTHER EAGLE-EYE TO BE PLACED ON MY EBAY WISH-LIST.

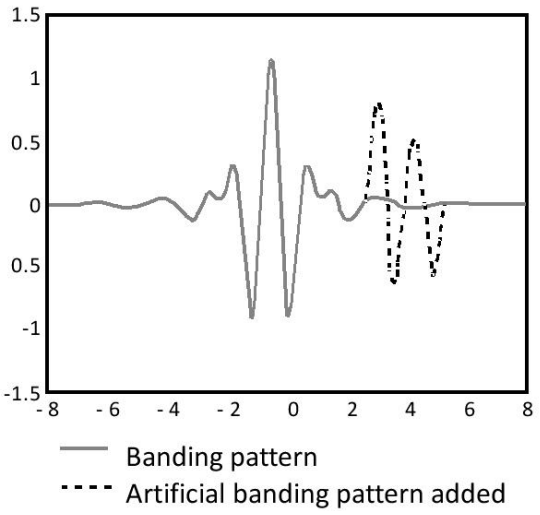
I ALSO NOTICED AN ADVERTISEMENT ON THE SAME PAGE AS THE REVIEW, FOR 'DEVIL' THE ULTIMATE IN SOFTWARE UNPROTECTION! DOES ANYONE HAVE ANY MORE INFORMATION ABOUT IT? CAN ANYONE REMEMBER BUYING AND/OR USE IT? I CAN'T EVER RECALL EITHER SEEING IT OR COMING ACROSS A REVIEW OF IT. I LOVE THE SALES BLURB THOUGH - 'WHAT THE SOFTWARE GODS HAVE HIDDEN, THIS LITTLE DEVIL SHALL REVEAL'.

STEGANOGRAPHY

LES TAYLOR

THIS ARTICLE FIRST APPEARED IN SMALL PRINTER

IN TODAY'S DIGITAL WORLD SECURING CONTENT IS VERY IMPORTANT IN TERMS OF VERIFYING THE AUTHENTICITY AND PROTECTING COPY-RIGHT OF A DOCUMENT. PRINTED MATERIAL IS A DIRECT ACCESSORY TO CRIMINAL ACTS WITH FORGERY OR ALTERATION OF DOCUMENTS USED FOR PURPOSES OF IDENTITY, SECURITY OR RECORDING TRANSACTIONS. AS WELL AS TERRORIST ACTIVITIES INCLUDING INSTRUCTION MANUALS, TEAM ROSTERS, CORRESPONDENCE AND MEETING NOTES. LAW ENFORCEMENT WOULD FIND IT VERY DESIRABLE TO BE ABLE TO IDENTIFY WHAT PRESS WAS USED, WHEN IT WAS PRINTED AND WHO PRINTED IT.

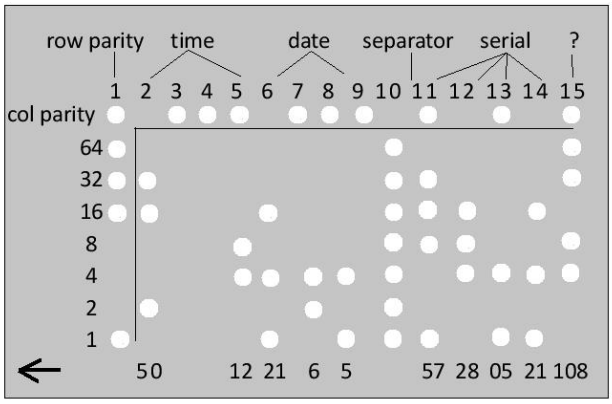


MANY TECHNIQUES ARE AVAILABLE TO PROTECT AUDIO, VIDEO AND THE EXECUTABLE CODE OF PROGRAMS. THE SECURING OF DOCUMENTS IS NOT NEW; IN 1282 WATERMARKS BEGAN TO APPEAR IN PAPER. THESE ARE CREATED BY PLACING THIN WIRES IN THE MOULD USED TO CREATE THE PAPER AND ARE USED TO IDENTIFY THE PAPER MAKER OR THE SPECIFIC MOULD USED TO CREATE THE PAPER. BY THE EIGHTEENTH CENTURY WATERMARKS APPEARED IN BANK NOTES TO DETER COUNTERFEITING AND ALSO ON REGULAR PARCHMENT TO DESIGNATE A TRADE-MARK, DATE OR PLACE. TODAY, WATERMARKING IS USED IN CONJUNCTION WITH OTHER SECURITY FEATURES LIKE COLOURED FIBRES EMBEDDED WITHIN THE PAPER DURING MANUFACTURE, SPECIAL INKS THAT CHANGE COLOUR, HOLOGRAMS AND MICRO TEXT.

PROFESSORS JAN ALLEBACH AND EDWARD DELP FROM THE SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING HAVE A PASSION TO TRACE DOCUMENTS THAT HAVE BEEN PRINTED ON LOW COST CONSUMER PRINTERS. THE FIRST STRATEGY IS PASSIVE AND INVOLVES CHARACTERISING THE PRINTER AND FINDING FEATURES ON THE PRINTED PAGE THAT ARE CHARACTERISTIC OF A PARTICULAR MODEL OF PRINTER, THIS IS CALLED THE INTRINSIC SIGNATURE. IN ELECTRO-PHOTOGRAPHIC (EP) PRINTERS THIN LINES RUNNING ACROSS THE PRINT CAN BE SEEN WITHIN LARGE MID-TONE AREAS" SAYS PROF ALLEBACH, "THIS IS KNOW AS 'BANDING' CAUSED BY THE GEARS WITHIN THE PRINTER MECHANISM". A SECOND FAULT CALLED 'GHOSTING' SHOWS REPEATED RESIDUAL WEAKER IMAGES OF A PREVIOUSLY PRINTED IMAGE IN THE DIRECTION OF PRINT. THIS IS ENOUGH FOR PROF CHIU AND HIS TEAMS OF GRADUATES USING WAVELET FILTERING AND TEMPLATE MATCHING TECHNIQUE SO THEY CAN IDENTIFY ANY PRINTER MODEL WITH PINPOINT ACCURACY.

"IN THE RESEARCH" ADDED PROF DELP "WE FOUND THAT IT WAS EASIER TO TAKE MEASUREMENTS FROM INDIVIDUAL LETTERS, AND THAT AN "ARTIFICIAL" BANDING SIGNAL COULD BE INCLUDED TO ENCODE INFORMATION, SUCH AS THE PRINTER SERIAL NUMBER AND DATE OF PRINTING, IN EVERY PRINTED PAGE. THESE SIGNAL FREQUENCIES AND AMPLITUDES LIE BELOW THE HUMAN VISUAL SENSITIVITY".

THIS WORK HAS LED TO A SECOND STRATEGY LED BY PROFESSOR GEORGE CHIU FROM THE SCHOOL OF MECHANICAL ENGINEERING ALSO AT PURDUE UNIVERSITY USING THE PRINTER MECHANISM TO EMBED INTO THE PRINTED PAGE IDENTIFYING INFORMATION. "THINK OF THIS AS A WATERMARK" SAID PROF CHIU, "SHIFTING INDIVIDUAL CHARACTERS WORDS OR LINES, IS VERY POPULAR FOR HIDING MESSAGES IN PLAIN VIEW". THESE ELEMENTS ARE SHIFTED UP OR DOWN, APPROXIMATELY 1/600TH INCH, TO ENCODE A ONE OR ZERO, THE BINARY CODE ALL COMPUTERS WORK WITH". THIS METHOD IS ROBUST AGAINST ANY SCAN-PRINT ATTACK OR PHOTOCOPY GENERATION LOSS. ALTHOUGH THE PAGE CAN BE TRACED BACK TO A PARTICULAR PERSON, IT CANNOT BE USED TO DETERMINE WHETHER THE DOCUMENT HAS BEEN ALTERED OR IF IT IS THE ORIGINAL LEGAL COPY. FOR THIS A FRAGILE WATERMARK CAN BE USED TO DETERMINE IF ANY PART OF THE DOCUMENT HAS BEEN ALTERED AS THE WATERMARK FAILS TO BE DETECTED AFTER THE SLIGHTEST CHANGES IN THE DOCUMENT ARE MADE.



How to translate the hidden dots.
2005-06-21 12:50 serial 21052857

AND FINALLY, BACK IN 2004 AN ARTICLE APPEARED IN PC WORLD STATING THAT PRINTER COMPANIES HAVE BEEN QUIETLY ENCODING THE SERIAL NUMBER AND THE MANUFACTURING CODE OF THEIR COLOUR LASER PRINTERS AND COLOUR COPIERS ON EVERY DOCUMENT THOSE MACHINES PRODUCE. PETER CREAN, A SENIOR RESEARCH FELLOW AT XEROX, SAYS HIS COMPANY'S LASER PRINTERS, COPIERS AND MULTIFUNCTION WORK-STATIONS PUT THE "SERIAL NUMBER OF EACH MACHINE CODED IN LITTLE YELLOW DOTS" IN EVERY PRINTOUT. THE MILLIMETRE-SIZED DOTS APPEAR ABOUT EVERY INCH ON A PAGE, NESTLED WITHIN THE PRINTED WORDS AND MARGINS. "IT'S A CLEAR TRAIL BACK TO YOU, LIKE A LICENSE PLATE," CREAN SAID, "AND, IT'S BEEN IN PRINTERS FOR DECADES".

NEXT TIME YOU MAKE A PRINTOUT FROM YOUR COLOUR LASER PRINTER, SHINE A LED FLASHLIGHT BEAM ON IT AND EXAMINE IT CLOSELY WITH A MAGNIFYING GLASS. YOU MIGHT BE ABLE TO SEE THE SMALL, SCATTERED YELLOW DOTS PRINTED THERE THAT COULD BE USED TO TRACE THE DOCUMENT BACK TO YOU. THE ENCODING MECHANISM IS WITHIN A CHIP THAT IS EMBEDDED NEAR THE LASER; DON'T BOTHER TRYING

TO DISABLE THE MECHANISM AS YOU'LL PROBABLY JUST BREAK YOUR PRINTER.

THE POSSIBLE MISUSES OF THIS MARKING TECHNOLOGY ARE FRIGHTENING, INDIVIDUALS USING PRINTERS TO CREATE POLITICAL PAMPHLETS, ORGANIZE LEGAL PROTEST ACTIVITIES, OR EVEN DISCUSS PRIVATE MEDICAL CONDITIONS OR SENSITIVE PERSONAL TOPICS CAN BE IDENTIFIED BY THE GOVERNMENT WITH NO LEGAL PROCESS, NO JUDICIAL OVERSIGHT, AND NO NOTICE TO THE PERSON SPIED UPON. IF YOU WANT TO KNOW MORE ABOUT THIS SUBJECT THEN CONTACT THE ELECTRONIC FRONTIER FOUNDATION (EFF) A BODY OF PEOPLE WATCHING THOSE WHO ARE WATCHING YOU. MORE INFORMATION IS AVAILABLE ON THEIR WEBSITE AT:

HTTP://W2.EFF.ORG/PRIVACY/PRINTERS/DOCUCOLOR/INDEX.PHP

STEGANOGRAPHY IS THE ART OF WRITING HIDDEN MESSAGES IN SUCH A WAY THAT NO ONE, APART FROM THE SENDER AND INTENDED RECIPIENT, SUSPECTS THE EXISTENCE OF THE MESSAGE.

I FIRST CAME ACROSS THIS FASCINATING ARTICLE WHEN LES COMPILED IT FOR 'SMALL PRINTER' THE OFFICIAL MAGAZINE OF THE BRITISH PRINTING SOCIETY. I FIND IT A LITTLE UN-NERVING TO SAY THE LEAST!

CHECK OUT THE EFF WEBSITE IF YOU GET A CHANCE, I THINK THEY DESERVE MORE THAN A LITTLE SUPPORT.

---== o0o ==---

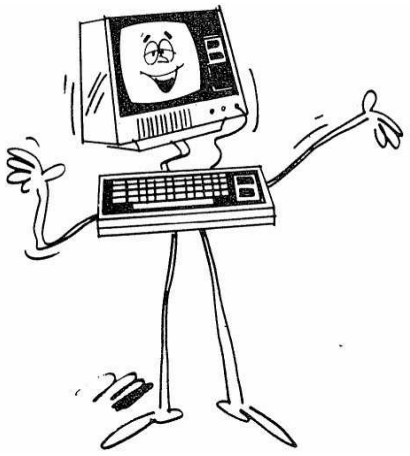
BITS, BYTES & NIBBLES (1)

BY PETER STONE.

THE FIRST IN AN OCCASIONAL SERIES OF TRS-80 PROGRAMMING TRIVIA TIPS.

IF YOU HAVE A PROGRAM THAT NEEDS TO TELL WHETHER IT'S RUNNING ON A MODEL1 OR A MODEL3, THEN CARRY OUT A PEEK 125H. THIS WILL BE 49H IN A MODEL3 & 20H IN A MODEL1.

THIS IS TO DO WITH THE FACT THAT THE RADIO SHACK LEVEL II BASIC MESSAGE STARTS AT MEMORY LOCATION 111H. THIS MEANS THAT MEMORY LOCATIONS 123H & 124H CONTAIN THE II IN THE ABOVE MESSAGE, & MEMORY LOCATION 125H CONTAINS A SPACE (20H). HOWEVER IN A MODEL 3 THE MESSAGE IS RADIO SHACK MODEL III BASIC, SO MEMORY LOCATION 125H CONTAINS 49H, (73 OR THE LETTER I) INSTEAD OF 20H.



HAPPINESS IS A NOT TOO-WARM M1 TANDY!

REPEATABLE RANDOMNESS

BY PETER STONE

THE ORIGINAL IDEA FOR THIS ARTICLE CAME FROM MY 'ADDICTION' TO SPIDER SOLITAIRE. I'VE PLAYED THIS GAME SINCE 2003 & NEVER ACHIEVED A WIN RATE ABOVE 35%. THIS & OTHER INDICATORS HAVE LED ME TO THE CONCLUSION THAT THE PROGRAM 'CHEATS', AS A FAIR GAME SHOULD, IN THEORY, PRODUCE A WIN RATE OF 50%. I BECAME SO DISILLUSIONED WITH SPIDER SOLITAIRE, THAT I SWITCHED BACK TO PLAYING FREECELL. I THEN NOTICED A 'PROBLEM' THAT I WANTED TO EXPLORE. NO MATTER HOW MANY GAMES I'VE RECENTLY WON OR LOST, THE PERCENTAGE WIN NOW SEEMS 'STUCK' AT 59%. MY FREECELL STATISTICS ARE WON - 1777 & LOST - 1260. SO I DECIDED TO INVESTIGATE THIS.

I DECIDED TO WRITE THIS PROGRAM ON THE TRS80, PARTLY AS AN EXAMPLE OF HOW TO PROGRAM THE RANDOM NUMBER GENERATOR AFTER THE 'ASSEMBLY MADE EASY' ARTICLE I WROTE FOR TRS8BIT0304, WHICH SHOWED THE LOCATIONS USED TO SEED THE RANDOM NUMBER GENERATOR, & PARTLY OUT OF NOSTALGIA.

IF YOU SWITCH ON A TRS80, ENTER THE PROGRAM BELOW, & RUN IT;

```
10 FOR X=1 TO 10
20 A=RND(6)
30 PRINT A,
40 NEXT
```

THE SEQUENCE OF NUMBERS 3,2,1,5,6,1,4,3,4,2 WILL BE GENERATED. IF YOU SWITCH THE MACHINE OFF, SWITCH IT ON AGAIN & RE-ENTER THE PROGRAM, THE SAME SEQUENCE OF NUMBERS WILL BE GENERATED. TO PRODUCE A DIFFERENT RANDOM STRING OF NUMBERS EACH TIME, THE LINE

```
 5 RANDOM
```

WOULD HAVE TO BE ADDED.

AT THIS POINT EXPLAINING ABOUT THE RND(X) COMMAND WOULD BE OF HELP. THE TANDY BASIC MANUAL STATES THAT IF YOU MAKE THE VALUE OF X EQUAL O ZERO, IE RND(0), THEN A SINGLE-PRECISION VALUE BETWEEN 0 & 1 IS GENERATED, (EG .423392, .173153, .120223, .77285, .927776, .128992, .515352, .391262, .631054, .307911, .0192572, .121761). HOWEVER IF X IS MADE EQUAL TO AN INTEGER NUMBER, WHICH MUST BE POSITIVE & NO GREATER THAN 32767, THEN AN INTEGER BETWEEN 1 & THE VALUE OF X IS RETURNED. AS AN EXAMPLE OF USING RND(0), WE COULD MULTIPLY EACH OF THE NUMBERS GENERATED BY 100,000 & THEN DOING AN INT ON THE NUMBER TO PRODUCE RANDOM NUMBERS GREATER THAN 32767.

SWITCHING THE MACHINE OFF & ON, & RELOADING A PROGRAM EACH TIME TO PRODUCE A REPEATABLE STRING OF RANDOM NUMBERS IS CUMBERSOME, ESPECIALLY IF YOU ARE FINE TUNING THE WIN/LOOSE THRESHOLD, SO A SIMPLER WAY OF DOING IT, IF AVAILABLE, IS PREFERABLE. THE TRS80 HAS FOUR MEMORY LOCATIONS, 16554, 16555, 16556 & 16557 WHICH ARE USED BY THE RANDOM NUMBER GENERATOR, SO SETTING THESE TO THE SAME VALUE EACH TIME THE PROGRAM IS RUN ACHIEVES THIS. FROM 'MICROSOFT BASIC DECODED & OTHER MYSTERIES', 16554 IS LABELLED AS RANDOM NUMBER SEED, 16555 CONTAINS A VALUE

FROM THE Z80 REFRESH REGISTER, & LOCATIONS 16556 & 16557 ARE THE LAST RANDOM NUMBER EXPRESSED AS TWO BYTES. WITH THIS IN MIND I WROTE THE PROGRAM BELOW. IT HAS BEEN WRITTEN FOR EASE OF UNDERSTANDING & DEBUGGING, & HAS NOT BEEN OPTIMISED IN ANY WAY. THIS IS LEFT AS AN EXERCISE FOR THE READER, AS MOST OF THE TEXTBOOKS SEEM TO SAY.

```
10 CLS
20 INPUT"ENTER A SEED NUMBER, BETWEEN 1 & 65535";SD
30 INPUT"ENTER LENGTH OF RUN";C
40 INPUT"ENTER THRESHOLD FOR WIN";V
80 WG=0:LG=0
90 CLS
100 GOSUB 10000
110 GOSUB 10100
9999 END
10000 '***** SETUP RANDOM NUMBER SEED *****
10020 MS=INT(SD/256)
10030 LS=SD-MS*256
10040 POKE 16554,0
10050 POKE 16556,MS
10060 POKE 16555,LS
10099 RETURN
10100 '***** CALCULATIONS *****
10110 FOR X=1 TO C
10120 Z=RND(100)
10130 IF Z>V THEN WG=WG+1 ELSE LG=LG+1
10135 IF WG=0 GOTO 10160
10140 PW=INT((WG/X)*100)
10150
'CW=INT((WG/(WG+LG))*100)
10160 PRINT@256,"NUMBER OF GAMES PLAYED";X
10170 PRINT@320,"NUMBER OF GAMES WON";WG
10180 PRINT@384,"NUMBER OF GAMES LOST";LG
10190 PRINT@512,"PERCENTAGE GAMES WON";PW;"%"
10200 'PRINT@576,"CHECK VALUE";CW;"%"
10210 'PRINT@640,"RANDOM NUMBER";Z
10290 NEXT X
10299 RETURN
```

BREAKING THE PROGRAM DOWN INTO SECTIONS, LINES 10 - 9999 SETUP THE INITIAL CONDITIONS, LINES 10100 - 10099 INITIALISE THE RANDOM NUMBER GENERATOR, & LINES 10100 - 10299 PERFORM THE ACTUAL CALCULATIONS. BEFORE I PROCEED TO DESCRIBE EACH SECTION IN DETAIL, I SHOULD EXPLAIN SOMETHING ABOUT MY PROGRAMMING STYLE. I'VE ALWAYS PLACED STATEMENTS SUCH AS END & RETURN ON LINES ENDING IN 9 (E.G 9999, 10099 ETC.) AS THIS ALLOWS SUBROUTINES TO START AT EASILY REMEMBERED LINES. BELIEVE ME, WHEN YOU'RE TRYING TO DEBUG A BASIC ACCOUNTING PROGRAM THAT IS 36K LONG, IT MAKES THINGS MUCH EASIER.

THE FIRST PART OF THE PROGRAM IS SHOWN BELOW.

```
10 CLS
20 INPUT"ENTER A SEED NUMBER, BETWEEN 1 & 65535";SD
30 INPUT"ENTER LENGTH OF RUN";C
40 INPUT"ENTER THRESHOLD FOR WIN";V
80 WG=0:LG=0
90 CLS
100 GOSUB 10000
110 GOSUB 10100
9999 END
```

GOING THROUGH THE FUNCTION OF EACH LINE,

10 CLEAR THE SCREEN
20 ENTER THE SEED NUMBER
30 ENTER THE NUMBER OF TIMES TO PERFORM THE CALCULATION
40 ENTER THE BIAS FOR THE RANDOM NUMBER TO BE CONSIDERED A WIN
80 CLEAR THE WIN-LOOSE VARIABLES

90 CLEAR THE SCREEN
100 GOTO SUBROUTINE TO SEED RANDOM NUMBER GENERATOR
110 GOTO SUBROUTINE TO PERFORM CALCULATIONS
9999 END OF PROGRAM

MOST OF THE LINES ARE SELF EXPLANATORY. I'LL DISCUSS THE SIGNIFICANCE OF THE SEED NUMBER IN THE NEXT SECTION. LINE 30 SIMPLY SETS THE NUMBER OF NUMBERS TO GENERATE. LINE 40 SETS THE LEVEL AT WHICH THE DECISION AS TO WHETHER THE CHOICE IS A WIN OR NOT IS MADE. AS AN EXAMPLE LETS SAY WE MAKE THE DECISION THAT THERE IS A 60% CHANCE OF WINNING, THEN WE WOULD ENTER A VALUE OF 40 HERE. OR 30 IF THERE WAS A 70% CHANCE OF WINNING.



MOVING ON, THE NEXT SECTION OF THE PROGRAM SEEDS THE RANDOM NUMBER GENERATOR READY FOR THE RUN. THE BASIC MANUAL SAYS THAT THIS IS AN INTERNAL FUNCTION & IS NOT AVAILABLE TO THE USER. HOWEVER IT CAN BE PROGRAMMED.

```
10000 '***** SETUP RANDOM NUMBER SEED *****
10020 MS=INT(SD/256)
10030 LS=SD-MS*256
10040 POKE 16554,0
10050 POKE 16556,MS
10060 POKE 16555,LS
10099 RETURN
```

AGAIN, GOING THROUGH THE FUNCTIONS OF EACH LINE,

10020 GENERATES THE MOST SIGNIFICANT BYTE (MSB) OF THE SEED VALUE
10030 GENERATES THE LEAST SIGNIFICANT BYTE (LSB) OF THE SEED VALUE
10040 SETS MEMORY LOCATION 16554 TO ZERO
10050 LOADS MEMORY LOCATION 16556 WITH THE MSB
10060 LOADS MEMORY LOCATION 16555 WITH THE LSB
10099 RETURNS TO THE NEXT LINE AFTER THE LINE THAT CALLED THE SUBROUTINE

AGAIN, THE PROGRAM LINES ARE REASONABLY SELF EXPLANATORY. IF YOU LOOK BACK AT THE PROGRAM LISTING, YOU CAN SEE THAT THE SEED NUMBER IS A NUMBER BETWEEN 1 & 65535, MAKING IT A 16 BIT NUMBER. LINES 10020 & 10030 TURN THIS INTO THE 8 BIT MS(B) & LS(B) VALUES TO BE STORED. A FEW MINUTES WORK WITH A CALCULATOR WILL SHOW HOW THESE TWO LINES WORK.

THE MEMORY LOCATIONS THAT THE SEED NUMBER IS POKED INTO, ARE THE LOCATIONS THAT THE RANDOM NUMBER GENERATOR SUBROUTINE STORES THE GENERATED NUMBER IN WHEN IT EXITS, & IS USED AS THE STARTING VALUE THE NEXT TIME A RANDOM NUMBER IS GENERATED. I DON'T INTEND TO EXPLAIN HOW THE SUBROUTINE ACTUALLY WORKS IN THIS ARTICLE AS IT ISN'T NECESSARY.

WE NOW GET TO THE PART OF THE PROGRAM THAT ACTUALLY DOES THE WORK.

```
10100 '***** CALCULATIONS *****
10110 FOR X=1 TO C
10120 Z=RND(100)
10130 IF Z>V THEN WG=WG+1 ELSE LG=LG+1
10135 IF WG=0 GOTO 10160
10140 PW=INT((WG/X)*100)
10150 'CW=INT((WG/(WG+LG))*100)
10160 PRINT@256,"NUMBER OF GAMES PLAYED";X
```

```
10170 PRINT@320,"NUMBER OF
GAMES WON";WG
10180 PRINT@384,"NUMBER OF
GAMES LOST";LG
10190 PRINT@512,"PERCENTAGE
GAMES WON";PW;"%"
10200 'PRINT@576,"CHECK
VALUE";CW;"%"
10210 'PRINT@640,"RANDOM
NUMBER";Z
10290 NEXT X
10299 RETURN
```

AS BEFORE, THE FUNCTIONS OF EACH LINE IS SHOWN BELOW.

10110 START OF THE LOOP TO PERFORM THE CALCULATIONS, DEPENDING ON THE VALUE OF C
10120 Z GETS A RANDOM NUMBER BETWEEN 1 & 100
10130 IF Z IS GREATER THAN THE WIN THRESHOLD, INCREMENT THE WIN COUNTER (WG), ELSE INCREMENT THE LOST COUNTER (LG)

10135 A LINE SIMPLY TO DEAL WITH THE POSSIBILITY THAT THE FIRST NUMBER IS A LOOSE, TO STOP THE NEXT LINE DELIVERING A 'DIVIDE BY ZERO' ERROR

10140 CALCULATE THE PERCENTAGE WIN RATE
10150 'THIS LINE WAS SIMPLY TO CHECK THAT THE PROGRAM WAS WORKING CORRECTLY, BY CALCULATING THE ABOVE LINE IN A DIFFERENT WAY

10160 PRINT THE NUMBER OF GAMES PLAYED (THE VALUE OF X)
10170 PRINT THE NUMBER OF GAMES WON,
10180 PRINT THE NUMBER OF GAMES LOST
10190 PRINT THE PERCENTAGE OF GAMES WON
10200 'DIAGNOSTIC VALUE, (NOT NEEDED NORMALLY)
10210 'DIAGNOSTIC VALUE, (NOT NEEDED NORMALLY)
10290 LOOP AGAIN
10299 RETURN TO MAIN PROGRAM

IT SHOULD BE FAIRLY OBVIOUS WHAT EACH LINE DOES, HOWEVER A FEW COMMENTS ON VARIOUS DECISIONS I MADE SHOULD EXPLAIN THINGS.

IN LINE 10120 THE VALUE OF 100 USED AS THE ARGUMENT OF THE RND STATEMENT, GENERATE A NUMBER BETWEEN 1% & 100%. THIS SIMPLIFIES THE CHOICE OF VALUE FOR THE WIN THRESHOLD. HOWEVER, CHANGING THIS TO 6 WOULD ALLOW FOR DICE SIMULATIONS. LINE 10135 IS INCLUDED AS THERE IS A POSSIBILITY THAT THE FIRST RUN THROUGH WILL PRODUCE A LOOSE, & THEN LINE 10140 WILL SULK & PRODUCE A 'DIVIDE BY ZERO' ERROR. THE REASON THAT IT IS NUMBERED 10135IS THAT I ADDED IT AFTER LINE 10140 HAD PRODUCED A 'DIVIDE BY ZERO' ERROR!

LINE 10140 CALCULATES THE WIN PERCENTAGE, & SHOULD BE EASY TO SORT OUT. LINE 10150 DOES THE SAME THING AS LINE 10140, BUT IN A DIFFERENT WAY. LINE 10140 CALCULATES THE WIN BY DIVIDING THE NUMBER OF GAMES WON BY THE TOTAL NUMBER OF GAMES, LINE 10150 DIVIDES THE NUMBER OF GAMES WON BY THE TOTAL OF THE NUMBER OF GAMES WON PLUS THE NUMBER OF GAMES LOST. USING THE INT FUNCTION IN BOTH OF THESE LINES, IS SIMPLY BECAUSE FREECELL ONLY USES INTEGERS IN ITS PERCENTAGES.

LINES 10160 - 10210 SIMPLY PRINT OUT THE VARIOUS VALUES ON THE SCREEN. THESE USE PRINT@ FOR EASE OF READING. YOU COULD, IF YOU WISHED, CHANGE THE PRINTING SO THAT THE MESSAGES ARE PRINTED FIRST, & ONLY THE VALUES ARE PRINTED ON THE RUN THROUGH, THIS AGAIN, IS LEFT AS AN

'EXERCISE FOR THE READER'. LINES 10200 PRINTS OUT THE VALUE WORKED OUT 10150 TO ALLOW FOR DIAGNOSTIC CHECKING, & LINE 10210 PRINTED THE RANDOM NUMBER, JUST TO SEE IF IT IS RANDOM, AGAIN FOR DIAGNOSTIC CHECKING. THE PROGRAM DOESN'T RUN SO FAST, THAT THESE LINES ARE RENDERED USELESS.

WELL, I'M AT THE END OF THIS ARTICLE ON REPEATABLE RANDOMNESS. I HOPE SOME OF THE IDEAS PRESENTED IN IT WILL BE USEFUL TO YOU. IT SHOULD BE REMEMBERED THAT THE RND COMMAND PRODUCES PSEUDO RANDOM NUMBERS, NOT TRUE RANDOM NUMBERS. HOWEVER FOR THE VAST MAJORITY OF APPLICATIONS, THESE WILL SUITABLE.

HAVE FUN.

--== oØo ==--

(CONTINUED FROM PAGE 2)

OF THE ZORLOF MANUAL WHICH I'VE INCLUDED, JUST IN CASE YOU FANCY A LITTLE PLAY TOO.

THERE IS ALSO AN EXAMPLE OF POKEING CHANGES TO A BASIC PROGRAM. THIS ALL STARTED WHEN I WAS ENTERING DATA STATEMENTS AND I WAS FOREVER FUMBLING ABOUT, TRYING TO LOCATE THE COMMA KEY!

PETER STONE HAS SENT IN A VERY INTERESTING ARTICLE ON REPEATABLE RANDOMNESS. THE RND FUNCTION TENDS TO BE ASSOCIATED ONLY WITH GAMES, BUT, AS PETER IMPLIES, THERE IS A MUCH MORE SERIOUS SIDE TO IT. I REMEMBER TRYING TO SET UP AN EXPERIMENT WITH A 24 HOUR RUN OF A SIMULATED COIN TOSSING PROGRAM. THE BIGGEST PROBLEM, AS I RECALL, WAS WITH MAIN-SPIKES!!

A COLLEAGUE FROM THE BRITISH PRINTING SOCIETY, (BPSNET.ORG.UK), LES TAYLOR, HAS SUPPLIED US WITH AN ARTICLE ON STEGANOGROPHY. THIS IS NOT DEDICATED TO TRS-80'S IN PARTICULAR, BUT I FELT, AFTER CHRIS FARA'S ARTICLE ON SECRET CODES IN THE PREVIOUS ISSUE, IT REALLY WAS MOST INTERESTING.

PETER STONE HAS INTRODUCED A NEW OCCASIONAL SERIES WHICH HE HAS ENTITLED 'BITS, BYTES & NIBBLES'. HE WOULD BE DELIGHTED IF YOU WOULD CONTRIBUTE ANY LITTLE BITS OF TANDY TRIVIA YOU HAVE COME ACROSS OVER THE YEARS.

THERE'S A GREAT TRS-80 FORUM I CAME ACROSS RECENTLY, SO JUST IN CASE IT'S NEW TO YOU TOO, HERE'S THE WEB ADDRESS-

HTTP://WWW.VINTAGE-COMPUTER.COM/VCFORUM/FORUMDISPLAY.PHP?46-TANDY-RADIO-SHACK

IT'S A BIT OF A MOUTHFUL SO I'VE PUT A LINK ON THE WEBSITE FOR US ALL TO USE.

AND FINALLY, JUST IN CASE YOU HAVEN'T NOTICED, THE WEB SITE CLOCKED UP 5000 HITS RECENTLY. THANK YOU ALL, FOR YOUR SUPPORT, ENCOURAGEMENT AND INTEREST.
LONG MAY WE ALL CONTINUE.



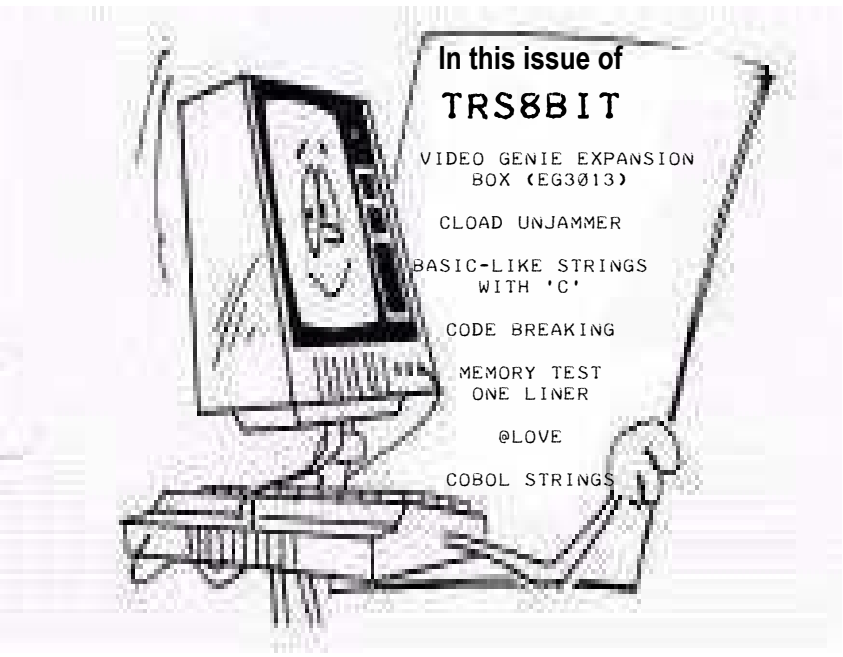
BYE FOR NOW,

DUSTY
(16K RULES)

TRS8BIT



SEPTEMBER ROLLS IN AND IT'S TIME FOR YET ANOTHER EDITION OF TRS8BIT. I USUALLY FIND THAT THIS IS ONE OF THE MOST ENJOYABLE TIMES OF THE YEAR. THE MAD SUMMER RUSH TO THE COAST HAS FINISHED AND MOST OF THE HOLIDAY MAKERS HAVE GONE HOME. WE LOCALS CAN NOW CATCH OUR BREATHS AND THIS YEAR, WE HAVE THE HIGHEST TIDE OF THE YEAR TO LOOK FORWARD TO TOO! THE TIDES CAN BECOME QUITE SPECTACULAR HERE, ON THE LINCOLNSHIRE COAST, ESPECIALLY WITH THE WEATHER, AND PARTICULARLY THE WIND ASSISTING IN LIVENING THINGS UP.
TALKING OF TIDES, DID YOU KNOW THAT TIDE TIMES AND DETAILS OF HEIGHTS ETC. ARE REGARDED AS CLASSIFIED INFORMATION BY THE POWERS THAT BE? IT'S A CRIMINAL OFFENCE TO REPRODUCE THEM WITHOUT PERMISSION AND PAYMENT OF A SUITABLE FEE. I THINK THIS STEMS BACK TO THE FIRST WORLD WAR AND 'DORA', THE DEFENCE OF THE REALM ACT.



HAS ANYONE A BASIC PROGRAM FOR THE MODEL 1 WHICH CALCULATES THEM? WHAT A WONDERFUL THOUGHT IT WOULD BE, ALSO BEING QUITE AN ACHIEVEMENT, TO BECOME 'SUBVERSIVE' ON A MODEL 1 AFTER ALL THESE YEARS EH?

UNFORTUNATELY, ALL INVESTIGATORY WORK ON THE ACULAB FLOPPY TAPE HAS GROUND TO A HALT. IT WASN'T HELPED BY ME, SENDING PETER STONE A CORRUPT FILE OF THE ROM I'D MANAGED TO EXTRACT. HOWEVER, COME THE DARKER NIGHTS AND WITH THEM A BIT MORE SPARE TIME, I'M HOPING FOR GREAT THINGS. I UNDERSTAND FROM PETER HE HAS CONTACTED MATTHEW REED REGARDING THE ACULAB AND HE'S QUITE HOPEFUL THINGS CAN BE SORTED OUT.

THE RESULTS OF THE M100 CLUB COMPETITION ARE POSTED ON THE M100 CLUB WEBSITE. THE QUALITY OF PROGRAMS ENTERED WAS, I THOUGHT, EXCEPTIONALLY HIGH. I'VE LISTED THE TOP 3 WINNERS ON PAGE 12, WITH JUST A SHORT DESCRIPTION OF EACH ENTRY. CHECK OUT THE WEBSITE THOUGH, IT'S REALLY INTRIGUING AS TO WHAT THE M100 HAS BEEN ABLE TO ACHIEVE WITH THESE PROGRAMS.

EBAY HAS BEEN RATHER QUIET OVER THE LAST 3 MONTHS ON THE TANDY FRONT. THERE'S THE USUAL £90 'BUT-IT-NOW' CO-CO'S, A COUPLE OF MODEL 1'S AND ANOTHER MODEL 3 DISK UPGRADE SYSTEM BUT NOT MUCH ELSE IN THE HARDWARE LINE TO AROUSE

INTEREST. I HAVE MANAGED TO 'WIN' HOWEVER, A COUPLE OF ISSUES OF PCW'S (PERSONAL COMPUTER WORLD) FROM 1981. IF ANYONE WOULD LIKE TO BORROW THEM, PLEASE CONTACT ME AND I'LL FIND OUT THE POSTAGE COSTS ETC.

THERE HAS BEEN VARIOUS SOFTWARE UPGRADES AVAILABLE FOR TRS-80 EMULATORS AND ACCESSORIES. MIGUEL DUTRA'S VIRTUAL DISK MANAGER IS UP TO RELEASE V0.7. MATTHEW REEDS TRS32 EMULATOR IS UP TO V1.24
IRA'S WEB SITE IS NOW OFFERING, WITH PERMISSION, VARIOUS DOWNLOADS FROM BIGFIVE, LAZYWRITER, DENNIS BATHORY-KITZS, WESTMORELAND AND GILMAN AND VERNON B HESTER. A VERY GENEROUS GESTURE BY ALL THE AUTHORS. I BELIEVE IT WAS INSPIRED BY A RECENT BIRTHDAY CELEBRATION OF THE MODEL 1 IN THE USA.

ANYWAY, DOWN TO THIS ISSUE. I'VE, AS USUAL, TRIED MY BEST TO FIND ARTICLES FOR ALL INTERESTS AND TASTES, BUT FEEDBACK WOULD BE APPRECIATED, IF YOU THINK I'M ON THE WRONG (OR RIGHT) TRACK OF COURSE.
THERE'S AN ARTICLE ON THE VIDEO GENIE EXPANSION BOX AND THE SUBSEQUENT PRINT PROBLEMS THAT CAN OCCUR.

A REALLY NEAT IDEA TO HELP WHEN THE COMPUTER LOCKS OUT WHEN YOU'RE CLOAD'ING.

2 ARTICLES ON STRING HANDLING IN A NON BASIC ENVIRONMENT, WHICH I HOPE 'C' AND COBOL AFICIONADO'S WILL ENJOY.

I'VE ALSO INCLUDED A

(CONTINUED ON PAGE 12)

AT THE
READY>
PROMPT



I CAME ACROSS THIS HANDY FORMULAE FOR CONVERTING DECIMAL VALUES TO THEIR MSB/LSB NUMBERS.
BY WAY OF AN EXAMPLE; IF X=17155 THEN LS=3 AND MS=67 CAN BE FOUND BY-
X=VARPTR(A):LS=XAND255:MS=(XAND-256)/256

VERY EARLY EXPANSION INTERFACES (TWISTED PAIR MOD TOGETHER WITH THE BUFFERED CABLE TYPE), COULD HAD A LOT OF PROBLEMS TRYING TO RUN DOUBLE SIDED DRIVES ON IT. I REMEMBER THAT ONE OF THE PROBLEMS WAS EVENTUALLY TRACKED DOWN TO THE FACT THAT TWO OF THE CONNECTORS ON THE DISK BUS WERE JOINED. IF I REMEMBER CORRECTLY, IS WAS 32 AND 34. IF YOU ARE HAVING THIS SORT OF PROBLEM, IT'S WORTH CHECKING OUT THIS POINT.

HERE'S A COUPLE OF LITTLE BITS OF PRINTER RELATED INTEREST. THERMAL PAPER, SIMILAR TO THAT USED ON THE QUICKPRINTER 2, IS STILL AVAILABLE FOR THE SINCLAIR SPECTRUM. IT'S 110MM WIDE AND THE DIAMETER OF THE REELS ARE 45MM.
ALSO, RIBBONS FOR THE DMP100 (TANDY LINEPRINTER VII) IS ALSO IDENTICAL WITH THE ONES USED ON THE COMMODORE VIC 1520/1525P, THE GORILLA BANNANA AND THE SEIKOSHA GP80A, GP100A AND THE GP250X.
DOES ANYONE KNOW WHERE (OR IF) THESE RIBBONS CAN STILL BE FOUND IN THE UK?

--== O0O ==--

PORT FDH AND ADDRESS 37EBH FOR THE VIDEO GENIE EG3Ø13 EXPANSION BOX.

KEN GREY (WITH THANKS TO MIKE MORRIS)

OWNERS OF THE VIDEO GENIE EG3Ø13 EXPANSION BOX HAVE TO LIVE WITH MODIFYING TRS-8Ø SOFTWARE WHICH ADDRESSES THE PRINTER AT 37EBH WHEN THEIR SYSTEM PORTS THE PRINTER AT FDH. A VERY SIMPLE HARDWARE MODIFICATION (2 DIODES) WILL ALLOW THE USE OF SOFTWARE WHICH USES EITHER ADDRESS OR PORT. NO MORE SUPERZAP OR PROZAP, SEARCHING FOR 32 EB 37 OR 3A EB 37 TO CHANGE THEM INTO ØØ D3 FD AND ØØ DB FD. WHICHEVER SYSTEM THE SOFTWARE USES THIS LITTLE MOD WILL ACCEPT.

IT HAS SAVED ME A TREMENDOUS AMOUNT OF WORK AS BEING A USER OF DATA-BASES AND WORD PROCESSORS, I USE THE PRINTER MORE THAN MOST PEOPLE.
THE FIRST JOB IS TO GET INSIDE THE EXPANSION BOX. THIS IS DONE BY REMOVING THE 5 SCREWS FROM THE BOTTOM. GENTLY REMOVE THE TOP COVER TO REVEAL THE PCB. TURN THE BOC SO THE EDGE CONNECTORS ARE FACING YOU. FROM THE CENTRE OF THE CONNECTOR THAT GOES TO THE COMPUTER, GOING TOWARDS THE BACK OF THE PCB, IS A ROW OF IC'S, THE ONE NEAREST BEING A 74LS156 AND THE ONE BEHIND A 74LS155. THESE ARE Z33 AND Z34 RESPECTIVELY. OWNERS OF THE SERVICE MANUAL SHOULD HAVE NO DIFFICULTY IN LOCATING THESE COMPONENTS.

WHAT IS REQUIRED NOW IS (1) A FINE SOLDERING IRON, (2) 2 IN1418DIODES AND (3) A STEADY HAND! JOIN THE ANODE ENDS OF THE 2 DIODES TOGETHER AND CONNECT TO PIN 6 OF Z33 (THE 74LS156). TAKE ONE OF THE DIODE CATHODES TO PIN 5 OF Z34 AND THE OTHER TO PIN 11 OF Z34 (THE 74LS155). THAT'S IT! YOU HAVE COMPLETED THE NECESSARY MODIFICATIONS. ENSURE THAT SWITCHES 6 & 7 OF DP1 ARE SET TO OFF. (DP1 IS THE SWITCH WITH THE RED SPOT). BEFORE PUTTING THE COVER BACK ON, WITH EVERYTHING CONNECTED, FROM DOS BASIC TYPE THE FOLLOWING:-

```
FOR X=1 TO 25: POKE &H37EB,13: OUT255,13: NEXT
```

YOU SHOULD GET 5Ø CARRIAGE RETURNS, 25 FROM 37EBH AND 25 FROM PORT FDH. BOX IT ALL BACK UP, FORGET ALL ABOUT ANY SOFTWARE MODS FROM NOW ON AND SLEEP HAPPY AT NIGHT!

GOOD PRINTING.



CLOAD JAMS

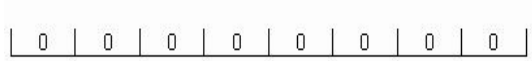
BY JOHN CARDIFF

THIS ARTICLE FIRST APPEARED IN NATGUG NEWS

IT IS VERY ANNOYING WHEN YOU ARE LEFT STARING AT TWO ** AT THE TOP RIGHT HAND CORNER OF THE SCREEN AND THE KEYBOARD LOCKED OUT AFTER TRYING TO 'CLOAD' A PROGRAM FROM CASSETTE WITH THE VOLUME TOO HIGH OR TOO LOW.

THE 'CLOAD' ROUTINE STARTS BY FINDING A SYNC BYTE 'A5'HEX FROM THE TAPE, PUTS THE '**' ON THE SCREEN, AND THEN LOOKS FOR 3 'D3'HEX BYTES IN A ROW WHICH ARE AT THE BEGINNING OF ALL BASIC PROGRAMS. IF THE VOLUME ISN'T SET CORRECTLY, THE DATA FROM THE TAPES GETS OUT OF SYNC AND YOU ARE STUCK IN THE ROM WHILE IT GOES ROUND IN A LOOP LOOKING FOR THREE SUCCESSIVE 'D3'HEX'S.

THE WAY THE DATA IS READ FROM TAPE IS IN BITS. 8 BITS MAKING UP ONE BYTE. D3H FOR INSTANCE. EVERY BIT HAS A MARKER IN FRONT OF IT, SO THE FORMAT ON THE TAPE FOR ONE BYTE ØØH LOOKS



LIKE THIS -
THE SPIKES ARE MARKERS AND THE GAP BETWEEN EACH IS WHERE THE DATA BIT IS. IN THIS CASE, EIGHT ZERO BITS. Ø3H IN BIT FORM IS ØØØØ ØØ11 THEREFORE THE TAPE FORMAT WOULD LOOK LIKE



THE SPIKE THAT MAKES THE MARKER IS ALSO USED TO MAKE A 1 BIT BETWEEN THE MARKERS. D3H IN BIT FORM IS 11Ø1 ØØ11 THUS ON TAPE



HOWEVER, IF THE VOLUME IS WRONG, YOU MIGHT READ TOO MANY SPIKES OR MISS SOME SPIKES ALTOGETHER. THE MACHINE SOON CAN'T TELL THE DIFFERENCE BETWEEN A MARKER SPIKE OR A DATA SPIKE AND GETS OUT OF SYNC.

ON THE NEXT PAGE IS A SHORT MACHINE LANGUAGE PROGRAM TO PUT ON TAPE SO THAT WHEN YOU PLAY IT YOU WILL GET BACK TO THE READY> PROMPT, WITHOUT HAVING TO PRESS THE RESET BUTTON. IT WORKS BY SENDING THREE D3H'S AND SOME ZEROS (SIGNALLING THE END OF THE PROGRAM) WITH THE ODD BIT IN BETWEEN TO GET BACK INTO SYNC. IT DOES THIS EIGHT TIMES WHICH SHOULD BE ENOUGH BUT IF YOU WANT A LONGER TAPE (WHICH CAN START ANYWHERE) THEN LOAD C WITH A HIGHER NUMBER. (I.E. CHANGE LINE 1ØØ).

JUST IN CASE YOU MIGHT WISH TO MAKE A BASIC PROGRAM AS APPOSED TO A SYSTEM ONE, I'VE ALSO INCLUDED A SMALL BASIC PROGRAM FOR YOU TO USE. DON'T FORGET THE CHANGE FOR DISK BASIC USER CALLS.



```
; CLOAD UNJAMMER
; JOHN CARDIFF
;

 1                LENGTH:   EQU  8H
 2                TAPEON:   EQU  212H
 3                TAPOFF:   EQU  1F8H
 4                WRITE:    EQU  264H
 5                SPIKE:    EQU  1D9H
 6                          ORG  7000H
 7 7000 3E01      START:    LD   A,1
 8 7002 CD1202              CALL TAPEON     ;START RECORDER
 9 7005 0E08                LD   C,LENGTH
10 7007 0620      CONT:     LD   B,20H      ;PLENTY OF ZEROS
11 7009 AF                 XOR  A          ;ZERO A
12 700A CD6402    AGAIN1:   CALL WRITE
13 700D 10FB               DJNZ AGAIN1
14 700F 0603               LD   B,3
15 7011 3ED3               LD   A,0D3H     ;'BASIC' IDENTITY BYTE
16 7013 CD6402    AGAIN2:   CALL WRITE
17 7016 10FB               DJNZ AGAIN2
18 7018 CDD901             CALL SPIKE      ;WRITE A MARKER
19 701B 0687               LD   B,87H
20 701D 10FE      DELAY:    DJNZ DELAY
21 701F 0D                 DEC  C
22 7020 20E5               JR   NZ,CONT    ;LOOP 'LENGTH' TIMES
23 7022 CDF801             CALL TAPOFF     ;STOP RECORDER
24 7025 C9                 RET
25                         END
```

```
10 REM * CLOAD UNJAMMER  *
20 REM *  JOHN CARDIFF   *
30 DEFINT A-Z:CLS
40 PRINT"PRESS ANY KEY WHEN READY TO RECORD"
50 FOR I=28672 TO 28705:READ A:POKE I,A:NEXT
60 I$=INKEY$:IF I$="" THEN 60
70 A!=28672/256:A=A!:B=(A!-A)*256
80 POKE 16526,B:POKE 16527,A:A=USR(0)
90 DATA 62,1,205,18,2,14,8,6,32,205,100,2,16,251
100 DATA 6,3,62,211,205,100,2,16,251,205,217,1,6
110 DATA 135,16,254,13,32,230,201

N.B. FOR DISK BASIC CHANGE LINE 80 TO -

80 DEFUSR1=&H7000:A=USR1(0)
```

--== o0o ==--

```
/*
**  BASIC-LIKE STRING OPERATIONS
**     FOR THE PUBLIC DOMAIN
**          BOB STOUT
*/

#INCLUDE <STDIO.H>
#INCLUDE <STRING.H>
#INCLUDE <STDARG.H>
#INCLUDE <LIMITS.H>
#INCLUDE <ASSERT.H>
#INCLUDE "SNIPTYPE.H"
#INCLUDE "BASTRNGS.H"

STATIC INT STRALLOC_PTR;
STATIC CHAR *STRINGS[8];
STATIC INT STR_TAG[8];

/*
**  STRALLOC() IS THE KEY FUNCTION IN THIS PACKAGE,
    MAINTAINING A POOL OF
**  REUSABLE STRINGS.
*/

CHAR *STRALLOC(SIZE_T LENGTH)

      REGISTER INT I;

      IF (UINT_MAX == LENGTH)          /* ASSUME SIZE_T ==
                                          UNSIGNED INT    */
            RETURN NULL;

      I = STRALLOC_PTR++;
      ++LENGTH;                        /* ALLOW FOR
                                          TERMINATING NUL */

      IF ((!STRINGS[I])  (LENGTH > STRLEN(STRINGS[I])))

            STRINGS[I] = (CHAR *)REALLOC(STRINGS[I],
LENGTH);
            ASSERT(NULL != STRINGS[I]);
            STR_TAG[I] = -1;

      ELSE  STR_TAG[I] = 0;
      STRALLOC_PTR &= 7;
      RETURN (STRINGS[I]);
      /* MAINTAINS 8 STRINGS IN A CIRCULAR BUFFER */

/*
**  FREE THE STRING POOL.
*/

VOID STR_FREE(CHAR *STRING)

      REGISTER INT I;
```

```
        FOR (I = Ø; I < 8; ++I)
                IF (STRINGS[I] == STRING)
                        IF (STR_TAG[I])
                                FREE(STRINGS[I]);
                        RETURN;

/*
**  RETURN THE LEFTMOST N CHARACTERS FROM A STRING,
    EQUIVALENT TO LEFT$
*/

CHAR *LEFT(CHAR *STRING, SIZE_T N)

        CHAR *BUF;
        SIZE_T STRLENGTH = STRLEN(STRING);

        IF (N > STRLENGTH)
                N = STRLENGTH;
        BUF = STRALLOC(N);
        MEMCPY(BUF, STRING, N);
        BUF[N] = NUL;
        RETURN BUF;


/*
**  RETURN THE RIGHTMOST N CHARACTERS FROM A STRING,
    EQUIVALENT TO RIGHT$
*/

CHAR *RIGHT(CHAR *STRING, SIZE_T N)

        CHAR *BUF;
        SIZE_T STRLENGTH = STRLEN(STRING);

        IF (N > STRLENGTH)
                N = STRLENGTH;
        BUF = STRALLOC(N);
        STRCPY(BUF, &STRING[STRLENGTH-N]);
        RETURN BUF;


/*
**  RETURN A SUBSTRING, N CHARACTERS LONG BEGINNING AT
    POSITION M,
**  EQUIVALENT TO MID$
*/

CHAR *MID(CHAR *STRING, SIZE_T M, SIZE_T N)

        CHAR *BUF;
        SIZE_T STRLENGTH = STRLEN(STRING);

        IF (M > STRLENGTH)
                RETURN NULL;
        IF (N > (STRLENGTH - M))
                N = STRLENGTH - M;
```

```
        BUF = STRALLOC(N);
        MEMCPY(BUF, &STRING[M-1], N);
        BUF[N] = NUL;
        RETURN BUF;

/*
**  STRING CONCATENATION FUNCTION, EQUIVALENT TO
A$=B$+C$+...
*/

CHAR *STRING_ADD(CHAR *STRING, ...)

        VA_LIST ARG_PTR;
        CHAR *TEMP1, *TEMP2, *BUF;

        VA_START(ARG_PTR, STRING);
        TEMP1 = STRING;
        DO

              IF(NULL == (TEMP2 = VA_ARG(ARG_PTR, CHAR *)))
                    BREAK;
              BUF = STRALLOC(STRLEN(TEMP1) + STRLEN(TEMP2));
              TEMP1 = STRCAT(STRCPY(BUF, TEMP1), TEMP2);
         WHILE (NULL != TEMP2);
        RETURN TEMP1;


/*
**  CREATE A STRING OF REPEATED CHARACTERS, EQUIVALENT TO
STRING$()
*/

CHAR *STRING(INT CH, SIZE_T N)

        CHAR *BUF;

        IF (N)
              BUF = STRALLOC(N);
              MEMSET(BUF, CH, N);
         BUF[N] = NUL;
        RETURN BUF;



/*
**  BASIC INSTR$() WORK ALIKE - RETURNS POSITION (1-BASED)
**  OF FINDSTR IN STRING, STARTING SEARCH AT POSITION
**  START_POS (ALSO 1-BASED).
**
**  FUNCTION SUGGESTED BY TIKA CARR
*/

UNSIGNED INT INSTR(CONST UNSIGNED INT START_POS,
                   CONST CHAR         *STRING,
                   CONST CHAR         *FINDSTR)
        CHAR *P;
```

```
        UNSIGNED INT POS;

        IF (START_POS > STRLEN(STRING))
              RETURN Ø;
        ELSE  POS = START_POS - 1;     /* BASIC OFFSETS ARE
                                          ONE-BASED, NOT
                                          ZERO-BASED AS IN C
*/

        IF (NULL != (P = STRSTR(STRING + POS, FINDSTR)))
              RETURN (INT)(P - (CHAR *)STRING) + 1;
/*      POSITION Ø = POSITION 1 */
        ELSE  RETURN Ø;


#IFDEF TEST

/*
**  DEMO MAIN()
*/

MAIN()

        CHAR *X = "EUROPEAN", *Y = "HARDWARE", *Z =
"SKATERS", *Q;
        CHAR *A = "THIS IS A TEST", *B = "IS" ,
             *C = "\NSEARCHING FOR \"%S\" IN \"%S\" STARTING
AT POSITION %D\N";
        UNSIGNED INT I, POS;

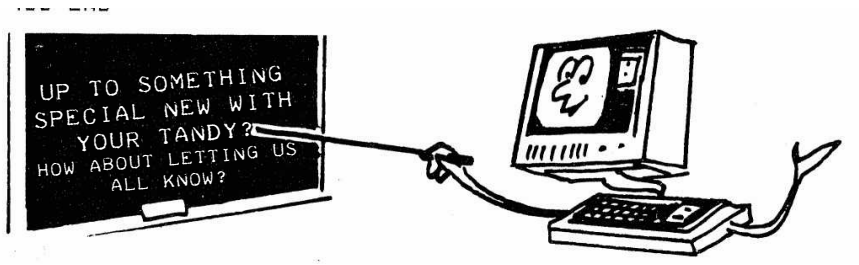
        Z = STRING_ADD(LEFT(X, 2), RIGHT(Y, 2), MID(Z, 2, 2),
"!", NULL);
        Q = STRING('!', 4);
        PRINTF("%S%S\N", Z, Q);

        FOR (I = 2; I < STRLEN(A); I+= 2)

              PRINTF(C, B, A, I);
              IF (Ø != (POS = INSTR(I, A, B)))
                    PRINTF("FOUND AT POSITION %D\N", POS);
              ELSE  PUTS("NOT FOUND");


        RETURN Ø;


#ENDIF /* TEST */
```



CODEBREAKING

IN THE 1ST HALF OF THE 19TH CENTURY, INTELLIGENCE ACTIVITY CONCENTRATED ON INVENTING CODES; IN THE 2ND HALF, IT BEGAN TO PAY AS MUCH ATTENTION TO BREAKING THEM. THIS SHIFT OF EMPHASIS WAS MAINLY DUE TO THE WORK OF A GERMAN OFFICER, MAJ. KASISKI. THE HISTORICAL REASON FOR THE LACK OF INTEREST IN CODE BREAKING WAS SIMPLY THAT NO ONE KNEW HOW TO DO IT!
THE VIGENERE CIPHER, WHICH WAS GETTING ON FOR 300 YEARS OLD. WAS STILL THOUGHT TO BE IMPREGNABLE. IT WAS KASISKI WHO SHOWED HOW TO BREAK IT, AND HIS METHOD HAS BECOME THE CORNER-STONE OF ALL MODERN ANALYSIS.

IT WAS ALREADY KNOWN THAT, IN EVERY LANGUAGE, CERTAIN LETTERS OCCUR MORE FREQUENTLY THAN OTHERS. IN ENGLISH, THE MOST COMMON LETTERS ARE 'ETAONRIS' (WHICH CAN BE EASILY REMEMBERED BY THE PHRASE - 'A SIN TO ER'. (IT'S A RECOGNISED FACT THAT 60 PER CENT OF THE ENGLISH LANGUAGE CAN BE WRITTEN USING JUST THESE EIGHT LETTERS!)

THE FREQUENCY TABLE OVER LEAF, SHOWS THE OCCURRENCE OF LETTERS IN ENGLISH AND IS TYPICAL OF THE TYPE OF TABLE USED BY ANALYSTS.

WHAT LASISKI SHOWED WAS THAT, IN ALL LANGUAGES, BESIDES SINGLE LETTERS APPEARING FREQUENTLY, CERTAIN COMBINATIONS OF LETTERS APPEAR FREQUENTLY AS WELL. IN ENGLISH, THE MOST FREQUENTLY RECURRING DOUBLE LETTERS (CALLED BIGRAMS) ARE 'TH' (168 TIMES PER 1000 WORDS), HE (132), AN (92), RE (91), ER (88), IN (86), ON (71), AT (68), AND ND (61). THE MOST FREQUENT TRIPPLE LETTERS (TRIGRAMS) ARE; CON, ENT, ERS, EVE, FOR, HER, TED, TER, THE, TIO AND VER. THESE DISCOVERIES LED TO OTHERS. FOR EXAMPLE, MORE THAN HALF OF ALL ENGLISH WORDS BEGIN WITH A,O,S,T OT W. MORE THAN HALF END IN D,E,S OR T. THE MOST FREQUENT ENDINGS OF BRITISH SURNAMES ARE SON, TON, ER, FORD, MAN, AND BY. THE MOST FREQUENTLY USED ENGLISH WORDS ARE THE, OF, AND AND TO. THE MOST REPEATED NOUN IS 'TODAY'. KASISKI ALSO POINTED OUT THAT IT WAS IMPOSSIBLE TO WRITE A LONG MESSAGE WITHOUT THESE REPETITIONS. RECOGNISING AND COUNTING THESE REPETITIONS IS HOW CODE IS EVENTUALLY BROKEN.

PRESENTED WITH A CODED MESSAGE, AND WITHOUT ANY IDEA OF HOW IT WAS ENCIPHERED, THE CRYPTANALTSIST CAN MAKE A NUMBER OF ASSUMPTIONS. IF THE LETTERS 'A SIN TO ER' OCCUR FREQUENTLY, THEY ARE DEALING WITH A TRANSPOSITION. THAT IS BECAUSE A TRANSPOSITION CIPHER ONLY REARRANGES THE LETTER ORDER. IT DOES NOT CHANGE THE LETTERS THEMSELVES, SO

THE MOST FREQUENTLY USED LETTERS AND WORDS IN A MESSAGE REMAIN THE SAME. IF, HOWEVER, THE 'A SIN TO ER' LETTERS OCCUR VERY RARELY, THEN THE CIPHER IS A SUBSTITUTION, AS THE OBJECT OF A SUBSTITUTION IS TO REPLACE THE LETTERS WITH OTHERS. SO IF THE CRYPTOGRAM HAS 'Z' AND 'Q' AS THE MOST FREQUENT LETTERS, IT'S OBVIOUS THAT THEY STAND FOR SOMETHING ELSE (PROBABLY 'E' OR 'T').

WITHOUT FREQUENCY TABLES, A SOLUTION IS ALMOST IMPOSSIBLE. THEY ARE SO IMPORTANT THAT THEY CAN HELP EXPERT CODEBREAKERS, BREAK CIPHERS IN ANY LANGUAGE - EVEN CHINESE AND EVEN WHEN THE ANALYST CANNOT UNDERSTAND CHINESE SYMBOLS!

Letters		*Frequency of occurrence in 1,000 words*	*Frequency of occurrence in 1,000 letters*
1	E	591	131.05
2	T	473	104.68
3	A	368	81.51
4	O	360	79.95
5	N	320	70.98
6	R	308	68.32
7	I	286	63.45
8	S	275	01.01
9	H	237	52.59
10	D	171	37.88
11	L	153	33.89
12	F	132	29.24
13	C	124	27.58
14	M	114	25.36
15	U	111	24.59
16	G	90	19.94
17	Y	89	19.82
18	P	89	19.82
19	W	68	15.39
20	B	65	14.40
21	V	41	9.19
22	K	19	4.20
23	X	7	1.66
24	J	6	1.32
25	Q	5	1.22
26	Z	3	0.77

M100 ANNIVERSARY COMPETITION

HERE IS A BRIEF REPORT OF THE COMPETITION WINNERS FROM RICK HANSON'S M100 SITE.

1ST PLACE WENT TO RON WIESEN WITH HIS 'SUN COMPASS'. SOFTWARE AND HARDWARE MODS, TURN A M100 INTO A SUN COMPASS.

2ND PLACE WENT TO ROGER SCHMIDT WITH HIS BITMAP VIEWER PROGRAM.

JOINT 3RD PLACES WENT TO CURTIS F KAYLOR WITH A PROGRAM CALL MATHS COMMAND, (WHICH IS ONLY 10 LINES LONG) AND FRED WHITAKER WITH HIS ARTERIAL BLOOD GAS INTERPRETATION PROGRAM.

THE COMPETITION WAS WELL SUPPORTED AND IT MUST HAVE BEEN DIFFICULT TO PICK A WINNER JUDGING BY THE QUALITY OF ALL THE ENTRANTS. THE TOP PRIZE WAS $100.

--== o0o ==--

(CONTINUED FROM PAGE 2)

REPRINT FROM AN EDITION OF 80-US JUST TO REMIND US ALL OF THE DIFFERENCES BETWEEN THE EXATRON STRINGY FLOPPY AND THE ACULAB FLOPPY TAPE. I MUST CONFESS, I HADN'T REALISED THAT MUCH OF THE EXATRON SOFTWARE WAS NOT HELD WITHIN THE ROM; (THE DATA I/O FOR EXAMPLE).

AND, OF COURSE, THERE HAS TO BE A ONE LINER EH! THIS IS A FOLLOW-ON FROM MY NUMBERS MEMORY TEST FROM THE LAST EDITION. MY ELDEST GRANDSON, 6, WAS FINDING REMEMBERING THE NUMBERS TOO EASY, SO, NOT ONLY DOES HE NOW HAVE TO REMEMBER LETTERS, IT WILL, AS AN ADDED BONUS, IMPROVE HIS KEYBOARD KNOWLEDGE. (HE'S SO GOOD AT REMEMBERING THOUGH, LORNA TAKES HIM WITH HER TO THE SUPERMARKET TO SAVE HAVING TO WRITING HER SHOPPING LIST DOWN!!)

PLEASE DON'T FORGET THAT THE NEXT ISSUE IS THE XMAS ONE. ANYTHING SPECIAL YOU'D LIKE TO SEE, PLEASE LET ME KNOW. UNTIL THEN

TAKE CARE

DUSTY
(16K RULES)



HERE'S A LITTLE ONE-LINER TO NOT ONLY TEST YOU MEMORY, BUT ALSO IT SHOULD IMPROVE YOU KEYBOARD SKILLS. DON'T FORGET, THE LARGE GAPS AFTER THE INITIAL INPUT AND PRINT STATE-MENTS ARE CREATED BY THE DOWN ARROW AND NOT THE SPACE BAR!

```
0 INPUT"

PRESS ENTER";Z$:CLS:A=RND(4)+3:FORB=1TOA:C=RND(26)+64:PRINTCHR$(
C);" ";:A$=A$+CHR$(C):NEXTB:FORD=1TOA*45:NEXTD:CLS:INPUT"PLEASE
ENTER LETTERS";X$:IFX$=A$THENPRINT"
WELL DONE, CORRECT":RUNELSEPRINT"
SORRY,THEY WERE ";A$:RUN
```

@Love



Problems and solutions with ESF

Model I Level II 16K plus Stringy Floppy

Stephen W. Cox, Cincinnati, OH

The purchase of a TRS-80 Model I in June of 1980 was the beginning of a love-hate relationship which balanced on a thin edge until the balance in my checkbook allowed the purchase of an Exatron Stringy Floppy in May of 1981. The Stringy Floppy (affectionately called ESF) definitely tipped the scale to the love side.

My purpose in writing this is to share with non-ESF owners as well as ESF users, some of the ESF's good points and bad points as I see them.

In my prior ESF days, one thing which caused TRS-80 hate was the need to apologize to friends for a four minute load. I could see the enthusiasm fade from their faces as I tried to explain why a computer should take so long. Shrinking four minutes to 25 seconds is an incredible task which the ESF does miraculously well.

It must be pointed out that if there are four 16K programs on an ESF tape (wafer) and the wafer is at the beginning of the first program, the total load time for the fourth program will be about one minute and 40 seconds. That still beats four minutes.

My unit arrived complete with power supply, connecting cables, programs and manual. It was all well packed.

I did have trouble with the starter kit programs which came with the ESF. There are eight programs on the wafer. Some are SYSTEM and some BASIC. None of the BASIC programs would load, so I made backup copies of the SYSTEM programs and while trying to load the BASIC again, the ESF ate the tape (pulled out of the wafer). I've had one other tape eaten. I returned them and Exatron replaced them free.

The second starter kit tape would not load at all, so I called the toll free number and talked to a technician. He agreed to send one more starter kit tape. When it arrived, it worked and I've had no tape problems since.

One of the nicest things about an ESF is its lack of RAM usage. Data handling is done with software which takes about 1100 bytes and sits below BASIC, but the @ Save, @ Load and @ New commands are in a ROM which is called up by typing SYSTEM (Enter) then typing /12345 (for debounce) or /12346 (for non-bounce). Four bytes are then loaded below the memory size to point to the ROM and the memory size reset to protect the bytes.

Even these four bytes do cause problems. In some BASIC programs which contain machine language, the program sets its own memory size. Since the ESF ROM looks at the memory size to find where its four control bytes are, any change will cause the ESF not to start on an @ Load or @ Save. While following the above procedure will bring back the ESF control, it's a problem which can be eliminated by using the program in Listing 1.

This program will tell where and what to POKE for auto memory size and relocation of the ESF bytes. Non ESF owners can use the memory sizer only, by skipping the numbers below with (*) in front of them. To use the program:

1. Type SYSTEM (Enter).
2. Type /zero (Enter).
3. Enter the desired memory size.

*4. Type SYSTEM (Enter).

*5. Type /12345 or /12346 (Enter).

6. Load the program in Listing 1 and run it.
7. At the start of your program, add a line which pokes each address with the values shown. (Non ESF owners should poke only addresses 16561 and 16562).
8. End the line with a CLEAR 50 (or whatever your program requires).

Now, when your program runs, it will set its own memory size and relocate the ESF bytes. If your program already sets its own memory size, that line should be removed from the program.

SYSTEM tapes cause a different problem. If the ESF bytes are overwritten while a SYSTEM program is loading, the drive will not shut off and you may end up with a memory size message. The only way I've found around this is to set the memory size below the starting address of the SYSTEM program and recall the ESF. Now the system program will load. I've found most SYSTEM programs will load with memory size set at 18000. If not, they should work with no memory size set. A nice feature with SYSTEM tapes is the auto start, which starts running the program as soon as it is loaded.

Backing up BASIC programs is as easy as @ Save and SYSTEM programs are almost as easy. First, the ESF-80 monitor is used to load in the SYSTEM program and display the start, end and auto start addresses. Then the WRITE command is used to put the program on a wafer.

Beware: The start and auto start addresses are not always correct. Programs which contain multiple

origins or load copyrights on the screen while loading, can fool the monitor. If the normal procedure doesn't work, try using the starting address as the auto start address. If it still won't work, try using the auto start address as the starting address. I have found that one of these solutions will work about 70 percent of the time.

Some programs can't be copied (like Pinball by Acorn) but patches are available (for a fee) to allow their use with the ESF.

Even more impressive than program loading, is the data handling. As stated, the Data I/O program supplied with the starter kit is used for data handling. It adds the commands @ Open, @ Close, @ Clear, @ Print and @ Input. While the ESF manual is well written, I do feel that it is lacking in the area of data handling. Exatron does provide two good Data I/O demo programs as a manual supplement which help to fill in the gaps. One program I wrote to keep track of my recordings took 45 minutes to run on cassette. It runs in under three minutes with the ESF.

The only real problem I have found with the ESF is the four to five weeks it takes to get wafers when I order them from Exatron. While the cost of a wafer is not expensive ($3 to $5, depending on length), because of the time lag, I tend to order a dozen at a time which is a mild expense each time. But then, love is not always easy.

All in all, the Stringy Floppy does what it claims and does it well. Most of the problems I've experienced can be corrected with programming. I feel it's a good buy at a good price.

Listing 1 @ Love

```
0 REM AUTO MEMORY SIZER FOR MODEL I 16K
   AND ESF
10 CLS
20 INPUT"If you don't have a printer ty
  pe - SCREEN - <ENTER>
 other wise type - <ENTER> ";AN$
30 IF AN$="SCREEN"THEN 40   ELSE 50
40 POKE16422,88:POKE16423,4
50 CLS
60 V=PEEK(16561)+PEEK(16562)*256
70 IFV>32757 THEN LPRINT"MEMORY IS NOT
  SET LOW ENOUGH TO NEED THIS PROGRAM":
  GOTO180
80 LPRINT"  Addresses to POKE"
90 LPRINT"         ESF BYTES"
100 FORX=V+1TOV+4:LPRINT"POKE";X;",";PE
  EK(X):NEXT
110 LPRINT"Machine program will start
  at  address ";V+6
120 LPRINT"Auto POKE Addresses "
130 LPRINT"16561";PEEK(16561),"16562";P
  EEK(16562)
140 LPRINT"Memory is set at ";V
150 LPRINT"Don't forget a  CLEAR 50 "
160 IFAN$="SCREEN"THEN 180
170 IFAN$<>"SCREEN"THEN AN$="SCREEN":GO
  TO30
180 POKE16422,141:POKE 16423,5 ■
```

--== oØo ==--

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  REFMODIFICATION.

AUTHOR.  MICHAEL COUGHLAN.

* AN EXAMPLE PROGRAM USING REFERENCE MODIFICATION, INTRINSIC
* FUNCTIONS AND THE INSPECT.
* THE PROGRAM SOLVES A NUMBER OF TASKS SUGGESTED BY JUDY
* YAEGER IN HER ARTICLE "CHARACTER MANIPULATION USING
* COBOL WHERE SHE ATTEMPTED TO SHOW THE LIMITATIONS OF
* COBOL STRING HANDLING.

DATA DIVISION.
WORKING-STORAGE SECTION.
01   XSTR          PIC X(50) VALUE "   THIS IS THE FIRST SOURCE
STRING".
01   XSTR2         PIC X(32) VALUE "THIS IS THE SECOND SOURCE
STRING".
01   STRSIZE       PIC 99 VALUE 32.
01   CHARCOUNT     PIC 99 VALUE ZEROS.
01   ENDCOUNT      PIC 99 VALUE ZEROS.
```

```
Ø1  YSTR          PIC X(4) VALUE SPACES.
Ø1  CHARPOS       PIC 99 VALUE ZEROS.
Ø1  STRLENGTH     PIC 99 VALUE ZEROS.

PROCEDURE DIVISION.
BEGIN.

*   TASK1 SUBSTRING(XSTR, STARTPOS, LENGTH)
*
*   EXTRACT A SUBSTRING FROM A STRING GIVEN THE STARTPOS
*   AND LENGTH. SOLUTION - USE REFERENCE MODIFICATION TO GET
*   THE SUBSTRING. IN THIS EXAMPLE WE GET 3 CHARACTERS
*   STARTING AT POSITION 9
    DISPLAY "TASK1 = " XSTR(9:3)

*   TASK2 PREFIX(XSTR,LENGTH)
*
*   EXTRACT THE FIRST LENGTH NUMBER OF CHARS FROM A STRING
*   SOLUTION - USE REFERENCE MODIFICATION STARTING AT
*   POSITION 1.
*   IN THIS EXAMPLE WE GET THE FIRST 7 CHARACTERS
    DISPLAY "TASK2 = " XSTR(1:7)

*   TASK3 SUFFIX(XSTR,LENGTH)
*
*   EXTRACT THE LAST LENGTH NUMBER OF CHARS FROM A STRING
*   SOLUTION - USE REFERENCE MODIFICATION WITH START OF
*   SUBSTRING DEFINED AS THE FULLSTRINGLENGTH -
*   SUBSTRINGLENGTH + 1
*   IN THIS EXAMPLE WE GET THE LAST 13 CHARACTERS.
    MOVE 13 TO STRLENGTH
    DISPLAY "TASK3 = " XSTR2((STRSIZE - STRLENGTH) +
1:STRLENGTH)

*   TASK4: TRIMBLANK(XSTR)
*
*   REMOVE TRAILING BLANKS FROM A STRING
*   SOLUTION 1
*   USE THE REVERSE INTRINSIC FUNCTION TO REVERSE THE STRING
*   THEN USE THE INSPECT TALLYING TO COUNT THE NUMBER OF
*   SPACES AT THE
*   BEGINING OF THE REVERSED STRING.  THE SUBSTRING LENGTH
*   IS THEN THE
*   FULLSRINGLENGTH - CHARCOUNT.
*   USE REFERENCE MODIFICATION OF GET THE SUBSTRING.
    DISPLAY "TASK4 BEFORE = " XSTR "<<<<<<"
    MOVE Ø TO CHARCOUNT
    INSPECT FUNCTION REVERSE(XSTR) TALLYING CHARCOUNT
        FOR LEADING SPACES
    DISPLAY "TASK4 AFTER = "XSTR(1:5Ø - CHARCOUNT)
"<<<<<<<".

*   SOLUTION 2
*   USE REFERENCE MODIFICATION AND THE PERFORM..VARYING TO
*   KEEP REDUCING THE SIZE OF THE SUBSTRING UNTIL A NON
*   SPACE CHARACTER
*   IS ENCOUNTERED.  THEN USE REFERENCE MODIFICATION TO
```

```
*    GET THE SUBSTRING.
     PERFORM VARYING CHARCOUNT FROM 50 BY -1
           UNTIL XSTR(CHARCOUNT:1) NOT = SPACE
     END-PERFORM
     DISPLAY "TASK4 AFTER = "XSTR(1:CHARCOUNT) "<<<<<<<".

*    TASK5 LEFT_TRIMBLANK(XSTR)
*
*    REMOVE LEADING BLANKS FROM A STRING.
*    SOLUTION - USE THE INSPECT TO COUNT THE LEADING BLANKS
*    AND REFERENCE
*    MODIFICATION TO GET THE SUBSTRING FROM THE POINT
*    INDICATED BY CHARCOUNT
*    AND FOR FULLSTRLENGTH - CHARCOUNT CHARACTERS.
     MOVE 1 TO CHARCOUNT.
     INSPECT XSTR TALLYING CHARCOUNT FOR LEADING SPACES
     DISPLAY "TASK5 =" XSTR(CHARCOUNT: 50 - CHARCOUNT)

*    TASK6 INDEX(XSTR,YSTR)
*
*    FIND THE LOCATION OF THE FIRST OCCURRENCE OF SUBSTRING
*    YSTR IN XSTR. SOLUTION - USE THE INSPECT..TALLYING TO
*    COUNT THE CHARACTERS BEFORE THE FIRST OCCURRENCE OF THE
*    SUBSTRING. CHARCOUNT HAS THE LOCATION.
*    IN THIS EXAMPLE WE GET THE POSITION OF THE SUBSTRING
*    "SOURCE".
     MOVE 1 TO CHARCOUNT
     INSPECT XSTR TALLYING CHARCOUNT FOR CHARACTERS
         BEFORE INITIAL "SOURCE".
     DISPLAY "TASK6 FIRST OCCURRENCE IS IN CHAR POSITION "
CHARCOUNT

*    TASK7 CINDEX(XSTR,YSTR)
*
*    FIND THE LOCATION OF THE FIRST OCCURRENCE OF ANY OF
*    THE CHARACTERS IN SUBSTRING XSTR, IN STRING YSTR
*    SOLUTION - USE THE PERFORM..VARYING AND REFERENCE
*    MODIFICATION TO LOCATE EACH OF THE CHARACTERS IN THE
*    YSTRING.  THEN USE THE INSPECT TO FIND THE POSITION OF
*    EACH IN THE XSTRING.  RETURN WHICHEVER IS THE LEAST.
     MOVE "FRED" TO YSTR
     MOVE 51 TO ENDCOUNT
     PERFORM VARYING CHARPOS FROM 1 BY 1 UNTIL CHARPOS > 4
        MOVE 1 TO CHARCOUNT
        INSPECT XSTR TALLYING CHARCOUNT FOR CHARACTERS
             BEFORE INITIAL YSTR(CHARPOS:1)
        IF CHARCOUNT < ENDCOUNT MOVE CHARCOUNT TO ENDCOUNT
        END-IF
     END-PERFORM
     DISPLAY "TASK7 FIRST OCCURRENCE IS IN CHAR POSITION "
ENDCOUNT
     DISPLAY "THE CHARACTER IS " XSTR(ENDCOUNT:1)
     STOP RUN.
```

TRS8BIT

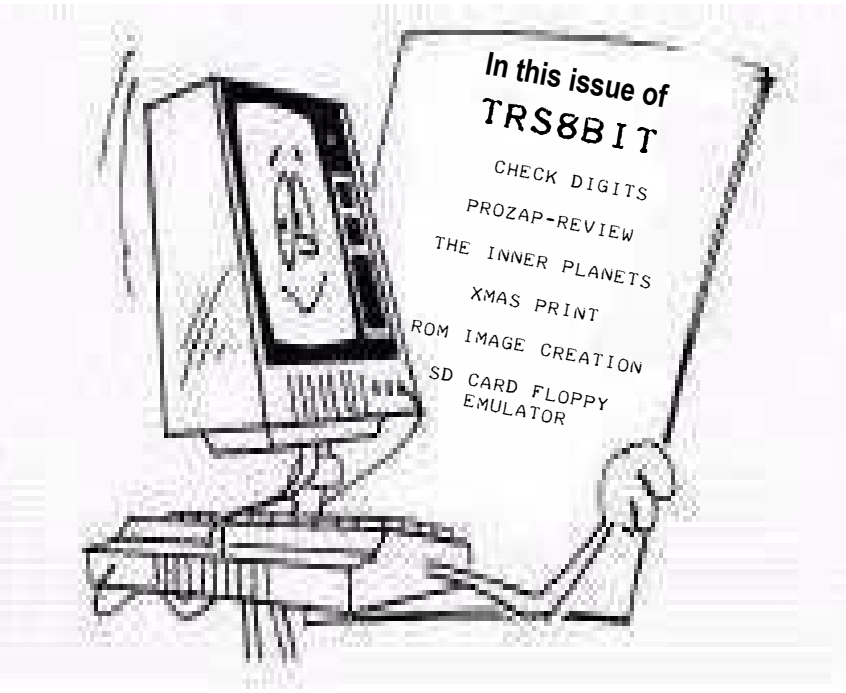


HELLO EVERYONE AND WELCOME TO THE 2010 XMAS EDITION OF TRS8BIT.

DEE & I WISH YOU ALL A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR.

THERE HAS BEEN A FAIR BIT OF DEVELOPMENT IN THE CURRENTLY AVAILABLE EMULATOR RELATED SOFTWARE SO, JUST TO ENSURE YOU ARE UP TO DATE, THE FOLLOWING SHOULD BE OF INTEREST TO YOU.
MIGUEL DUTRA HAS RELEASED V0.50 OF HIS VIRTUAL FLOPPY DISK MANAGER AND MATTHEW REED HAS UPDATED HIS TRS32 EMULATOR TO VERSION 1.25, TRSTOOLS TO V1.04 AND TRSREAD/TRSWRITE TO V4.27. THERE ARE THE USUAL LINKS ON THE WEBSITE TO MAKE UPDATING EASIER. IT ONLY TAKES A FEW SECONDS.

EBAY HAS BEEN CONSIDERABLY BUSIER OVER THE LAST 3 MONTHS. A COUPLE OF MODEL 1'S MADE AN APPEARANCE AND FETCHED ABOUT £30 EACH (ONE WITH AN EI AND A DISK DRIVE!) AND A COUPLE OF VIDEO GENIES (3001 & 3003) WERE UP FOR GRABS, MAKING JUST UNDER £70 EACH.



I RECEIVED AN EMAIL FROM PAUL RUTHERFORD ASKING IF I COULD 'ADVERTISE' A NUMBER OF TANDY ITEMS HE HAD FOR SALE. I PLACED A COUPLE OF LINES ON THE WEBSITE AND POSTED A LISTING ON TRS-80 OLD-COMPUTER FORUM. HE EMAILED BACK WITHIN A COUPLE OF WEEKS TELLING ME THAT MOST OF THE BITS HAD SOLD SO HE IS A VERY HAPPY BUNNY. THE ONLY BITS LEFT WERE SOME 80-MICROS. I WAS SO PLEASED WE COULD HELP. PAUL HAS ALSO HAD SOME REQUEST FOR ADDITIONAL BIT AND PIECES, SO IF YOU'VE ANY BIT YOU WISH TO CLEAR OUT, PLEASE EMAIL HIM.

PETER STONE HAS BEEN BEAVERING AWAY WITH THE ACULAB ROM MEMORY DUMP I SENT TO HIM. HE MANAGED TO PRODUCE A BINARY IMAGE OF THE ROM WHICH MATTHEW'S EMULATOR RECOGNISES, (IN PART DUE TO THE EXATRON'S ROM BEING MUCH SMALLER). I'M NOW EXCITEDLY WAITING TO SEE IF MATTHEW CAN PRODUCED THE UNBELIEVABLE, AND GET THE FT UP AND RUNNING ON THE EMULATOR.

FOR THOSE OF YOU WHO FOLLOW THE WEBSITE CLOSELY, YOU'LL HAVE NO DOUBT SEEN MY SUGGESTION FOR A 'TANDY TYPE' WORKSHOP IN THE SPRING. IT

WAS JUST AN IDEA THAT POPPED INTO MY HEAD AND I WONDERED IF ANYONE ELSE FANCIED THE IDEA. MY MAIN THOUGHT, IS TO HAVE A VERY CHEAP ONE DAY (SATURDAY?) MEET SOMEWHERE WHERE THERE'S GOOD BUS AND RAIL CONNECTIONS AND ALSO LOTS OF CHEAP ACCOMMODATION, SHOULD ANYONE NOT WISH TO TRAVEL TOO FAR IN A DAY. I WOULD ESTIMATE THAT EVEN HALF A DOZEN PEOPLE WOULD MAKE IT WORTH THE EFFORT, SO IF YOU FANCY A TRIP OUT WITH YOUR TANDY, PLEASE EMAIL ME AND I'LL SEE WHAT I CAN COME UP WITH.

I'VE BEEN VERY LUCKY ON EBAY RECENTLY, AND MANAGED TO WIN A FEW (NO NUMBERS MENTIONED, I DON'T WANT TO UPSET DEE!) PERSONAL COMPUTER WORLDS FROM THE EARLY 1980'S. THEY HAVE, BY THEIR VERY NATURE, A LIMITED NUMBER OF INTERESTING ARTICLES FOR THE TANDY RANGE, BUT ABOVE ALL, THEY DO HAVE SOME WONDERFUL COMPUTER ADVERTS. I SHALL FEATURE THEM THROUGHOUT THE COMING YEAR, STARTING WITH THIS ISSUE. BOTH TANDY, VIDEO GENIE AND MISCELLANEOUS BIT WILL BE REPRODUCED.

NOW-DOWN TO THIS ISSUE. I'VE MANAGED TO GET A VARIED SELECTION OF ITEMS FOR YOUR DELIGHT. THERE'S AN ARTICLE ABOUT CHECK DIGITS WHICH HAD ME SCRATCHING MY HEAD FOR A WHILE. A REVIEW OF NIGEL DIBBEN'S 'PROZAP'.
A PROGRAM TO DISPLAY THE PATHS OF THE INNER PLANETS AROUND THE SUN, WHICH WITH A BIT OF JUDICIOUS JIGGLING, I'VE ALSO MANAGED TO GET DOWN TO 1 LINE.

AS IT'S CHRISTMAS, THERE'S THE USUAL XMAS PRINT, THIS YEAR FEATURING AN XMAS TREE! JUST TO PUT YOU IN THE RIGHT MOOD SO TO SPEAK. PETER STONE HAS, AGAIN, DONE US PROUD, WITH AN ARTICLE ON ROM IMAGE CREATION. THIS IS ALL PART OF THE EXERCISE OF TRANSFERRING THE ACULAB FLOPPY TAPE ROM TO A ROM IMAGE FILE WHICH WILL ALLOW THE USE OF AN FT ON MATTHEW REED'S EMULATOR.
AND FINALLY, FROM INFORMATION I CAN ACROSS ON THE WEB, ARE BRIEF DETAILS OF THE SD-CARD FLOPPY EMULATOR. IF ANYONE IS USING THIS, WE'D ALL LIKE TO HEAR HOW YOU'RE GETTING ON WITH IT. IT SOUNDS MOST IMPRESSIVE.
I HOPE YOU FIND IT AN INTERESTING AND VARIED SELECTION!

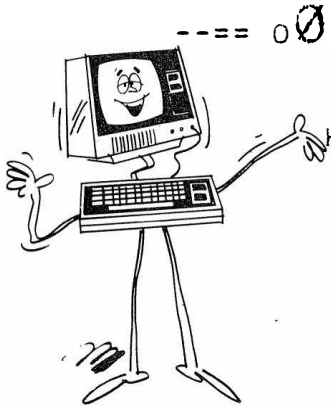
ALL THAT REMAINS FOR ME TO DO IS TO, ONCE AGAIN, WISH YOU ALL A MERRY CHRISTMAS AND A HAPPY NEW YEAR. I LOOK FORWARD TO HEARING FROM YOU IN 2011.

BYE FOR NOW

DUSTY



--== O0O ==--



HAPPINESS-
PLAYING
WITH YOUR
M1 WHEN
IT'S
SNOWING
OUTSIDE!

CHECK DIGITS

BY DAVE BARROW

(ORIGINALLY PUBLISHED IN PCW)

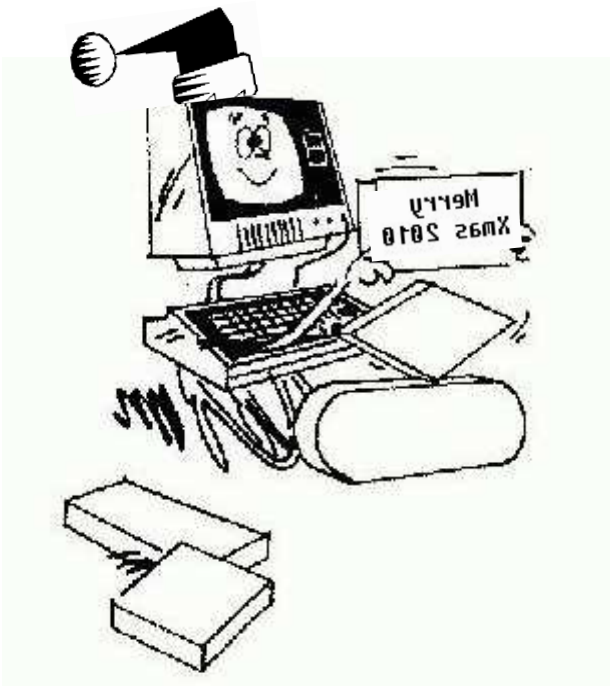
AT THE END OF CHAPTER 8 IN LANCE LEVENTHAL'S Z80 ASSEMBLY PROGRAMMING, HE SET AN EXERCISE TO CALCULATE A CHECK DIGIT FROM A STRING OF BCD DIGITS WITH THE WEIGHTING 1, 3 AND 7, MOD 10. DAVE BARROW CAME UP WITH THIS NEAT ROUTINE.

TO REDUCE THE CHANCES OF MISS-KEYING IMPORTANT NUMBERS, FOR EXAMPLE, A CLIENT OR PATIENT NUMBER, IT IS COMMON FOR SUCH NUMBERS TO HAVE A CHECK DIGIT ADDED ON THE END. THE CHECK DIGIT IS CALCULATED BY MULTIPLYING EACH DIGIT OF THE BASIC NUMBER BY A WEIGHTING DIGIT FROM A FIXED SEQUENCE, ADDING ALL THE RESULTS TOGETHER, DIVIDING THE FINAL RESULT BY 10 AND TAKING THE ONE-DIGIT REMAINDER AS THE CHECK DIGIT.

IF, FOR EXAMPLE, THE WEIGHTING DIGITS ARE 1, 3 AND 7, A BASIC NUMBER OF 2784 BECOMES 27843 WITH THE CHECK DIGIT, 3, CALCULATED:-

```
2 X 1 =  2
7 X 3 = 21
8 X 7 = 56
4 X 1 =  4
        --
        83 / 10 = 8 REMAINDER 3
```

IF ANY DIGIT IN THE NUMBER 27843 IS ENTERED INCORRECTLY, IT IS PROBABLE THAT THE NUMBER WILL FAIL THE CHECK DIGIT TEST AND THE USER CAN BE ASKED TO RE-ENTER THE NUMBER CORRECTLY.



NOTE.
WHEN I TRIED TO ASSEMBLE THE SOURCE FILE OVERLEAF, IN ZEN, IT BAULKED AT THE 'SLI' INSTRUCTION, (SHIFT LEFT AND INCREMENT (IT'S CLASSED AS NON-EXISTENT BUT IT REALLY IS A 'NEAT' IDEA WITH 1, 3 & 7). I HAD TO 'CHEAT' AND PUT A 'NOP' INSTRUCTION IN ITS PLACE JUST TO GET THE REST OF THE PROGRAM TO ASSEMBLE OK AND THEN EDIT THE LISTING OVERLEAF WITH THE CORRECT FIGURES.

```
 1                              ORG  8000H
 2 8000 F5                      PUSH AF
 3 8001 C5                      PUSH BC
 4 8002 D5                      PUSH DE
 5 8003 AF                      XOR  A          ;INITIALISE CHECKSUM AND
 6 8004 57                      LD   D,A        ;MULT. FACTOR TO ZERO
 7 8005 5E              NXTBZ:  LD   E,(HL)     ;SAVE BYTE
 8 8006 0E02                    LD   C,2        ;SET DIGIT COUNTER
 9 8008 C5              NXTDZ:  PUSH BC         ;SAVE BYTE AND COUNTER
10 8009 47                      LD   B,A        ;GET NEXT DIGIT IN
11 800A ED6F                    RLD             ;LOW-NIBBLE C
12 800C 4F                      LD   C,A        ;AND
13 800D 78                      LD   A,B        ;CHECKSUM IN A
14 800E CB32                    SLI  D          ;M M*2+1
15 800F 42                      LD   B,D        ;ADDITIVE LOOP COUNT
16 8010 81             MADDZ:   ADD  A,C        ;CHECKSUM CHECKSUM
17 8011 27                      DAA             ;(DIGIT * M)
18 8012 10FC                    DJNZ MADDZ      ;
19 8014 CB52                    BIT  2,D        ;IF M=7 THEN M=0
20 8016 2801                    JR   Z,TDIG     ;
21 8018 50                      LD   D,B        ;
22 8019 C1             TDIG:    POP  BC         ;RESTORE BYTE AND DIGIT COUNTS
23 801A 0D                      DEC  C          ;
24 801B 20EB                    JR   NZ,NXTDZ   ;
25 801D 73                      LD   (HL),E     ;RETORE BYTE AND
26 801E 23                      INC  HL         ;POINT TO NEXT
27 801F 10E4                    DJNZ NXTBZ      ;
28 8021 E60F                    AND  0FH        ;MOD 10
29 8023 D1                      POP  DE         ;
30 8024 C1                      POP  BC         ;
31 8025 47                      LD   B,A        ;RETURN CHECKSUM IN B
32 8026 F1                      POP  AF         ;
33 8027 C9                      RET             ;
34                              END
```



IF I RECALL CORRECTLY, LAURIE SHIELDS DID MANY LATER IMPROVEMENTS TO ZEN (I THINK HE EVEN BROUGHT OUT A CP/M VERSION!), SO IF YOU HAVE A LATER VERSION THAN MINE, (I USE ZEN84) WOULD YOU LET ME KNOW HOW YOU GET ON?

DOES IT ASSEMBLE OK WITH EDTASM OR THE MICROSOFT EDITOR/ASSEMBLER? OR CAN YOU THINK OF ANOTHER WAY AROUND THE PROBLEM?

REVIEW OF

PROZAP

BY KEN GREY

THE BASIS OF THIS ARTICLE FIRST APPEARED IN NATGUG NEWS.

PROZAP WAS WRITTEN BY NIGEL DIBBEN OF COMPAK FAME.

THE FIRST THING YOU NOTICE IN THE MANUAL IS THAT PROZAP CAN BE SET UP TO READ A MIXTURE OF DRIVES ON THE SAME CABLE AND YOUR BACK-UP COPY, ZAPPING THE APPROPRIATE CHANGES ONLY TAKES A COUPLE OF MINUTES. AT THE SAME TIME YOU HAVE THE OPPORTUNITY TO SET THE DRIVE STEPPING RATE FOR EACH DRIVE. THE PROGRAM LIES IN MEMORY FROM 8000H TO APPROX. A000H SO IT WILL NOT INTERFERE WITH MOST DOS PROGRAMS BUT COULD OVERLAP SOME OF YOUR PROGRAMS. PROZAP WILL COPY ALL BUT DOUBLE DENSITY AND IMPROPERLY FORMATTED TRACKS. THERE ARE SOME VERY NICE AND NOVEL FEATURES IN THE PROGRAM, FOR EXAMPLE TYPING PROZAP, FOLLOWED BY YOUR PROGRAM FILE SPEC, PROZAP WILL IMMEDIATELY ENTER ITS RELATIVE FILE MODE, SEEK THEN DISPLAY THE FILE. ANOTHER FEATURE IS THE SYSTEM SELECTION ENABLING YOU TO SPECIFY THE TYPE OF DOS, VIZ TRSDOS, NEWDOS, VTOS, LDOS OR UNKNOWN. ENTERING 'R' IN THE COMMAND MODE TAKES YOU TO THE RELATIVE FILE SECTOR MODE, TO ACCESS YOUR FILE ALL YOU HAVE TO DO IS ENTER THE FILENAME/EXT AND PROZAP WILL GIVE A FILE MAP SHOWING WHERE, HOW MANY SECTORS ETC. THE FILE HAS, THEN, HITTING <ENTER> WILL TAKE YOU TO THE DISPLAY MODE OF THE FILE'S FIRST RELATIVE SECTOR. QUICK, NEAT AND EASY. A SAMPLE OF THE DISPLAY SHOWN BELOW.

THE COPY DISK FACILITY IS, AGAIN, NEAT AND EASY TO USE AND BETTER THAN MOST OTHER ZAP PROGRAMS I'VE TRIED. TWO FACILITIES I REALLY LIKE ARE THE FIND BYTE/WORD, MUCH BETTER THAN GETTING EYE STRAIN PEERING AT THE VDU LOOKING FOR THAT ELUSIVE BYTE. THE OTHER IS THE ASCII MODIFY WHERE ALL YOU DO IS PLACE THE CURSOR OVER THE ASCII CHARACTER IN THE RIGHT HAND COLUMN AND TYPE! I USED THIS FACILITY TO MODIFY THE SHORT-HAND COMMANDS IN COMPAK TO GIVE ME A PERMANENT FIX WITH THE COMMANDS THAT WERE MORE USEFUL TO ME THAN THOSE PROVIDED.

A SINGLE KEY STROKE WILL GIVE YOU A FILES HASH CODE AND THE MATCH FACILITY WILL LET YOU COMPARE ONE SECTOR WITH ANOTHER, DISPLAYING THE DIFFERENCES. THERE ARE MANY USEFUL FEATURES OF THIS PROGRAM THAT CAN NOT BE FOUND IN ANY OTHER SINGLE PROGRAM AND ONCE YOU HAVE THE 'FEEL' OF USING PROZAP, IT SAVES HOURS OF WORK FOR THE ARDENT DISK ZAPPER.

THE MANUAL IS WELL-WRITTEN AND QUITE EASY TO FOLLOW BUT DO READ THE MANUAL CAREFULLY, OR, LIKE ME, YOU COULD MAKE THE ODD MISTAKE AND GET RATHER HOT UNDER THE COLLAR ONLY TO FIND THAT IT'S YOU AND NOT THE PROGRAM.

PROZAPS COMMANDS
IN COMMAND LEVEL
<ENTER> SPECIFY - DRIVE
TRACK SECTOR
RECALL BUFFER

(CONTINUED ON PAGE 6)

(CONTINUED FROM PAGE 5)

COPY ALL OR PART OF DISK
LOAD DIRECTORY SECTOR
EXIT TO DOS
ENCIPHER PASSWORD
TRACK IDENTITY CHECK MODE
TRACK/SECTOR USAGE
SELECT DOS TYPE
RELATIVE FILE SECTOR MODE
READ TRACK INTO MEMORY
ENTER DEBUG

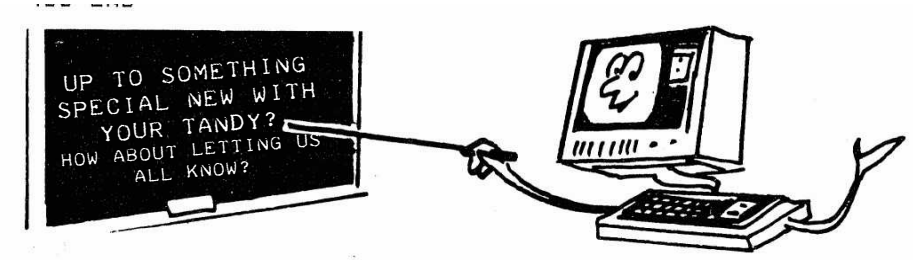
IN DISPLAY LEVEL
HEX MODIFY
ASCII MODIFY
ENCIPHER PASSWORD
DISPLAY HASH CODE
DISPLAY SAME TRACK FROM
DRIVE N:
JUMP TO BYTE NN
DISPLAY TRACK/SECTOR USAGE
LOAD BUFFER FROM MEMORY
SELECT SECTOR NUMBERING
LIMITS
OUTPUT SECTOR
PRINT SECTOR TO LINE
PRINTER

RELATIVE FILE SECTOR MODE
SAVE SECTOR TO MEMORY
GOTO NEW SECTOR
ENTER DEBUG
FIND BYTE XX
FIND WORD XXXX
ZERO BYTES
<CLEAR KEY> RETURN TO
COMMAND LEVEL

FROM THE ABOVE YOU CAN SEE
THAT PROZAP IS VERY
VERSATILE PROGRAM WITH A
LOT OF FACILITIES.

```
D=00 00 1151613A 0100FC53 3555B7C4 2444CDB7   .Qa:...S5U..$D..
T=13 10 6011B064 213A63CD DE552ABA 64223660   '..d!:c..U*.d"6'
S=02 20 2A146322 3C603A17 63324260 CD46600E   *.c"<':.c2B'.F'.
B=00 30 0A41215A 63CD3257 CD5557CD 4D60CDC8   .A!Zc.2W.UW.M'..
A=05 40 5F28i3FE 062810FE 1CCA0D55 FE1DCA0D    (...(.....U....
     50 55CDA260 20EBAF77 DD3404CD 6B5710D8   U..' ..w.4..kW..
     60 21055422 FD54CDD8 553A3357 B720272A   !.T".T..U:3W. '*
     70 DA647CB5 20202ABC 643AB864 B720012B   .d!.  *.d:.d. .+
     80 7CA53C28 11FDCBEA CE22DA64 CDCE5FFD   !.<(.....".d.._.
     90 CBEABEC2 335241CD 35602163 63CD3257   ....3RA.5'!cc.2W
     A0 CD5557CD 4D607EFE 062004DD CB00C6CD   .UW.M'~.. ......
     B0 CE5FDDCB 00862805 CDA26020 E9DD3404   . ....(... ' ..4.
     C0 CD6B5710 DBCD3560 412A4C65 CD2C572A   .kW...5'A*Le.,W*
     D0 5265E5CD 5557CD4D 60CDC85F 2003AF18   Re..UW.M'.._ ...
     E0 0CFE0628 08CDA260 20EFE118 16BEE1C5   ...(...' .......
P___ F0 D50600ED 5B4C6520 121ABE13 23200C10   ....[Le ....# ..
```

--== o0o ==--



THE INNER PLANETS

(HOLTS- EAT YOUR HEART OUT)

FROM AN ORIGINAL IDEA BY C WARD

HERE'S ONE OF THOSE IDEAS WHICH GREW, JUST LIKE TOPSY. I CAME ACROSS A PROGRAM, WRITTEN ORIGINALLY BY C. WARD, FOR THE ZX81, WHICH CALCULATED AND DISPLAYED THE POSITION OF THE INNER PLANETS, MERCURY, VENUS, EARTH AND MARS, RELATIVE TO THE SUN.

I THOUGHT IT SHOULDN'T BE TOO DIFFICULT TO CONVERT TO RUN ON THE MODEL 1 AND AS IT WAS A REALLY NEAT LITTLE PROGRAM, I REALLY FANCIED HAVING A GO. AND THEN I GOT TO THINKING, (ALWAYS RISKY), WOULDN'T IT MAKE A GREAT ONE-LINER FOR THE XMAS EDITION? WOULD IT BE POSSIBLE TO SQUEEZE IT ALL DOWN?

THE ONLY SERIOUS PROBLEM I ASSUMED I WOULD ENCOUNTER WOULD BE DUE TO THE LACK OF DETAIL WITH THE MODEL 1'S GRAPHICS. MIND YOU, I SHOULD NOT WISH TO NAVIGATE TO OR FROM THE PLANETS USING A MODEL 1 AS A 'TOMTOM'!

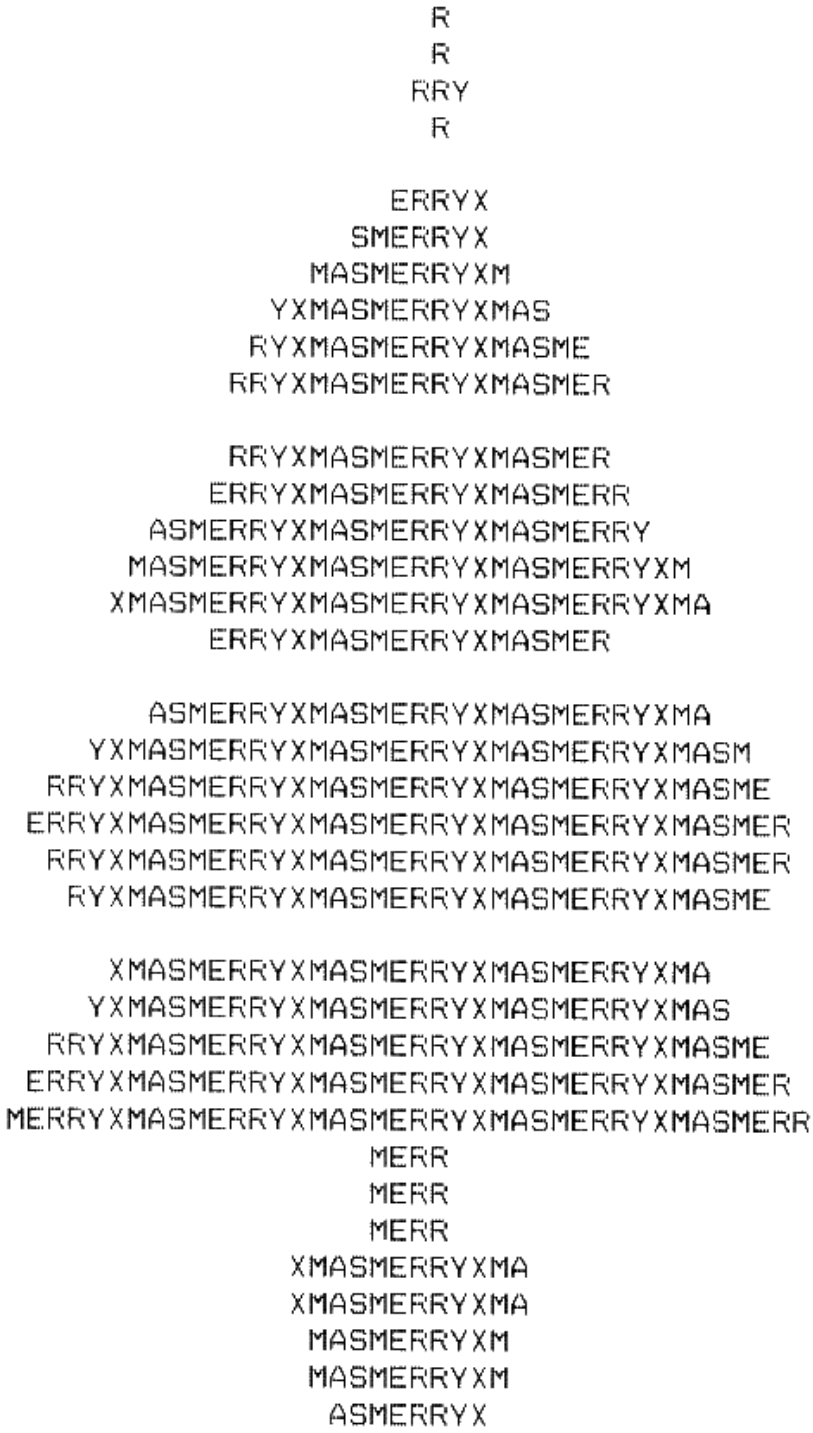
I HAVE MODIFIED AND REDUCED THE ORIGINAL PROGRAM DOWN TO EIGHT LINES WHICH ALLOWS IT TO BE EASILY FOLLOWED. THE ONE-LINER IS AT THE BOTTOM OF THE PAGE, AND LIKE 'C', IS A BIT OBTUSE TO FOLLOW. I'VE HAD TO REDUCE THE NUMBER OF CHARACTERS FOR ALL THE CALCULATING FIGURES DUE TO THE LACK OF SPACE, BUT EVEN SO, IT STILL SEEMS TO RUN IN AN EFFECTIVE MANNER. YOU WILL NEED TO ISSUE A 'CLS' COMMAND BEFORE RUNNING. IS THIS ANOTHER MODEL 1 FIRST? HOPE IT MAKES YOU SMILE ANYWAY!

I HAVEN'T PLACED THE PROGRAMS AS DOWNLOADS ON THE WEBSITE AS THEY ARE QUITE SMALL, HOWEVER, AS ALWAYS, IF YOU HAVE A PROBLEM TYPING THEM IN, PLEASE EMAIL ME.

```
10 A(1)=-327.95626:A(2)=0:A(3)=-258.47927:A(4)=-135.14322:P=3.14
2857:B(1)=-.524037:B(2)=-.98561:B(3)=-1.6021291:B(4)=-4.0923507
20 C(1)=20:C(2)=13.180432:C(3)=9.5330396:C(4)=5.1013216
30 CLS:SET(64,24)
40 FORX=1TO4:X(X)=64+SIN(A(X)/180*P)*C(X):Y(X)=24+COS(A(X)/180*P
)*C(X):NEXTX
50 FORX=1TO4:SET(X(X),Y(X)):NEXTX
60 FORX=1TO4:A(X)=A(X)-B(X):NEXTX
70 FORX=1TO4:RESET(X(X),Y(X)):NEXTX
80 GOTO 40
```

AND HERE IT IS AS A ONE-LINER

```
0 SET(64,24):A(1)=-328:A(3)=-258:A(4)=-135:P=.314:B(1)=-.5:B(2)=
-.98:B(3)=-1.6:B(4)=-4.1:C(1)=20:C(2)=13:C(3)=9:C(4)=5:FORZ=0TO1
STEP0:FORX=1TO4:RESET(X(X),Y(X)):X(X)=64+SIN(A(X)/18*P)*C(X):Y(X
)=24+COS(A(X)/18*P)*C(X):SET(X(X),Y(X)):A(X)=A(X)-B(X):NEXT:NEXT
```



DEE AND I WISH YOU ALL A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR

THANK YOU ALL FOR YOU HELP, ENCOURAGEMENT AND SUPPORT OVER THE LAST 4 YEARS, I'VE ENJOYED EVERY MINUTE AND I'M LOOKING FORWARD TO PRODUCING VOLUME 5!

XMAS 2010

IT WOULDN'T BE CHRISTMAS WITHOUT THE USUAL XMAS FESTIVE 'BIT-O-NONSENSE' SO IN THE TRUE TRADITION OF TRS8BIT, THIS YEAR I'VE GONE FOR A FESTIVE XMAS TREE TO BRIGHTEN UP YOUR HOLIDAY.

YOU MIGHT NOTICE THAT THERE IS AN EXTRA LINE IN THIS YEARS PROGRAM. NUMBER 175 HAD BEEN ADDED TO ALLOW FOR BLANK LINES, BUT OTHER THAN THAT, (AND THE DATA STATEMENTS) IT'S THE USUAL ONE, SO IT WONT REQUIRE MUCH ADDITIONAL TYPING IF YOU WISH TO RUN IT.




```
10 CLEAR500
20
M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASME
RRYXMASMERRYXMAS"
30 GOSUB 200
40 READ S
50 IF S=0 GOSUB 150
60 READ L
70 T$=MID$(M$,S,L)
80      X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256
90 REM THIS AND THE NEXT LINE ARE NOT NEEDED ON 16K SYSTEMS
100      IF X>32767 THEN X=X-65536
110          FOR Y=1TOL
120          POKE X+S+Y,ASC(MID$(T$,Y,1))
130          NEXT Y
140 GOTO 40
150 LPRINT P$
160 GOSUB200
170 READ S
175 IFS=0THENLPRINTP$:READS
180 IF S=99 THEN END
190 RETURN
200 P$=STRING$(63,32)
210 RETURN
300 DATA 22,1,0,22,1,0,21,3,0,22,1,0,0,20,5,0,18,7,0
310 DATA 16,10,0,14,14,0,13,17,0,12,19,0,0,12,19,0,11,21,0
320 DATA 8,25,0,7,28,0,6,30,0,11,20,0,0,8,28,0,5,33,0,3,36,0
330 DATA 2,38,0,3,37,0,4,35,0,0,6,30,0,5,32,0,3,36,0,2,38,0
340 DATA 1,40,0,19,4,0,19,4,0,19,4,0,15,12,0,15,12,0,16,10,0
350 DATA 16,10,0,17,8,0
440 DATA 99
```

The b



TRS-80 Model I
from **£379**
4K Level I
£559
16K Level II
(illustrated).

£559.00* (16K Level II)

TRS 80 Model I Microcomputer Here's the push button brain, the entertainer, the tutor, the timesaver. Fun for the children, a helper for the businessman and the teacher, a catalogue for the housewife, an analyser for the investor and an informer for the salesman. Run Maths, English, Chess, Draughts and video game programmes for educational fun.
Easy to learn and operate — you can even write your own programmes. Suddenly you have a ready and reliable source of brainpower — put it to work immediately.



£289.00* **£439.00***

TRS 80 Model I C.P.U.'s Complete and ready to run from your TV monitor. Just plug in and start computing. The lowest priced 4 K level I CPU contains 4096 bytes of user memory and can be expanded to 16K within the keyboard unit. The 16K level II CPU is a more powerful and flexible version using an additional 12K ROM. Level II also incorporates a calculator style numeric keypad for faster data entry.

rains.



£2,199.00*

RS 80 Model II Microcomputer A bigger, more powerful brother to Model I. Designed for more data storage and versatility. Like Model I is completely modular allowing easy, plug-in expansion. It's available memory will allow maximum use of future languages.

RS-80 COMPUTER CENTRES NOW OPEN AT –
irmingham – Edgbaston Shopping Precinct, Hagley Road.
radford – 214, Forster Square.
ristol – Colston Centre, Colston Avenue.
ondon – 1-2, Seacoal Lane, Ludgate Hill, EC4.
111, Kingsway, WC2.
lanchester – 30, Market Place, Deansgate.

FULL COMPUTER FACILITIES AVAILABLE AT –
Leeds – 72, Merrion Centre.
Liverpool – 168, Market Way, St. Johns Centre.
Wednesbury – Bilston Road.
Wimbledon – 124-126, The Broadway, London, SW19

*All prices include VAT.

IT'S POSSIBLE, ONLY AT Tandy

Better Equipment. Lower Prices. No Middlemen!

CREATING A ROM IMAGE

BY PETER STONE

EARLIER THIS YEAR DUSTY MENTIONED THAT HE'D ACQUIRED AN ACULAB FLOPPY TAPE UNIT, & HOW HE PLANNED TO GET THE ROM CONVERTED INTO AN IMAGE FILE. HE CONTACTED ME FOR SOME INFORMATION ON PARTS OF THE TRS80 MEMORY MAP, & I OFFERED TO HAVE A GO AT TURNING THE DATA FILE HE HAD INTO A ROM IMAGE. THIS ARTICLE IS THE RESULT OF THAT WORK.

DUSTY SENT ME A BASIC LISTING THAT HE'D BEEN PLAYING WITH. WHEN I LOOKED AT IT AND IT DIDN'T APPEAR TO BE THE RIGHT LENGTH, BEING ONLY 828 BYTES IN LENGTH. I USE MATTHEW REED'S EMULATOR, SO I WROTE AN EMAIL TO MATTHEW ASKING ABOUT WHAT FORMAT HIS EMULATOR USED AS AN IMAGE. I HAPPENED TO MENTION THAT THE DATA I HAD WAS ONLY 828 BYTES LONG. MATTHEW ANSWERED, TELLING ME THAT HIS EMULATOR COULD HANDLE INTEL HEX, /CMD, AND RAW IMAGES, & THAT THE ESF.ROM IMAGE WAS 1920 BYTES LONG. BASED ON THIS, I CONTACTED DUSTY, GOT A FULL DATA FILE, ALONG WITH A COMMENT THAT THE FIRST ONE MUST HAVE BEEN EATEN. I THEN REMEMBERED THAT SOME YEARS AGO I'D HAD TO DO SOME WORK WITH MOTOROLA DATA STATEMENT FORMATS & PROGRAMMING THESE INTO EPROMS. AT THE TIME I'D BEEN SENT SOME INFO ON DIFFERENT FILE TYPES BY MOTOROLA. AMAZINGLY I STILL HAD THE INFO, & AFTER READING ABOUT INTEL HEX FORMATS I DECIDED TO GO WITH A RAW IMAGE FILE FORMAT. THE GIST OF THE DATA FILE DUSTY SENT ME IS SHOWN BELOW;

```
20 DATA 33,151,25,34,179,65,33,159,55,34,22,64
30 DATA 33,44,48,34,4,64,205,41,54,175,33,0
40 DATA 68,34,173,65,38,72,119,35,34,164,64,50
•
•
1670 DATA 255,0,255,195,186,50,195,201,53,195,196,52
1680 DATA 195,211,52,195,212,52,195,248,54,195,249,54
1690 DATA 255
1700 DATA 999
```

NOW, IT'S BEEN OVER SIX YEARS SINCE I LAST PLAYED WITH ANY SORT OF DATA FILES, & THEY WERE ONLY SEQUENTIAL TYPES, SO I DECIDED TO PLAY A BIT. I WROTE A PROGRAM THAT OPENED A SEQUENTIAL FILE & WROTE THE DATA STATEMENTS TO IT. THE RESULTS OF EXAMINING THE FILE, IN A SUPERZAP TYPE FORMAT, IS SHOWN BELOW;

	0001 0203 0405 0607 0809 0A0B 0C0D 0E0F	0123456789ABCDEF
00	3333 2C31 3531 2C32 352C 3334 2C31 3739	33,151,25,34,179
10	2C36 352C 3333 2C31 3539 2C35 352C 3334	,65,33,159,55,34
20	2C32 322C 3634 0D33 3020 3638 2C33 342C	,22,64.30 68,34,
30	3137 332C 3635 2C33 382C 3732 2C31 3139	173,65,38,72,119
40	2C33 352C 3334 2C31 3634 2C36 342C 3530	,35,34,164,64,50
50	0D35 3020 4441 5441 2032 3235 2C36 372C	.50 DATA 225,67,
60	3230 352C 3737 2C32 372C 3139 352C 3235	205,77,27,195,25
70	2C32 362C 3232 372C 3632 2C32 392C 3138	,26,227,62,29,18
80	2C	,

WELL, IT COULDN'T BE THAT EASY, COULD IT? AT THIS POINT, THE OUTSIDE WORLD INTRUDED & I HAD MOVE HOUSE, & SO PUT THIS PROJECT ON HOLD.

AT THE END OF SEPTEMBER I DECIDED TO PICK UP THE ACULAB PROJECT AGAIN, & STARTED THINKING ABOUT HOW TO PRODUCE A ROM IMAGE. I STARTED READING ALL THE INFO I HAD ON DATA FILE FORMATS & PROGRAMMING. I WAS STARTING TO THINK ABOUT POKING DATA ELEMENTS INTO SOME SORT OF RANDOM ACCESS FILE, WHEN I CAME ACROSS SOME INFO THAT HAD ESCAPED THE DOWNSIZING I'D HAD TO CARRY OUT AS I MOVED. THIS INFO CAME FROM THE TIME I'D WORKED AS A COMPUTER TECHNICIAN FOR A LOCAL POLYTECHNIC, & WAS SOME NOTES I'D WRITTEN ON THE WAY THE OS WAS TO SET UP FOR STUDENT USE. THIS OS WAS NEWDOS80V2, WHICH IS WHAT I USE ON THE EMULATOR, & IT GAVE ME AN IDEA ON A POSSIBLE WAY TO PROCEED. BASICALLY, WHEN WE SET UP THE OS, WE SET THE MEMORY SIZE TO 3K SHORT OF THE TOP OF MEMORY. THIS SPACE WAS RESERVED FOR OUR TECHNICIAN WRITTEN DEVICE DRIVERS OR PROGRAMS. THE REASON FOR THIS, WAS THAT IF A WARM REBOOT IS EXECUTED, SUCH AS A CMD"S" TO QUIT BASIC, THEN THE CONTENTS OF THIS AREA OF MEMORY WOULD REMAIN UNAFFECTED.

I REASONED THAT IF I INCREASED THIS SPACE TO 4K, THEN POKED THE DATA STATEMENTS INTO THIS AREA & THEN DUMPED IT TO DISK SOMEHOW, IT WOULD RESULT IN A ROM IMAGE. I STARTED BY MAKING A COPY OF MY NEWDOS80V2 DISK IMAGE. FROM THE DOCUMENTATION I FOUND THAT THE SYSTEM AP VARIABLE SET THE VALUE OF HIMEM. LOADING THE COPY I'D MADE IN DRIVE 0, I USED THE SYSTEM :0, AP=61440 & RESERVED THE TOP 4K OF MEMORY FOR MY OWN USE. (61440=0F000H). BEFORE PROCEEDING ANY FURTHER, I DECIDED TO USE SUPERZAP TO LOOK AT THE WAY THE MEMORY HAD BEEN FORMATTED AT BOOT UP, BEFORE I STARTED, THIS IS SHOWN BELOW.



BASED ON THIS, I DECIDED TO EMULATE A TRUE EPROM & SET ALL THE BYTES IN THE AREA I WAS PLANNING ON USING TO FFH, EMULATING A BLANK EPROM.

THE NEXT THING I HAD TO SORT OUT WAS HOW TO POKE THE DATA STATEMENTS INTO THE RESERVED MEMORY. THE COMMAND TO USE IS POKE M,D WERE M IS A BYTE ADDRESS & D IS DATA IN THE RANGE 0 TO 255. NOW THE ONLY PROBLEM IS THAT THE BYTE ADDRESS IS AN INTEGER VALUE, WHICH MEANS THAT IT LIES IN THE RANGE OF 0 TO 32767, ABOVE THIS THEN NEGATIVE NUMBERS HAVE TO BE USED. SIMPLY PUT, 32767D IS 07FFH WHICH IS

0111111111111111B WHEREAS 32768D IS 8000H WHICH IS 1000000000000000B, BUT THE MSB IS USED AS A SIGN BIT WHEN INTEGER NUMBERS. HENCE USING NEGATIVE NUMBERS FOR ADDRESSES ABOVE 32767. (D=DECIMAL, H=HEX & B=BINARY). FORTUNATELY THE LEVEL II MANUAL GIVES A FORMULA TO CALCULATE THESE ADDRESSES, WHICH IS NUMBER = -1*(65536-DESIRED ADDRESS). I TURNED THIS INTO THE FOLLOWING DEMO BASIC PROGRAM;

```
5  CLS                'CLEAR SCREEN
10 M=61440            'START OF RESERVED MEMORY
20 FOR C=1 TO 10      'COUNT 10 LOCATIONS
30 X=-1*(65535-M)     'CALCULATE THE NEGATIVE VALUE TO POKE
35 PRINT C,X          'PRINT THE MEMORY VALUE & THE POKE VALUE
40 M=M+1              'INCREMENT COUNT
50 NEXT               'LOOP
```

THIS PROGRAM WAS WRITTEN TO CALCULATE THE NEGATIVE NUMBERS REQUIRED. FROM THIS, I WROTE THE FOLLOWING PROGRAM;

```
20 FOR X= -4096 TO -2047    'FOR X=61440 TO 63489 (F000H TO
F801H)
30 POKE X,255               'FILL MEMORY LOCATION WITH 0FFH
50 NEXT                     'LOOP
60 END                      'DONE
```

THIS WAS TO EMULATE A BLANK EPROM AS I'D ALREADY DECIDED TO DO, AS SHOWN BELOW;

```
F000  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
F010  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
F020  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
F030  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
F040  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
F050  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
```

IT WAS A SHORT EXERCISE TO MODIFY THE PROGRAM TO POKE THE ACTUAL VALUES INTO MEMORY. AS SHOWN BELOW;

```
10 FOR X= -4096 TO -2047    'LOOP AS ABOVE
20 POKE X,255               'AS ABOVE
30 NEXT                     'AS ABOVE
100 X=-4096                 'START OF RESERVED MEMORY
110 READ A                  'READ DATA VALUE
120 IF A=999 GOTO 9999      'CHECK IF END OF FILE (EOF) THEN
9999
130 POKE X,A                'LOAD VALUE INTO MEMORY
140 X=X+1                   'INCREMENT MEMORY VALUE
150 GOTO 110                'LOOP
9999 END                    'END OF PROGRAM
10000 DATA 33,151,25,34,179,65,33,159,55,34,22,64 'START OF
DATA
10010 DATA 33,44,48,34,4,64,205,41,54,175,33,0
10020 DATA 68,34,173,65,38,72,119,35,34,164,64,50
.
.
```

```
11650 DATA 96,0,193,10,163,200,195,251,3,0,255,0
11660 DATA 255,0,255,195,186,50,195,201,53,195,196,52
11670 DATA 195,211,52,195,212,52,195,248,54,195,249,54
11680 DATA 255
11690 DATA 999      'END OF DATA
```

A FOR -- NEXT LOOP IS THE MOST ELEGANT WAY OF PROGRAMMING THE LOOPS THE PROGRAM NEEDS. UNFORTUNATELY A READ STATEMENT CAN'T BE WRITTEN INTO A FOR - NEXT LOOP. SO THE SLIGHTLY LESS ELEGANT WAY, (AT LEAST TO ME), USED IN LINES 100 TO 140 IS REQUIRED. ALL OF THE PROGRAMS LISTED ABOVE ARE SELF EXPLANATORY. BY THE WAY, IT SHOULDN'T BE INFERRED FROM THE ABOVE, THAT IT WAS A STRAIGHT FORWARD TASK. I CRASHED THE EMULATOR A FAIR NUMBER OF TIMES WITH GETTING LOOPS WORKING WRONGLY. AN EXAMPLE IS IN LINE 140, I INITIALLY HAD X=X-1 HERE, & IT CRASHED INTO THE STACK & VARIABLES STORED AT THE END OF THE PROGRAM AREA, WITH INTERESTING RESULTS AT THE TIME.

AFTER RUNNING THE PROGRAM, I HAD THE DATA IN THE RESERVED AREA OF MEMORY, SHOWN BELOW;

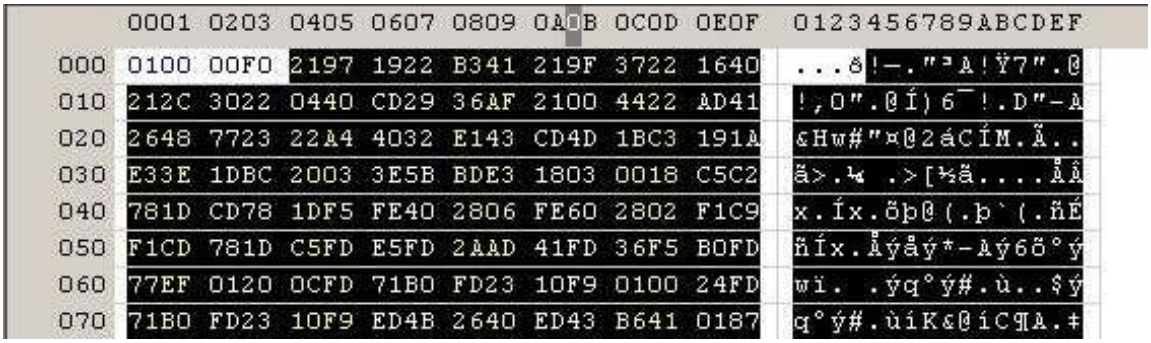
```
 F000  2197 1922 B341 219F 3722 1640 212C 3022 !.".A!.7".@!,0"
 F010  0440 CD29 36AF 2100 4422 AD41 2648 7723 .@.)6.!.D".A&Hw#
 F020  22A4 4032 E143 CD4D 1BC3 191A E33E 1DBC ".@2.C.M.....>..
 F030  2003 3E5B BDE3 1803 0018 C5C2 781D CD78 ..>↑........x..x
 F040  1DF5 FE40 2806 FE60 2802 F1C9 F1CD 781D ...@(..'(.....x.
 F050  C5FD E5FD 2AAD 41FD 36F5 B0FD 77EF 0120 ....*.A.6...w...
```

THE NEXT STAGE WAS TO FIGURE OUT A WAY OF GETTING THIS AREA OF MEMORY INTO A DISK FILE. I READ THE SECTION ON DEBUG IN THE NEWDOS80 MANUAL, BUT IT HAS NO WAY OF STORING AN AREA OF MEMORY TO DISK. I HAVE A COPY OF MON3, WHICH I WAS SURE COULD DO THE JOB, BUT WHEN I WENT TO LOOK FOR THE MANUAL, I REALISED THAT I'D LEFT IT BEHIND IN THE MOVE!

I STARTED READING THROUGH THE NEWDOS80 MANUAL & CAME ACROSS THE DUMP COMMAND. THE FORMAT OF THIS COMMAND IS DUMP,FILENAME,START ADDRESS,END ADDRESS,ENTRY POINT. I USED DUMP 0F000, 0F801, & ENDED UP WITH A FILE ON DISK,. EXAMINING IT WITH SUPERZAP REVEALED A PROBLEM IN THAT THE START ADDRESS & OTHER INFO WAS PLACED AT THE START OF THE FILE, AS SHOWN BELOW.

```
DRV 00  0100 00F0 2197 1922 B341 219F 3722 1640 ....!.".A!.7".@
1   10  212C 3022 0440 CD29 36AF 2100 4422 AD41 !,0".@.)6.!.D".A
1H  20  2648 7723 22A4 4032 E143 CD4D 1BC3 191A &Hw#".@2.C.M....
    30  E33E 1DBC 2003 3E5B BDE3 1803 0018 C5C2 .>.... >↑........
DRS 40  781D CD78 1DF5 FE40 2806 FE60 2802 F1C9 x..x...@(..'(...
95  50  F1CD 781D C5FD E5FD 2AAD 41FD 36F5 B0FD ..x.....*.A.6...
```

A THOUGHT WENT THROUGH MY MIND TO WRITE A PROGRAM THAT WOULD READ THE CONTENTS OF THE FIRST FILE, DISCARD THE FIRST FOUR BYTES & COPY THE REST ACROSS, HOWEVER, I NEEDED A QUICK FIX, (READ A KLUDGE), & SO I COPIED THE ROM IMAGE FROM THE DSK FILE I WAS USING, TO A DIRECTORY IN MY TRS80 FOLDER ON MY WINDOWS SYSTEM. I NEXT LOADED THIS FILE INTO PSPAD'S HEX EDITOR, & HIGHLIGHTING ALL THE BYTES IN THE FILE, FROM THE FIFTH ONE ONWARDS,



	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF
000	0100	00F0	2197	1922	B341	219F	3722	1640	...ð!—."³A!Ÿ7".@
010	212C	3022	0440	CD29	36AF	2100	4422	AD41	!,0".@Í)6¯!.D"-A
020	2648	7723	22A4	4032	E143	CD4D	1BC3	191A	&Hw#"¤@2áCÍM.Ã..
030	E33E	1DBC	2003	3E5B	BDE3	1803	0018	C5C2	ã>.¼ .>[½ã....ÅÂ
040	781D	CD78	1DF5	FE40	2806	FE60	2802	F1C9	x.Íx.õþ@(.þ`(.ñÉ
050	F1CD	781D	C5FD	E5FD	2AAD	41FD	36F5	B0FD	ñÍx.Åýåý*-Aý6õ°ý
060	77EF	0120	0CFD	71B0	FD23	10F9	0100	24FD	wï. .ýq°ý#.ù..$ý
070	71B0	FD23	10F9	ED4B	2640	ED43	B641	0187	q°ý#.ùíK&@íC¶A.‡

I DID A CUT & PASTE, WITH THE PASTE OPERATION STARTING AT THE FIRST BYTE IN THE FILE.



	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF
000	2197	1922	B341	219F	3722	1640	212C	3022	!—."³A!Ÿ7".@!,0"
010	0440	CD29	36AF	2100	4422	AD41	2648	7723	.@Í)6¯!.D"-A&Hw#
020	22A4	4032	E143	CD4D	1BC3	191A	E33E	1DBC	"¤@2áCÍM.Ã..ã>.¼
030	2003	3E5B	BDE3	1803	0018	C5C2	781D	CD78	.>[½ã....ÅÂx.Íx
040	1DF5	FE40	2806	FE60	2802	F1C9	F1CD	781D	.õþ@(.þ`(.ñÉñÍx.
050	C5FD	E5FD	2AAD	41FD	36F5	B0FD	77EF	0120	Åýåý*-Aý6õ°ýwï.
060	0CFD	71B0	FD23	10F9	0100	24FD	71B0	FD23	.ýq°ý#.ù..$ýq°ý#
070	10F9	ED4B	2640	ED43	B641	0187	30ED	4326	.ùíK&@íC¶A.‡0íC&
080	403E	AFFD	E1C1	C9F3	C5D5	E5FD	E52A	AD41	@>¯ýáÁÉóÅÕåýå*-A

I THEN COPIED THE FILE BACK TO THE DSK FILE, & RENAMED IT ESF.ROM, SO THAT THE EMULATOR WOULD RECOGNISE IT, & MOVED THE FILE INTO THE TRS32 DIRECTORY WITH MY MODEL1 ROM IMAGE, & ACTIVATED THE FLOPPY TAPE IN THE EMULATORS CONFIG PANEL. FOLLOWING A REBOOT OF THE EMULATOR, I USED SUPERZAP TO EXAMINE THE MEMORY SPACE WHERE THE ROM IMAGE WAS LOADED, WITH SUCCESS! HOWEVER, PAGING THROUGH THE MEMORY AREA, I DISCOVERED A PROBLEM IN THAT THE EMULATOR HAD ONLY LOADED THE FIRST 1920 BYTES OF THE ACULAB IMAGE. I'VE RECENTLY CONTACTED MATTHEW REED OVER THIS PROBLEM, & I'LL REPORT ANY MORE INFO THROUGH DUSTY.

NOW, I HEAR YOU ASK, WHAT HAS THIS TO DO WITH ME? A SIMPLE IDEA MIGHT BE TO LOCATE A DEBUGGING PROGRAM INTO THIS SPACE, & THEN BY RENAMING IT ESF.ROM, (SAY), IT CAN BE LOADED INTO MEMORY & IS THERE WHEN YOU NEED IT. WITHOUT HAVING TO RE-LOCATE IT IN MEMORY OR ALLOCATE ANY RESERVED MEMORY SPACE ETC. I KNOW THAT THIS HOLE IN THE MEMORY MAP HAS BEEN USED TO STORE A SPECIALIST DRIVER TO DRIVE A BRAILLE PRINTER, SOFTWARE TO DRIVE DEDICATED

HARDWARE & A MODIFIED CHARACTER GENERATOR. ONE FINAL THOUGHT, IF YOU DECIDE TO WRITE YOUR OWN SOFTWARE, REMEMBER THAT YOU'LL HAVE ASSEMBLE THE PROGRAM TO START AT 0000H, AS THE ADDRESS DECODING TAKES PLACE IN HARDWARE. SINCE DOING THIS PROJECT, I'VE BEEN THINKING ABOUT WRITING A SPECIALISED VERSION OF DUMP TO COPY THE MEMORY CONTENTS STRAIGHT TO A FILE. I HOPE THIS PROVIDES INSPIRATION FOR SOME OF YOU TO BE CREATIVE, IF YOU DO, WHY NOT SEND AN ARTICLE TO DUSTY.

---== o0o ==---

HxC Floppy Emulator Project

© 2006 – 2010 HxC2001

(General specifications version 1.2-22/08/10)

Description:

3", 3"1/2 and 5"1/4 Floppy disk drive emulator based on SDCard.



Supported/tested computers:

Atari ST,
Amiga,
Amstrad CPC6128(+),
MSX2,
Oric + MicroDisk,
SuperWildcard,
Synthésiseur Korg DSS-1,
PC,
And many more…
The complete list can be found at this address :
http://hxc2001.free.fr/floppy_drive_emulator/index.html#FILESSUPPORTED

Floppy images files that can be used with this emulator:



The HxC Floppy Emulator allows to convert your floppy images files for the emulator.
Below is the actual list of supported file images type:
ST, STT, MSA, ADF, ADZ, DMS, CPC DSK (standard and extended), CopyQM DSK, Oric DSK, IMG, SMC, MSX DSK, IMD, Zx Spectrum DSK, BetaDisk TRD, BetaDisk SCL, Thomson TO8D SAP,
Thomson TO8D FD, Dragon 64 VDK, Sam Coupé MGT,
Sam Coupé SAD, TI99 PC99, Apricot disk image, PC88
D88, x68000 HDM (beta), x68000 XDF (beta),
Ensoniq EPS Keyboard EDE, Ensoniq Mirage Sampler EDM, Oberheim DPX1 Sampler DPX, Emax & Emax II Sampler *.EM1 and *.EM2
custom raw floppy geometry…

An AmigaDOS or FAT12 floppy disk image can also be generated from spares files.
(Note : New files/computers support is regularly added. Please checkout the project website for the lastest update : http://hxc2001.free.fr/floppy_drive_emulator/)

Some details :

Floppy interface:

HE10 34 pins floppy connector :
 Shugart compatible mode supported.
 PC compatible mode supported.
Tested on PC, Atari ST, Amiga, Amstrad CPC, Thomson, MSX, ZX Spectrum series computers... (complete list here : http://hxc2001.free.fr/floppy_drive_emulator/index.html#FILESSUPPORTED)
 Two floppy disk drives emulation.

Power supply:
 5V +/- 10% standard power floppy connector input.

User Interface:

 3 LEDs ("Power LED","Floppy access LED"," SDCard access LED ").
 3 buttons ("Next","Select/Eject","Previous").
 1 audio transducer. (Head Step and User interface sound).
 2*16 chars Alphanumerical LCD.
(Note : LCD and buttons can be put on an external front panel)
 On Screen display software available for Atari ST and CPC computers.
(Amiga version planned)

SDCard support:

SD Card up to 2GB.
 SDHC Card supported up to 32GB.
(10Mhz SPI bus mode. Average byterate: ~500KB/s)

SDCard Filesystem :

 FAT32 supported. Subdirectory and long name file supported.

Read / Write support:

Track mode based floppy emulator (Full track pre-encoded in the HFE image file)
 Read support: Most of existing formats (FM/MFM/Amiga track...) supported**.**
Custom tracks supported.
 Write support: ISO MFM 256/512/1024Bytes sector write supported**.**
Amiga Write support since the PCB revision C.

Floppy bitrate supported:

 250/300Kbits/s (DD floppies)
 500Kbits/s (HD floppies)
(others bitrates possible)

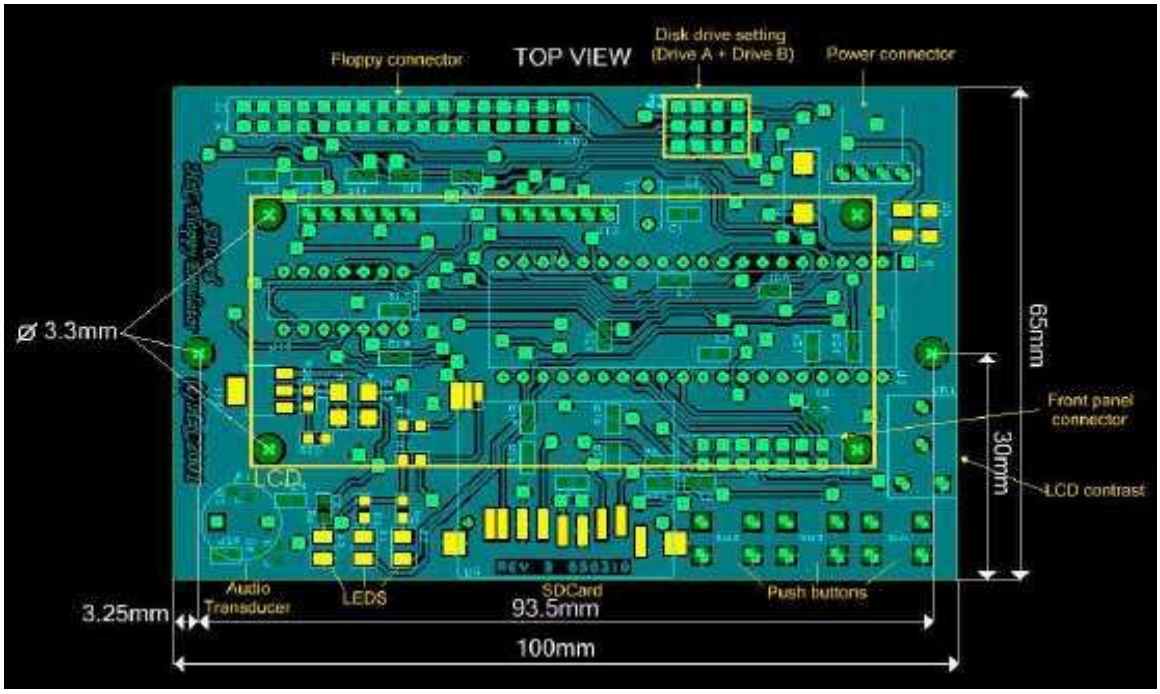
(Note: Variable bitrate not supported for the moment. So protected floppy disk image (IPF and STX file format) file support will be only partial ! If you look for a device supporting IPF / STX please a have a look to the USB HxC Floppy Emulator device)

RPM:

 300 RPM , 360 RPM supported.
(others RPM possible).

Tracks / Side:
 Up to 255 tracks per floppy.
 1 or 2 sides.
Additionnal features:
 Firmware update via the SDCard.
 Last Loaded Floppy Image autostart at power up.
 Fast floppy image loading (<<1second), no conversion time.

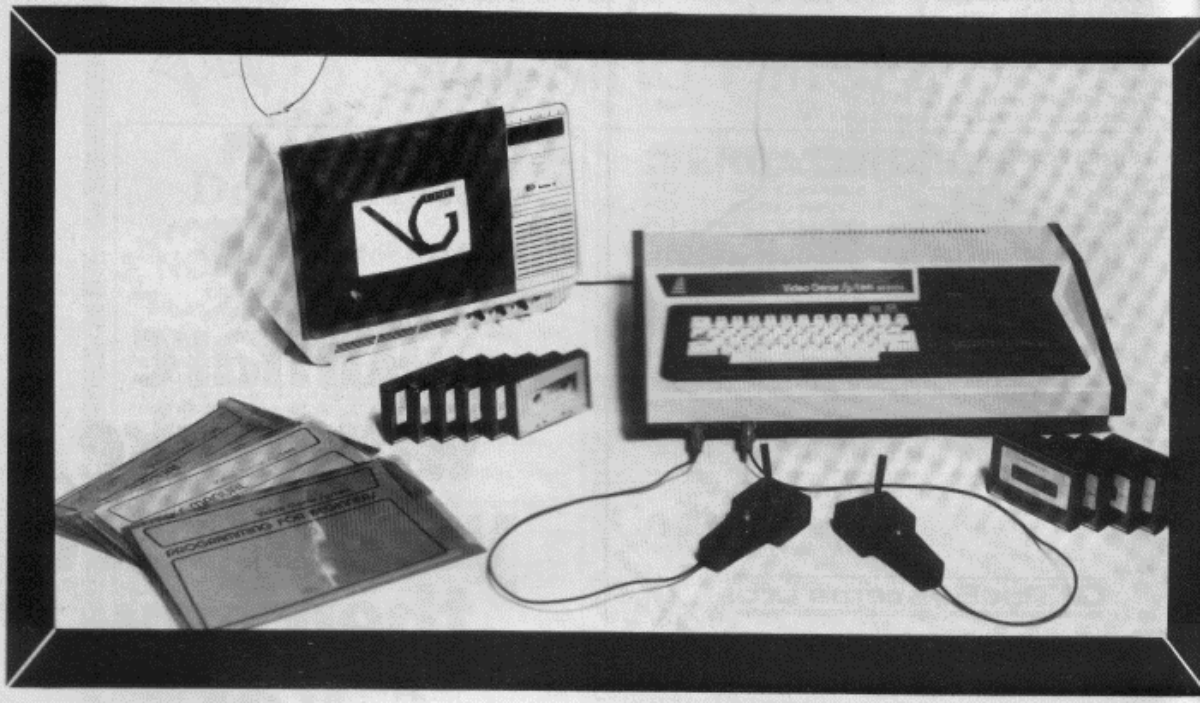
Mechanical drawing:



For more information, software update and firmware update :
Project website : http://hxc2001.free.fr/floppy_drive_emulator
Project Forum : http://www.torlus.com/floppy/forum
Official seller : http://www.mmj.pl/~lotharek/atari/
Some youtube videos showing the device :
http://www.youtube.com/watch?v=MzfUZorBayw
http://www.youtube.com/watch?v=g6dBTyfPrkg
http://www.youtube.com/watch?v=xeJvNCPMU-c

ComServe

COMPUTER SHOP *PRESENTS*



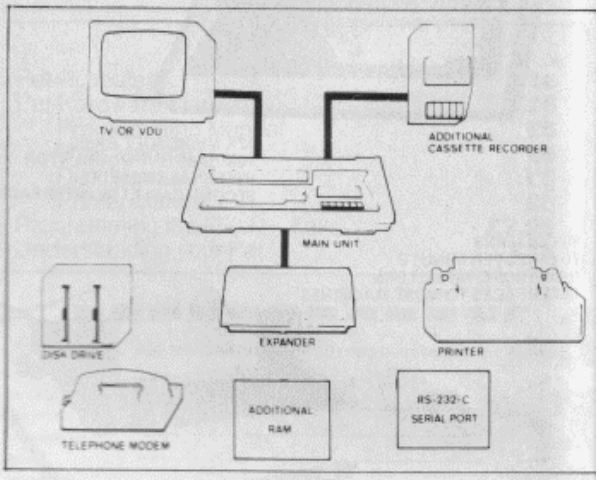
THE ELEGANT, EXPANDABLE

video genie system

£364 inc VAT. Postage. Requires T.V. or monitor — £70 if bought with Genie

Comprises standard 16K computer. 3 manuals. Demonstration tape. Lead for additional cassette player. Lead for monitor.

FREE Standard cover. Head cleaning/demangetising tape.
FREE Programs: Games. Utilities. Subroutines.
FREE Standard soundkit if wanted (installation £5 extra)
FREE Comserve joysticks if wanted (installation £7 extra)
FREE Coloured plastic folders for manuals or screen overlays.



CWO to

ComServe

98 TAVISTOCK STREET, BEDFORD, BEDFORDSHIRE
TELEPHONE (0234) 216749

We supply various printers.
We will supply S100 expansion box, colour boards, disk drive, fast tape drives, RS232 interfaces as available.
We supply some '80, Genie and CP/M programs at 20% discount.
We buy some secondhand programs (in manufacturers' original pack and with original documentation only)

S.A.E. for further information. Items and prices are as at time of going to press and are subject to alteration.

TRS8BIT

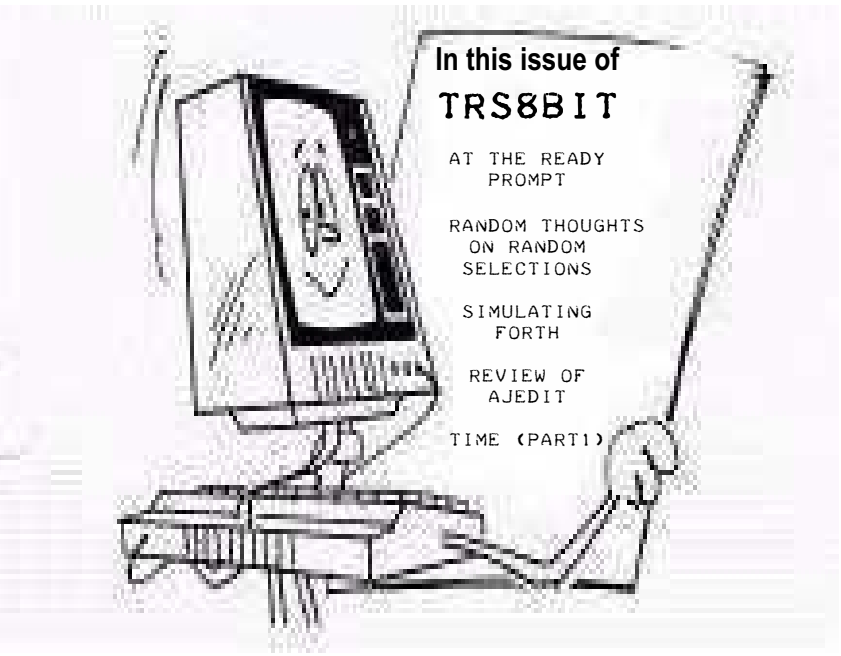


HELLO, AND WELCOME TO THIS, THE FIRST EDITION OF 2011. IT IS THE START OF OUR 5TH YEAR, BUT EVEN SO, I HOPE I'VE STILL MANAGED TO FIND BITS WITH 'TANDY' INTEREST FOR YOU ALL TO ENJOY AND EXPERIMENT WITH.

IT ALL SEEMS TO HAVE GONE VERY QUIET ON THE TANDY FRONT OVER THE LAST FEW MONTHS. EVEN ON EBAY, IF YOU EXCLUDE THE 'BUY IT NOW' MERCHANTS, THERE'S NOT BEEN MUCH UP FOR GRABS. A MODEL 1 FETCHED JUST UNDER £35, AS DID A MODEL 100. A FAULTY MODEL 4 DIDN'T EVEN MANAGE THAT MUCH!

THE WEBSITE IS STILL ATTRACTING VISITORS, I THINK OFFERING TRS8BIT FOR SALE ON EBAY, AS AN ADVERTISING PLOY, HAS HELPED.
I HAD A VERY COMPLIMENTARY EMAIL ABOUT THE ONE-LINER PLANETS PROGRAM IN THE LAST ISSUE. PLEASE KEEP IN TOUCH THOUGH, AND LET ME KNOW IF THERE IS ANY TYPE OF ARTICLE YOU WOULD LIKE TO SEE FEATURED. YOU COULD EVEN SEND IN AN ARTICLE FOR PUBLICATION, IT WOULD BE MUCH APPRECIATED.

IN THIS ISSUE, 'AT THE READY PROMPT' MAKES A WELCOME RETURN. JUST A FEW SNIPPETS OF INTERESTING IDEAS FOR YOU.
YOU CAN FIND OUT WHICH ARE THE 5 MOST 'POPULAR' LOTTERY NUMBERS (AND WHO KNOWS, WIN YOURSELF A MILLION PERHAPS).
IF YOU FANCY HAVING A GO AT FORTH THERE'S A REPRINT OF AN ARTICLE BY OWEN BISHOP FROM THE JUNE 1□82 EDITION OF COMPUTING TODAY, FOR YOU TO TRY. IT ONLY NEED A 16K MODEL 1 TOO!
HAVING COME ACROSS MANY ADVERTS FOR HARDING'S AJEDIT PROGRAM, I FOUND A DETAILED REVIEW BY KEN GREY, FIRST PUBLISHED IN NATGUG NEWS.
AND FINALLY, PETER STONE HAS PRODUCED PART ONE OF A SERIES OF ARTICLES ON DISPLAYING DIFFERENT TYPES OF CLOCKS.



HERE WE ARE IN 2011, AND THERE'S STILL ORIGINAL AND FACINATING ARTICLES FOR A MODEL 1 TANDY APPEARING!
BRILLIANT

WELL, THAT ABOUT WRAPS UP THIS ISSUE. I HOPE YOU FIND IT OF INTEREST.
THERE'S A COUPLE OF VERY RARE ADS. FOR YOU GENIE FANS!



BYE FOR NOW

DUSTY

AT THE
READY>
PROMPT



A COUPLE OF SEIKOSHA/TANDY LPVII CLONES RECENTLY CAUGHT MY EYE IN VARIOUS OLD ADVERTISEMENT. THE FIRST WAS IN THE GUISE OF THE COMMODORE "VIC 20 DOT MATRIX PRINTER" (WITH AN OPTIONAL SERIAL RS232 INTERFACE) AND THE SECOND HAD A BBC LOGO EMBLAZONED UPON IT AND WAS CALLED "ACORN GP80A PRINTER". PERHAPS FURTHER POSSIBLE SOURCES FOR RIBBON CARTRIDGES. SOMETHING ELSE I'LL KEEP MY EYE OUT FOR ON EBAY!

SCRIPSIT UNDER NEWDOS 80V2 ONE OF THE MANY FRUSTRATIONS EXPERIENCED WHEN USING SCRIPSIT IS IF YOU SUDDENLY NEEDED TO SAVE TEXT WITHOUT KNOWING WHICH DISK HAD SUFFICIENT FREE SPACE! NOW, THANKS TO BYTE 6/82 I HAVE FOUND THAT SCRIPSIT, UNDER ND80, YOU CAN USE THE 'DFG' KEYS TO CALL UP THE MINI-DOS AND THEN DO A 'FREE' OR A 'DIP'. TYPE MDRET AND YOU ARE BACK WITH YOUR SCRIPSIT PROGRAM.

AS ALWAYS, I'M KEEN TO TRY ANYTHING TO SAVE A COUPLE OF BYTES.
HERE'S, YET ANOTHER, SIMPLE LITTLE BASIC IDEA TO DO JUST THAT.

IF YOU WISH TO INCREASE THE VARIABLE P, BY ONE, EACH TIME YOU LOOP THROUGH A SUBROUTINE BUT DO NOT WISH THE VALUE TO EXCEED THE VALUE OF VARIABLE M, THEN INSTEAD OF

```
10 P=0:M=10
20 P=P+1
30 IF P>M THEN P=M
40 PRINT P;
50 GOTO 20
```

TRY

```
10 P=0:M=10
20 P=P-(P<M)
30 PRINT P;
40 GOTO 20
```

EVERY LITTLE HELPS!

AND TO FINISH WITH, HERE'S A LITTLE PUZZLE TO GET THE 'OLD GREY MATTER' RUNNING. A CERTAIN NUMBER ENDS WITH THE DIGIT 6. WHEN THE 6 IS TAKEN FROM THE END OF THE NUMBER AND PLACED AT THE BEGINNING, A NEW NUMBER IS FORMED WHICH IS 6 TIMES THE ORIGINAL NUMBER. WHAT IS THE ORIGINAL NUMBER. TO BE FAIR, 6 IS THE EASIEST NUMBER TO FIND (SO I AM TOLD, ANY ONE ANY IDEAS WHY?). IF YOU DON'T BELIEVE ME, TRY IT WITH 2,3,4,5,7,8 OR □!

ComServe
computer shop



"HOBBYIST" GENIE
VIDEO GENIE with 16K RAM. ●3 manuals. ● Leads ● demonstration tape ● cover ● head cleaner ●demagnetiser ● 2 joysticks ● sound kit with speaker installed ● demonstration games tape for sound and joysticks and graphics. 1 year guarantee.
£369.00 inc VAT and carriage

ACULAB SYSTEM
VIDEO GENIE with 32K RAM ● 5 manuals ● leads ● demonstration tape ● cover ● cleaning tape ● Aculab drive ● power supply ● 11 cartridges ● extended basic with disk like commands ● compress, merge and renumber ● 12" monitor ● expansion lead.
£666.00 inc. VAT and carriage





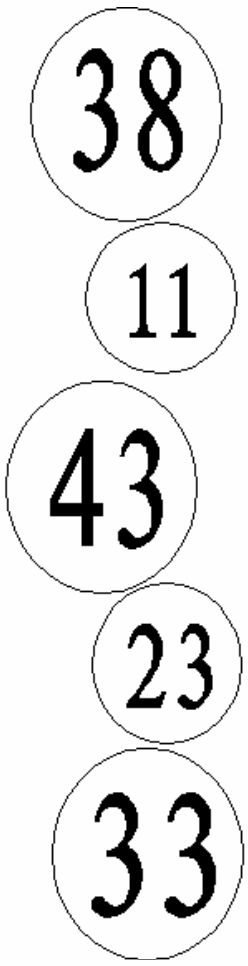
DISK SYSTEM
GENIE with expansion box, 48K RAM ● 5 manuals ● twin 40 track drives ● 35 track compatible ● cover ● cleaning tape ● cleaning disk ● 10 diskettes ● Newdos 80 operating system ● 12" monitor.
£1190.00 inc. VAT and carriage

CENTRONICS 737-2	EPSON MX80FT	GP 80
£399	£430.	£230
inc VAT, cable & carriage. Proportional spaced printer.	inc. VAT & carriage. Quality printer with graphics	inc. VAT & carriage Hobby printer with Hi-Res graphics

ComServe
98 TAVISTOCK STREET, BEDFORD, BEDFORDSHIRE
TELEPHONE (0234) 216749

RANDOM THOUGHTS ON RANDOM SELECTIONS

LIKE MANY THOUSANDS OF OTHER 'LUCKY' PEOPLE, I STARTED PLAYING THE NATIONAL LOTTERY EVERY SATURDAY EVENING WHEN IT FIRST STARTED, ALL THOSE YEARS AGO, AND BECAUSE I'VE ALWAYS DONE IT, I'VE CONTINUED TO DO IT! IT'S JUST THAT THOUGHT AT THE BACK OF YOUR MIND, "IF I STOP THIS WEEK, I'LL BET THE NUMBERS WILL COME UP". THEY NEVER HAVE THOUGH, AND I DON'T SUPPOSE THEY EVER WILL IN REALITY. HOWEVER, IT DOES MAKE FOR A BIT OF EXCITEMENT ON A SATURDAY EVENING.



HOWEVER, ONE OF THE THINGS I'VE QUITE OFTEN REGRETTED IS THAT I NEVER BOTHERED TO KEEP A RECORD OF ALL THE NUMBERS DRAWN AND THE ORDER IN WHICH THEY APPEARED ETC. ETC. I ALWAYS FELT THAT IT WOULD BE NICE TO KNOW WHICH WAS THE MOST-DRAWN NUMBER AND JUST HOW MANY TIMES THOSE 'LUCKY' NUMBERS OF MINE HAS APPEARED.

THINKING BACK TO PETER STONE'S ARTICLE ON RANDOM NUMBERS A COUPLE OF ISSUES AGO, I BECAME FIRED-UP WITH ENTHUSIASM AND DECIDED TO CHECK UP AND FIND OUT JUST WHAT WERE THE MOST POPULAR NUMBERS. I ASSUMED THAT THE 'RAW' INFORMATION WOULD BE HELD AT THE LOTTERY H.Q. WEBSITE, AND AFTER DOWN LOADING IT, I'D WRITE A BASIC PROGRAM ON MY 16K MODEL 1 AND CREATE SOME STATISTICS.

WELL, NO DOUBT YOU'LL NOT BE SURPRISED TO LEARN, I'VE BEEN BEATEN TO IT! I CAME ACROSS THE WEBSITE

HTTP://
LOTTERY.MERSEYWORLD.COM

AND FOUND THAT ALL THE HARD WORK HAD BEEN DONE FOR ME.
(IT NEVER CEASES TO AMAZE ME, JUST WHAT IS AVAILABLE ON THE INTERNET IF YOU LOOK HARD ENOUGH!)

AND JUST IN CASE YOU TOO ARE INTERESTED AS TO WHAT ARE THE MOST POPULAR NUMBERS. WELL, AT THE TIME OF PENNING THESE FEW NOTES, THE NUMBER 38 HAS APPEARED 220 TIMES, NUMBERS 11 & 43 HAVE APPEARED 215 TIMES AND NUMBERS 23 & 33 HAVE APPEARED 214 TIMES.

OH, AND BY THE WAY, THE NUMBER WHICH HAS BEEN PICKED THE FEWEST NUMBER OF TIMES IS 20. IT HAS ONLY MANAGED 160 APPEARANCES!

--== o0o ==--

SIMULATING FORTH

Owen Bishop

If you like the look of FORTH as a programming language but are not yet convinced it is the right one for you, why not try out our simulator? Written in BASIC, it will allow you to produce simple programs with ease.

Having been impressed by D S Peckett's series on GOING FORTH in recent issues of Computing Today, I felt that this language had a lot to offer; indeed I wondered if it might be more suitable than BASIC for many of my current projects. I resolved to investigate the language further.

However, the prices of even the simplest versions available in the UK or from the USA are not cheap. Undoubtedly the prices are fair and reasonable but, before spending such an amount, I wanted to be certain that FORTH would suit my requirement fully. More significantly, I wondered if *I* would suit FORTH's requirements. Two points left me in doubt:

1) Would I be able to work easily in Reverse Polish Notation?

2) Would I find it easy to handle the stack?

Both these features are unfamiliar to users of BASIC.

FORTH, Right!

I tried writing a few lines of FORTH and then doing a dry run on paper. I found I was continually scribbling long columns of figures to show what the stack should hold at each stage, often forgetting to delete items already absorbed in a computation or printed out of the stack. The obvious solution to all this tedium was a program in BASIC to do what FORTH would do — the result is this FORTH simulator program.

Make no mistake though, this is by no means a substitute for a true FORTH. It runs only one line at a time and it is ***terribly*** slow. The latter feature distinguishes it at once from FORTH, which is fantastically fast (one of the main reasons that I want to use it). Also, it does not do as much as FORTH.

Despite the above reservations, many readers may find this BASIC version a useful way of 'tasting and testing' the concepts of RPN and the use of a stack-oriented language. Like FORTH, it has a dictionary and you can define new words. Also like FORTH, it is threaded, in that you can define a new word using words you have already defined. One special feature of the simulator is that it displays the stack at each stage of the operation of the program line. It does all that 'scribbling on paper' and does not make mistakes. Here, the slow running of the BASIC version is an asset, since it goes at just the right speed to let you follow what is happening as each stage is executed.

Specification

Dictionary: Holds up to 20 words, each of which can have up to 10 terms to define it. If you want more words or longer definitions, you can easily alter the DIMAW statement in line 20. If you re-define a word, the program uses only the most recent definition and ignores earlier ones. One problem with the TRS-80 is that the colon acts as a delimiter in INPUT lines, so we can not use this to introduce the 'colon definition' as in FORTH. The hash sign is used instead. Thus we can define SQUARE by the line:

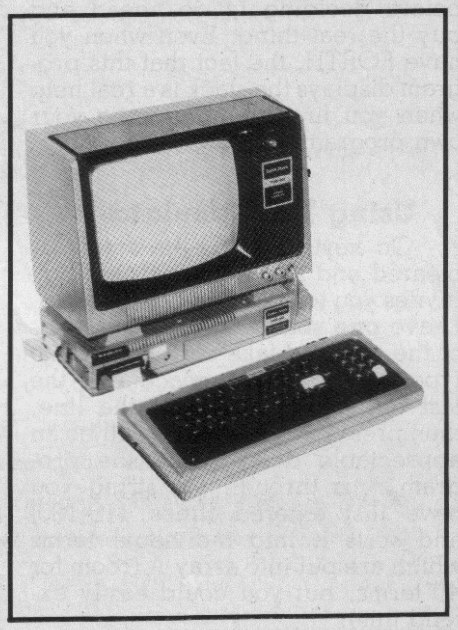
```
# SQUARE DUP *
```

The terminal semicolon is not required. Apart from this, the simulator uses exactly the same operators and commands as FORTH.

Stack: Holds up to 10 values and displays all 10 in a vertical column on the screen. The screen holds up to seven print-outs of the stack. These are printed with the tops of stack level; I prefer to think of a stack in this way, with new items pushing the others down and the stack rising when top of stack (TOS) is popped. If a line requires more than seven printings of stack, execution is halted to give you a chance to look at the display. You can then instruct it to continue; printing of stacks begins again from the left of the screen replacing the lists already there. The length of stack can be increased by altering the DIMS statement in line 10. Note that the simulator has only one stack.

To keep the program short and simple there are few error-trapping routines so it is important not to make mistakes when entering lines. However, there is a 'STACK EMPTY!' message.

Arithmetic operators: The program works with positive and



negative integers and has the complete set of operators: +, -, / and *.

Stack operators: DUP, SWAP, OVER, ROT and DROP.

Constants: You can define up to 21 of these (more if you alter DIMAC). Do not put constant definitions in a line with other instructions. The format is the standard one:

<value> CONSTANT <name> . . .
then press 'ENTER'

Variables: You can define up to 21 variables (array AV) by the format:

<value> VARIABLE <name> . . .
then press 'ENTER'

When the variable is used in a program, its address is put on top of stack followed by the letter 'V' to remind you that this is an address. The addresses are not addresses in RAM, but in the array AV, running from 0 to 20. The operator '!' (store) is usable in connection with variables.

Conditional operators: < > and =.

Conditional loops: There is one format:

<condition> IF <action 1> ELSE <action 2> THEN <continue>

The ELSE is ***not*** optional as in FORTH. If you have no 'action 2' you will need to put a dummy action in its place (eg DUP DROP). Nested loops are not provided for.

Finite loop: There is one format: <upper limit> <start point> DO <action> LOOP <continue> When in a loop, you can obtain the value of the index by using 'I'. Nested loops are not provided for.

The above is a small subset of FORTH, but should be enough for you to try your hand at the language before deciding to go ahead and buy the real thing. Even when you have FORTH, the fact that this program displays the stack is a real help when you first begin to write your own programs.

Using The Simulator

On keying RUN, the screen is cleared and the displayed message invites you to enter a line of FORTH. Leave **one** space between all terms of the line and take care not to mistype (do not type a space after the last term). Double-check the line, then press 'ENTER'. There will be an appreciable delay while the program runs through the string you have just entered (lines 110-180) and sorts it into individual terms which are put into array A (room for 40 terms, but you could easily extend this).

Next the program will ask for 'Command'. If you want to discard the line just entered, clear the stack and start again, type 'Q' (quit). If you want to step through the line and have the stack displayed at each step, type 'S' for 'step'. Otherwise, type any other letter and the line will be excecuted as a whole, displaying the stack only at the end.

if there are more than seven steps and the screen is full, execution waits until you press any key.

When the print command (.) is used, the FORTH print-out appears on the bottom line of the screen. Subsequent print-outs will appear on the same line, to the right. This is not the same format as in FORTH, but has the advantage that at the end of execution you can see your original line, the states of the stack, the FORTH responses ('OK', etc.) and print-outs, all on the screen at once.

When execution is complete, the program waits (line 330) for a keypress. If you press 'R' the program is re-run and you lose all defined words, constants, variables and the stack. If you press 'C', however, you retain everything but the stack, which is cleared. If you press any other key, you simply return to enter a new line with everything intact. This allows you to go on from line to line when trying a more extensive program.

VARIABLES

A$ Array of terms of FORTH program line.
AA$ A character taken from A$.
AC$ Array of CONSTANTS.
AS$ Term to be put on top of the stack.
AV$ Array of VARIABLES.
AW$ Array of words (DICTIONARY).
B$ Input commands.
C Number of stack displays on screen.
D Number of responses printed at bottom of screen.
DF 'DO . . . LOOP' index at finish.
DJ Value of J for 'DO'.
DS 'DO . . . LOOP' index at start.
FF Flag for function found; = 0 if not found, = 1 when found. Used in execution to indicate than an operation has been identified and executed.
FS Flag for single-step; = 1 for single-step.
FT Flag in 'IF . . ELSE . . THEN' loop; = 0 if condition is false, = 1 if true (ie FT = TOS)
FW Flag for word found; = 1 when search of dictionary has found a word.
J The number of the term currently being executed.
J1 Temporary register for J in 'IF . . . ELSE . . . THEN' loops.
K Loop counter.
LA Number of characters in input string A$.
N Array of number of terms in each word in AW$.
NC Number of most recently defined constant defined in AC$ (0 to 20).
NV As above, for variables in AV$.
NW As above, for words, except that words are listed from location 20 downwards (20 to 0).
S$ Stack; S$(0) is TOS.
T Number of terms in a line.
VA ASCII value of AA$.

Program Listing

```
 10  REM ** FORTH SIMULATOR IN BASIC
 20  CLEAR 2500:DEFSTR A,B,S:DEFINT C-R,T-Z:
     DIM A(40):DIM AC(20,1):DIM AV(20,1):DIM AW(20,10):
     NW=20:DIM N(10)
 30  REM ** CLEAR THE STACK
 40  FOR J=0 TO 9
 50  S(J)=" ."
 60  NEXT J
 70  CLS:PRINT "ENTER LINE OF FORTH PROGRAM"
 80  FOR J=0 TO 30:A(J)="":NEXT J:T=0:C=0:D=0
 90  INPUT A$
100  REM ** SEPARATE TERMS INTO LIST
110  LA=LEN(A$)
120  FOR L=1 TO LA
130  AA=MID$(A,L,1)
140  VA=ASC(MID$(A,L,1))
150  IF VA>47 AND VA<58 OR VA>64 AND VA<91 THEN
     A(T)=A(T)+AA
160  IF VA=32 THEN T=T+1
170  IF VA>41 AND VA<48 OR VA=33 OR VA=35 OR VA>59 AND
     VA<63 THEN A(T)=AA
180  NEXT L
190  REM ** READY TO PROCESS THE LINE
200  B=""
210  INPUT "COMMAND";B
220  IF B="Q" THEN 40
230  IF B="S" THEN FS=1
240  REM ** PROCESSING NEW WORDS, CONSTANTS AND
     VARIABLES
250  IF A(0)="#" THEN N(NW)=T-1:FOR K=1 TO T:
     AW(NW,K)=A(K):NEXT K:NW=NW-1:GOTO 70
260  IF A(1)="CONSTANT" THEN AC(NC,0)=A(2):
     AC(NC,1)="[SPC]"+A(0):NC=NC+1:GOTO 70
270  IF A(1)="VARIABLE" THEN AV(NV,0)=A(2):
     AV(NV,1)="[SPC]"+A(0):NV=NV+1:GOTO 70
280  J=0
290  GOSUB 1000
300  IF T>J THEN J=J+1:GOTO 290
310  IF FF=1 OR FW=1 THEN PRINT@960+D*8,"OK";:D=D+1
320  B=INKEY$:IF B="" THEN 320:REM ** WAITING AFTER
     PROCESSING AN INPUT
330  IF B="R" THEN RUN
340  IF B="C" THEN 30
350  GOTO 70
1000 FF=0:FW=0:REM ** EXECUTING NORMAL LINES
1010 IF A(J)="IF" THEN FT=VAL(S(0)):J=J+1:AS=S(1):
     GOSUB 4000:IF FT=0 THEN GOSUB 2000
1020 IF A(J)="ELSE" AND FT=1 THEN J=J+1:GOSUB 2500
1030 IF A(J)="THEN" THEN J=J+1
1040 IF A(J)="DO" THEN DS=VAL(S(0)):DF=VAL(S(1)):
     AS=S(1):GOSUB 4000:AS=S(1):GOSUB 4000:DJ=J:RETURN
1050 IF A(J)="LOOP" AND DS<DF-1 THEN J=DJ:DS=DS+1:FF=1:
     RETURN
1060 IF A(J)="LOOP" AND DS=DF-1 THEN FF=1
1070 IF A(J)="I" THEN AS=STR$(DS):GOSUB 4000
1080 IF ASC(LEFT$(A(J),1))>47 AND ASC(LEFT$(A(J),1))<58
     THEN AS="[SPC]"+A(J):GOSUB 3000
1090 IF ASC(LEFT$(A(J),1))=45 AND LEN(A(J))>1 THEN
     AS=A(J):GOSUB 3000
1100 GOSUB 7000
1110 GOSUB 8000
1120 GOSUB 9000
1130 IF FW=1 THEN RETURN
1140 IF S(0)=" ." AND FF=0 THEN 6000
```

SIMULATING FORTH

```
1150  IF A(J)="." THEN PRINT@960+D*8,S(0);" OK";:D=D+1:
      AS=S(1):GOSUB 4000
1160  IF A(J)="DUP" THEN AS=S(0):GOSUB 3000
1170  IF A(J)="DROP" THEN AS=S(1):GOSUB 4000
1180  IF S(1)=" ." AND FF=0 THEN 6000
1190  IF A(J)="+" THEN AS=STR$(VAL(S(0))+VAL(S(1))):
      GOSUB 4000
1200  IF A(J)="-" THEN AS=STR$(VAL(S(0))-VAL(S(1))):
      GOSUB 4000
1210  IF A(J)="*" THEN AS=STR$(VAL(S(0))*VAL(S(1))):
      GOSUB 4000
1220  IF A(J)="/" THEN AS=STR$(INT(VAL(S(0))/VAL(S(1)))):
      GOSUB 4000
1230  IF A(J)="<" THEN AS=" 0":IF VAL(S(0))>VAL(S(1))
      THEN AS=" 1":GOSUB 4000
1240  IF A(J)="<" THEN GOSUB 4000
1250  IF A(J)=">" THEN AS=" 0":IF VAL(S(0))<VAL(S(1))
      THEN AS=" 1":GOSUB 4000
1260  IF A(J)=">" THEN GOSUB 4000
1270  IF A(J)="=" THEN AS=" 0":IF VAL(S(0))=VAL(S(1))
      THEN AS=" 1":GOSUB 4000
1280  IF A(J)="=" THEN GOSUB 4000
1290  IF A(J)="SWAP" THEN S=S(0):S(0)=S(1):S(1)=S:FF=1
1300  IF A(J)="OVER" THEN AS=S(1):GOSUB 3000
1310  IF A(J)="!" THEN AV(VAL(S(0)),1)=S(1):AS=S(1):
      GOSUB 4000:AS=S(1):GOSUB 4000
1320  IF S(2)=" ." AND FF=0 THEN 6000
1330  IF A(J)="ROT" THEN S=S(2):S(2)=S(1):S(1)=S(0):
      S(0)=S:FF=1
1340  IF FF=0 THEN PRINT@960+D*8,A(J);"?";:D=D+1
1350  IF FS=0 AND J<T THEN RETURN
1360  FOR K=0 TO 9:REM ** PRINT STACK
1370  PRINT@256+K*64+C*8,"[8 SPC]";
1380  PRINT@256+K*64+C*8,S(K);
1390  NEXT K
1400  C=C+1:IF C>7 THEN C=0:GOSUB 5000
1410  RETURN
2000  J1=J+1:REM ** EXECUTING 'IF'
2010  IF A(J1)="ELSE" THEN J=J1+1:RETURN
2020  J1=J1+1:IF J1>T THEN PRINT@960+D*8,"NO IF!";:D=D+1
2030  GOTO 2010
2500  J1=J+1:REM ** EXECUTING 'ELSE'
2510  IF A(J1)="THEN" THEN J=J1+1:RETURN
2520  J1=J1+1:IF J1>T THEN PRINT@960+D*8,"NO ELSE!";:
      D=D+1
2530  GOTO 2510
3000  FOR K=9 TO 1 STEP-1:REM ** NEW TOS AND MOVE STACK
      DOWN
3010  S(K)=S(K-1)
3020  NEXT K
3030  S(0)=AS
3040  FF=1
3050  RETURN
4000  S(0)=AS:REM ** NEW TOS AND MOVE STACK UP
4010  FOR K=1 TO 8
4020  S(K)=S(K+1)
4030  NEXT K
4040  S(9)=" ."
4050  FF=1
4060  RETURN
5000  IF INKEY$="" THEN 5000:REM ** SCREEN IS FULL SO WAIT
      TO DISPLAY NEXT STACK
5010  RETURN
6000  PRINT@960+D*8,"STACK EMPTY!";:REM ** EMPTY STACK
      ERROR
6010  GOTO 320
7000  FOR K=0 TO 20:REM ** SEARCHING DICTIONARY
7010  IF A(J)=AW(K,1) THEN T=T+N(K):FW=1:FOR I=T TO
      J+N(K)+1 STEP-1:A(I)=A(I-N(K)):NEXT I:FOR I=1 TO
      N(K):A(J+I)=AW(K,I+1):NEXT I:RETURN
7020  NEXT K
7030  RETURN
8000  AS="AS":REM ** LOOKING FOR A CONSTANT
8010  FOR K=0 TO NC
8020  IF A(J)=AC(K,0) THEN AS=AC(K,1)
8030  NEXT K
8040  IF AS<>"AS" THEN GOSUB 3000
8050  RETURN
9000  AS="AS":REM ** FINDING THE ADDRESS OF A VARIABLE
9010  FOR K=0 TO NV
9020  IF A(J)=AV(K,0) THEN AS=STR$(K)+"V"
9030  NEXT K
9040  IF AS<>"AS" THEN GOSUB 3000
9050  RETURN
```


SPECIAL INTRODUCTORY PRICE – SAVE £10

FOR **TRS-80** USERS

Eprom Programmer

The LM124 EPROM PROGRAMMER when used with the Tandy TRS-80 Model I 16K Level II Microcomputer will program the following single rail eproms:

Intel	**2758, 2716, 2732**
Texas	**2758, 2508, 2516, 2532**

and other manufacturers compatible types

● The LM124 is a port based system having one IN port and three OUT ports ● It is fitted with a quality Zero Insertion Force socket and has a complete on-board power supply unit ● Completely enclosed in a robust plastics case ● No personality modules required ● All eprom types may be fully programmed in one pass ● Full operating instructions in comprehensive 20-page Users Manual ● Requires Tandy Interface Cable.

VIDEO GENIE and Model III adaptors available soon!

LM124 SOFTWARE FEATURES INCLUDE

● **Powerful Editing Facilities** 256-byte page display, single interpage keystroke, full cursor control, byte and nibble-wise movements, flashing cursor.

● **Compatible with EDTASM** Loads object code assembled by TRS-80 Editor/Assembler EDTASM, uses EDTASM's command syntax.

● **Comprehensive Command List** BURN, COPY, EXIT, FILL, LOAD, MEMORY, NEW, PAGE, READ, SAVE, TEST, ZERO.

● **Comprehensive Error Messages** Programming Error, Verification Error, Checksum Bad, Outside Range, Bad Parameters, Syntax Error, Illegal Command.

● **Plus** Graphical display during Burn, Automatic Verify, Burn Pause/Continue, Hard Copy.

● **SPECIAL INTRODUCTORY PRICE £57.50** including postage and packing. Valid until 12.6.82. The Users Manual is also available separately at a cost of £2.20 including postage.

LANDER MICROSYSTEMS

Telephone Romford 26325

32 Clockhouse Lane, Collier Row, Romford, Essex

VIDEO GENIE and TRS80

THE ROM–VIDEO GENIE & TRS 80 (3 ROM Models)

Are you fed up of loading a lower case drive every time you switch on? — Want your name inside our computer — Better loading (TRS80) — Sick of bouncy keyboards-s!

You need our new ROM—simply remove old—plug in new.

1) Firmware driver for lower case
2) Security code displays–(your name & post code?)–up to 21 characters
3) Improved tape loading
4) Alleviates repeating characters (key-bounce)
5) Checks for feature ROMS

£14.95 + VAT + P&P (80p)

FEATURE ROMS £18.95 + VAT + P&P (80p)
A series of ROMS starting with FEATURE 'A'

1) Single keystroke commands e.g. Shift A = Auto etc.
2) Flashing cursor (can be toggled on/off)
3) Repeating characters–with delay–(toggle)
4) Machine code monitor and editor
5) System load and save for backing up those system Programs –uses no RAM and so can deal with a FULL 16K program

NEW

FEATURE 'B'–extended basic
1) Merge–two basic programs to one
2) Renumber
3) Screen print
4) Various basic tools PLUS

Access and Barclaycard accepted.

Plug on for feature ROMS
Now you can simply plug on the back of your computer, insert ROM and away you go.
VG £29.95 inc. ROM A + VAT + P&P
TRS80 £34.50 + VAT + P&P (80p)

48K RAM–internal
TRS80 & VG £70 + VAT + P&P (80p)
Plug in our modules. Connect three wires (VG) or five (TRS80). You are not required to piggyback chips. Keep your old RAM (or sell it!). Compatible with expansion.

LOWER CASE HARDWARE (VG & TRS80)
Unplug two IC's, plug in two modules, connect three wires–EASY! £19.00 + VAT + P&P

BOOKS
"Disassembled Handbook for TRS80"–A Self Teach–Series of Books–Written by Bob Richardson of New York. Very well received by U.S. Reviewers, **Essential** Reading for TRS80, Video Genie & Radio Hams.
Vol. 1 Decoding the ROM & Calls, etc. **£6.45**
Vol. 2 Using Calls, to shorten programs, etc. **£8.75**
Vol. 3 D/A–AD, Spooling, etc. **£10.50**
Vol. 4 Teletype, Morse Techniques **14.50**
Vol. 5 Voice Systems **TBA**
+ Much else in each volume.
Various Books–Lists on application.
+ VG–Expansion–Disk Drives–Printers, etc. Details on application.

Try This:
M x 80T
£285 + CARR. + VAT (Cash/Chq. only)
M x 80 FT II
£345 + CARR. + VAT (Cash/Chq. only)

LNW PCB's & Full Manuals
High Res. Colour Computer
£108 (+ CARR. + VAT) Inc. Level 2 & Colour Roms.
Expansion (RS232-MEM-PRINTER-DISK-CLOCK) **£65 (+ CARR. + VAT)**

TANTEL Prestel Adaptor (colour)
£160 + CARR. + VAT

General Northern Microcomputers [GNOMIC]

46 Middle Street, Blackhall, Hartlepool, Cleveland. Tel. Peterlee (0783) 863871

A REVIEW OF AJEDIT

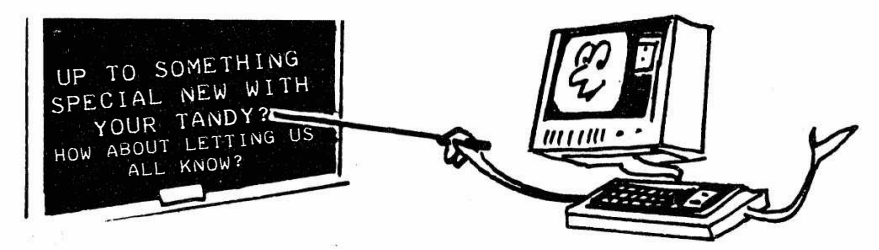
KEN GREY

BEFORE I REVIEW AJEDIT, I WOULD LIKE TO SAY SOMETHING ON THE SUBJECT OF REVIEWING PROGRAMS. IT IS NOT AN EASY TASK TO REVIEW A MAJOR PIECE OF SOFTWARE SUCH AS AJEDIT AND BE CERTAIN THAT YOU HAVE COVERED EVERY ASPECT, FOUND ANY BUGS (IF THEY EXIST) AND HIGHLIGHTED ALL THE INTERESTING OR SPECIAL POINTS. THERE ARE A NUMBER OF REASONS FOR THIS, FIRSTLY YOU MUST BE VERY WELL ACQUAINTED WITH THE ACTUAL PROGRAM BY HAVING USED IT FOR SOME TIME TO APPRECIATE THE VARIOUS POINTS. SECONDLY, YOU CAN NOT HELP FORMING PERSONAL OPINIONS BASED ON WHAT YOU LIKE OR DISLIKE ABOUT THE PROGRAM OR COMPARE IT WITH SOME FAVOURITE PIECE OF SOFTWARE. LASTLY, "YOU CAN'T PLEASE ALL OF THE PEOPLE ALL OF THE TIME" MEANING, WHAT I THINK IS GOOD IN A PROGRAM, SOMEONE ELSE MAY NOT, AND VISA VERSA. GIVE TEN REVIEWERS THE SAME PROGRAM AND AT THE END OF THE DAY YOU WILL HAVE TEN VERY DIFFERENT REVIEWS.

WORD PROCESSORS ARE BECOMING AS COMMON PLACE AS THE UBIQUITOUS TYPEWRITER BUT WITH FEWER VICES AND FAR MORE TO OFFER. THEY ARE TO BE FOUND EVERYWHERE FROM A FLEET STREET NEWSPAPER OFFICE TO THE HOME COMPUTER IN THE CORNER OF THE LIVING ROOM. TWO PROGRAMS IN PARTICULAR ELECTRIC PENCIL AND SCRIPSIT HAVE BEEN AROUND FOR A LONG TIME AND ARE FAMILIAR TO A LOT OF TRS8Ø USERS. WHEN MOLIMERX FIRST ADVERTISED THEIR AJEDIT WORD PROCESSOR I WAS VERY INTERESTED TO SEE HOW IT WOULD COMPARE TO THE OTHER WORD PROCESSORS ON THE MARKET AND IN JANUARY I PURCHASED THE PROGRAM. NOW, AFTER A FEW MONTHS, I FEEL AS THOUGH I KNOW THE PROGRAM WELL ENOUGH TO BE ABLE TO WRITE A REVIEW ABOUT IT.

AJEDIT COMES, WITH MANUAL, IN A PLASTIC COATED RING BINDER AND THE PROGRAM ITSELF IS ON A 33 TRACK TRSDOS DISK. IF YOU HAVE NOT USED A WORD PROCESSOR BEFORE THEN TAKE SOME TIME TO READ THE INTRODUCTORY SECTION OF THE MANUAL. ALSO, IF IT SO HAPPENS THAT IT IS YOUR FIRST VENTURE INTO A DISK BASED OPERATING SYSTEM, THE MANUAL WILL LEAD YOU THROUGH THE INITIALIZING STEPS VERY CLEARLY AND SIMPLY. AS WITH ALL SOFTWARE THE FIRST JOB TO BE DONE IS THE MAKING OF A BACK-UP COPY AND THIS IS VERY EASY WITH AJEDIT. IT CAN ALSO BE VERY READILY TRANSFERRED TO YOUR PREFERRED DOS, ALTHOUGH MOLIMERX DO RECOMMEND LEAVING IT ON TRSDOS.

TYPING AJEDIT WILL TAKE YOU STRAIGHT INTO THE PROGRAM WITH A PROMPT INDICATING THE MACHINE TYPE / MODEL. IN MY PARTICULAR CASE IT SHOWS "GENIE" AND THE VERSION NUMBER 1.4 THERE ARE THREE MODES IN THE PROGRAM. THE "COMMAND", "EDIT" AND "TEXT ENTER" MODES EACH HAVING IT'S OWN DISTINCT



CURSOR DESIGNED TO PREVENT CONFUSION AS REGARDS WHICH MODE YOU ARE IN. ON ENTRY YOU ARE IN THE "COMMAND" MODE.

WHEN IN THE COMMAND MODE THE CURSOR IS AN ASTERISK FOLLOWED BY A GREATER THAN SIGN FOLLOWED BY A FLASHING GRAPHIC BLOCK. IT IS IN THIS MODE THAT THE DISK IN/OUT AND PRINT OPERATIONS ARE CONTROLLED. A SUMMARY OF THE COMMANDS FOR THIS MODE IS AS FOLLOWS ...

* ANY VALID DOS COMMAND .. DIR, KILL, FREE ETC ARE SUPPORTED.
* LOAD A FILE .. A STRAIGHT FORWARD COMMAND WHICH LOADS A FILE FROM DISK.
* RELOAD A FILE .. THIS WILL RELOAD A FILE FROM DISK, OVERWRITING WHAT IS IN THE BUFFER AT PRESENT BUT RETAINING THE CURRENT PARAMETERS. THIS COMMAND SHOULD BE USED WITH CAUTION, ENSURE THE FILE BEING WORKED ON HAS BEEN SAVED TO DISK BEFORE ISSUING THE RELOAD COMMAND.
* APPEND A FILE .. THIS COMMAND WILL APPEND A FILE FROM DISK TO THE FILE ALREADY IN MEMORY. VERY USEFUL FOR ADDING BLOCKS OF TEXT FROM A PREVIOUSLY WRITTEN LIBRARY.
* WRITE A FILE .. WRITES A FILE TO DISK USING THE SAME FILENAME AS WAS USED TO LOAD THE FILE. THIS COMMAND WILL NOT WORK IF AN APPEND HAS BEEN USED. THE MAIN USE FOR THE WRITE COMMAND IS TO UPDATE A FILE.
* COPY A FILE .. THIS COPIES THE CONTENTS OF THE TEXT BUFFER TO DISK AND IS HOW YOU WOULD CREATE A NEW FILE ON DISK.
* MEMORY .. SIMPLE, PRINTS THE AMOUNT OF FREE MEMORY.
* EXIT FROM PROGRAM .. WHEN THIS COMMAND IS USED IT AUTOMATICALLY DOES A WRITE BEFORE RETURNING TO DOS AND IS THE 'SAFEST' WAY TO LEAVE THE PROGRAM.
* QUIT .. A STRAIGHT FORWARD EXIT FROM THE PROGRAM BACK: TO DOS.
* NEW .. CLEARS THE TEXT BUFFER.
* PRINT .. SENDS THE CONTENTS OF THE TEXT BUFFER TO THE LINE PRINTER.
* TIDY .. AN ODD ONE THIS, IT TIDIES UP THE TEXT IN THE BUFFER TO GIVE (AS THE MANUAL STATES) A REASONABLE AMOUNT OF WHAT YOU SEE IS WHAT YOU GET.
* EXIT TO ANOTHER PROGRAM. THIS COMMAND ALLOWS THE TEXT ITSELF TO SPECIFY THE NEW PROGRAM TO BE RUN. THE PROGRAM FIRST DOES A WRITE THEN CHECKS THE TEXT FOR A DOS COMMAND.

COMMENTS ON THE COMMAND MODE

I HAVE GIVEN ONLY BRIEF DETAILS OF THE FUNCTIONS OF THE COMMANDS IN THIS MODE AS THEY REALLY ARE QUITE COMPLEX IN THEIR OPERATION. MY FAVOURITE COMMAND IS THE EXIT, WHERE AN AUTOMATIC WRITE IS DONE PRIOR TO RETURNING TO DOS. THIS CAN SAVE THE EMBARRASSMENT OF LEAVING THE PROGRAM WITH ALL THE TEXT IN THE BUFFET AND NOTHING ON THE DISK BECAUSE YOU FORGOT TO WRITE OUT THE FILE (OR AS HAPPENS IN PENCIL) THE CURSOR WAS IN THE WRONG PLACE. I HAVE NOT AS YET FOUND A USE FOR THE EXIT TO ANOTHER PROGRAM (SPECIFIED IN TEXT) BUT I SUPPOSE IT COULD BE USEFUL. THE NEED FOR THE TIDY COMMAND IS THE WAY THE PROGRAM OPERATES, MORE ON THIS LATER.

PRESSING <ENTER> ONCE TAKES YOU INTO TIRE EDIT MODE WHICH I WILL COVER LATER, PRESSING <ENTER> AGAIN TAKES YOU TO THE TEXT ENTER MODE. IN THIS MODE THE CURSOR RESEMBLES EITHER A GOLF CLUB OR A LARGE REVERSE LETTER L. DISTINCTIVE WOULD BE AN UNDERSTATEMENT IN TERMS OF A DESCRIPTION OF THIS CURSOR. THE FIRST THING THAT BECOMES APPARENT IS THE LACK OF REPEAT KEY FACILITIES WHICH I PERSONALLY HAVE COME TO EXPECT AS NORMAL IN A WORD PROCESSOR. THE SECOND MOST NOTICEABLE THING IS THE NEED TO LEAVE THE TEXT ENTER MODE FOR EDITING EXCEPT FOR BACKSPACING OVER PREVIOUSLY TYPED CHARACTERS ON THE CURRENT LINE. THE CURSOR CAN NOT BE MOVED TO A PROCEEDING LINE WITHOUT BEING IN THE EDIT MODE. PRESSING SHIFT - UP ARROW ACTIVATES THE LOWER CASE AND DOING THE SAME AGAIN RETURNS TO UPPER CASE.

SOME OF THE BEST FEATURES OF AJEDIT APPEAR IN THE ENTER MODE, WITH TEXT EMBEDDED COMMANDS WHICH I HAVE NOT SEEN IN ANY OTHER WORD PROCESSOR AND WHICH CAN GIVE EXCELLENT FORMATTED PRINT OUTS OF THE TEXT. A LIST OF THE TEXT EMBEDDED COMMANDS ARE AS FOLLOWS ....

* SET PAGE NUMBER.
* SET LINE NUMBER ON WHICH PAGE NUMBER WILL BE PRINTED.
* SET PAGE LENGTH.
* SET FORM LENGTH.
* SET LINE NUMBER FOR PRINTING OF HEADING.
* SET LINE NUMBER FOR PRINTING OF FOOTING
* SPECIAL PARAMETER SETTING ALLOWS FOR SETTING OF THE LINE NUMBERS FOR THE PAGE NUMBER, HEADING ETC, ALL IN ONE COMMAND.
* SET NUMBER OF CHARACTERS IN A LINE.
* SET INDENTATION.
* CARRY OUT A SET NUMBER OF LINE FEEDS.
* SET INSTRUCTION FOR A LINE FEED AFTER EVERY CARRIAGE RETURN.
* TURN ON / OFF LISTING TO SCREEN DURING PRINTING.
* TURN ON / OFF OUTPUT TO PRINTER.
* START NEW PAGE. T TURN ON / OFF PAGE NUMBERING.
* INITIALISE LINE COUNTER.
* SPECIFY TEXT TO BE PRINTED AT THE TOP AND BOTTOM OF EACH PAGE.
* SEND CONTROL CODES TO PRINTER, THIS WILL ALLOW USE OF VARIOUS PRINT SIZES ETC.
* TURN TEXT FORMATTER ON / OFF.
* TURN ON / OFF AUTOMATIC PARAGRAPH RECOGNITION.

* FORCE PRINTING TO TERMINATE.
* PAUSE PRINTING AT THE AND OF EACH PAGE TO ALLOW FOR SINGLE SHEET INSERTION.
* RESTART PRINTING OF CURRENT TEXT.
* EJECT PAPER TO BOTTOM OF PAGE.
* PAUSE PRINTING UNTIL ANY KEY IS PRESSED.
* LINK TEXT FILES.
* CENTRALISE TEXT.
* ENABLE MAIL / MERGES.
* SPECIFY FILE TO BE USED FOR MAIL MERGE.
* CLEAR STRING SPACE
* TERMINATE PRINTING.
* TABULATE TO SPECIFIED COLUMN.

THE LIST IS VERY IMPRESSIVE AND THE PERMUTATIONS OF THE VARIOUS COMMANDS SEEM

(CONTINUED ON PAGE 12)

INNOVATIVE

TRS-80 SOFTWARE

FROM THE PROFESSIONALS

EXTENDED



Extended Basic from Snapp Inc. is, in our opinion, the best add on utility to the Basic Interpreters for the Model III and Model II TRS-80 machines that is available on the market at the moment. It incorporates into the Interpreter, commands which hitherto have had to be loaded in separate utilities, indeed some of the commands, to the best of our knowledge, are not even available in other utilities. What must be emphasised is that these modifications to the Interpreter are entirely transparent to the user, they literally patch the Interpreter and become an integral part of it. The enhancements unfortunately are too lengthy to explain in detail in this advertisement, but briefly they add the following functions:

Single keystroke commands to:

List first line of program
List last line of program
List previous line of program
List next line of program
Edit current line
List current line

Ten single keystroke abbreviations are provided for the Basic commands:

AUTO	CLS	DELETE	EDIT	KILL
LIST	MERGE	NEW	LLIST	SAVE

Cross Reference

A complete cross reference utility with output either to the screen or to the printer listing all line numbers in which numeric or string variables are held. The reference may be to all variables or individual ones.

Dump Utility

A dump utility to direct to the display or printer the variables used in the program *together with their current values.*

Renumbering

A renumber utility. This is a sophisticated renumbering facility permitting not only renumbering, but also the relocation and duplication of blocks of code. It also has a higher speed execution than the present renumbering routine in the Interpreter.

String or Keyword Cross Reference

This utility will find and cross reference all keywords (commands) in the program together with, and probably most importantly, strings. That is to say, it is possible to find the whereabouts of specific words in the text of the program. Output is to the screen or to the printer.

Compression Utility

This utility takes a Basic program and compresses it in a number of different ways including merging multiple statements into single lines. Compressed programs typically will occupy 30-40% less memory space and run 7-10% faster.

In addition to the above, the Model III version adds single step entry into Basic (similar to LDOS or NEWDOS) together with the ability to recover accidentally "deleted" Basic programs.

Model III **£67.50**
Model II **£94.50**
Both exclusive of V.A.T. No P. & P.

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 **[refundable]** plus 50p postage.

MOLIMERX LTD.

A. J. HARDING (MOLIMERX)

1 BUCKHURST ROAD, TOWN HALL SQUARE,
BEXHILL-ON-SEA, EAST SUSSEX.

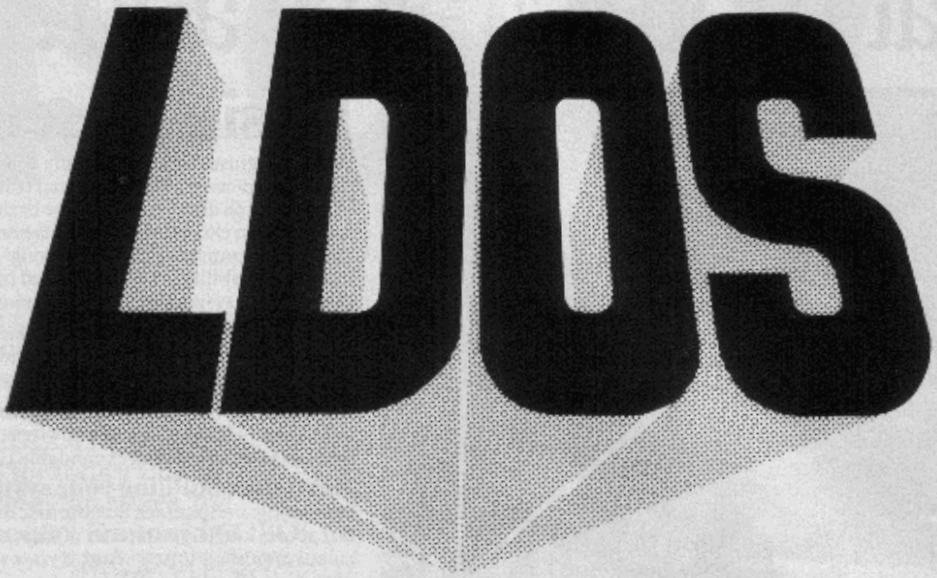
TEL: [0424] 220391/223636 **TELEX 86736 SOTEX G**

INNOVATIVE

TRS-80 SOFTWARE

FROM THE PROFESSIONALS

LDOS



First there were the TRSDOS's, 2.0, 2.1, 2.2 and 2.3. Then came Newdos+, essentially a patched version of the TRSDOS's but with a number of very useful commands and utilities added. Then VTOS 3.0 and VTOS 4.0. These constituted a departure from the earlier DOS's and featured Device Independence so that devices such as the keyboard, printer, VDU and disk drives could interact directly together. Then came Newdos80 which is a rewrite of Newdos+, adding new utilities and new Basic commands, its main features being the ability to mix different capacity drives on the same cable and the ability to use variable length records. Now from LOBO International comes LDOS, the fifth generation disk operating system for the TRS-80 microcomputer. It combines most of the advantages of the preceding disk operating systems and unlike some of them, is accompanied by a complete and readable set of documentation, which includes a Technical Section containing relevant addresses.

It is impossible to describe all of the features of LDOS in an advertisement. For instance it includes no less than 35 library commands as follows:—

APPEND	COPY	DEVICE	DIR	DO	FILTER	KILL
LIB	LINK	LIST	LOAD	MEMORY	RENAME	RESET
ROUTE	RUN	SET	SPOOL	ATRIB	AUTO	BOOT
BUILD	CLOCK	CREATE	DATE	DEBUG	DUMP	FREE
PROT	PURGE	SYSTEM	TIME	TRACE	VERIFY	XFER

All of the useful abbreviations in Newdos are included and the System Commands in Basic (CMD) now number eleven. A program called LBASIC/FIX is included, with which the normal TRSDOS Disk Basic may be patched to include a number of new commands and features. A Job Control Language is included and in fact is one of the most powerful features of LDOS. It allows the user to compile a sequence of commands or key strokes for later execution as a chain, with or without user intervention. There are too many new features to list them herein, but examples are: The ability to provide an audible signal, output through the cassette port. To flash or blink a one line message on the video display. A WAIT feature is included so that the machine can be put into a "sleep" state until such time as the system clock matches the time specified. And so on!

Hard disks in addition to single/double density, single/double sided, 8" and 5¼" floppies are supported although they may, of course, require hardware modifications. Utilities included in the package are:

BACKUP	COMMAND FILE	FORMAT	LCOMM
PATCH	RS232	KEY STROKE/MULTIPLIER	PRINTER FILTER

A Basic Renumber facility is included, as is a Basic Cross Reference function. Both are similar to the ones in Newdos+ and Newdos80. Most of the utilities are library commands which were existent in the previous DOS's, have been improved with the addition of new functions or facilities.

The prime development team of LDOS consisted of no less than 8 first rank programmers and they had the support and advice of six other well known programmers. They have done an excellent job to bring to the user what must be the best disk operating system so far produced for a microcomputer, which is destined to become the Standard DOS.

LDOS is totally upward compatible with TRSDOS, that is to say LDOS will be able to copy files and programs from TRSDOS disks onto LDOS formatted disks. As they are competitive disk operating systems, it is not suprising that the manual states that disks created under Newdos are not guaranteed to be compatible with LDOS, but we have not experienced any difficulty. We have done some work on investigating the compatibility of LDOS and the Video Genie and at the time of going to press we have found no incompatibilities. LDOS appears to run on the Video Genie without any problems at all. LDOS is compatible with either the Tandy or Electric Pencil lowercase modifications and Scripsit. LDOS is available for the Model I and Model III. A Model II version will be available shortly.

LDOS .....£85.00 **plus** VAT and £1.50 P&P.

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 [**refundable**] plus 50p postage.



MOLIMERX LTD.

A. J. HARDING (MOLIMERX)

1 BUCKHURST ROAD, TOWN HALL SQUARE,
BEXHILL-ON-SEA, EAST SUSSEX.

TEL: [0424] 220391/223636 **TELEX 86736 SOTEX G**



ENDLESS. IF YOU CAN'T FORMAT THE TEXT TO YOUR REQUIREMENTS WITH THESE COMMANDS THEN THROW AWAY THE PRINTER BECAUSE IT SURE AIN'T THE PROGRAM! LET US LOOK AT THREE OF THE COMMANDS TO GET SOME IDEA OF THE POWER OF THE FACILITIES IN THE PROGRAM.

THE CONTROL CODE COMMAND, WHICH IS SUPPLEMENTED BY A SPECIAL COMMAND ALLOWING YOU TO SPECIFY EITHER A CENTRONICS 737 OR AN EPSON PRINTER, WILL ALLOW YOU TO CHANGE ALL THE VARIOUS PRINT SIZES/TYPES THAT YOUR PRINTER WILL ALLOW, FOR EXAMPLE; CONDENSED, BOLD, ELONGATED AND ITALICS TO NAME BUT A FEW. THE COMMAND IS IN THE FOLLOWING FORMAT CC N1,N2 WHERE N1 AND N2 ARE THE REQUIRED CODES. CC 27,6□ FOR EXAMPLE WILL GIVE EMPHASISED CHARACTERS ON AN EPSON.

THE TABULATE COMMAND WAS ADDED IN THE LATER VERSIONS OF AJEDIT (FROM 1.4 ONWARDS) AND WITH ANOTHER CONTROL CHARACTER ADDED AT THE SAME TIME WILL ALLOW TABULATION FROM WITHIN A TEXT LINE. IMAGINE HAVING VARYING LENGTH DESCRIPTIONS OF ITEMS STARTING IN THE FIRST CHARACTER POSITION ON A LINE AND ENDING ANYWHERE FROM CHARACTER POSITION 30 TO 60 BUT YOU REQUIRE THE PRICES PRINTED OUT NEATLY IN COLUMN 70. WITH THE EMBEDDED TAB COMMAND IT IS VERY EASY TO OBTAIN THIS EFFECT. YOU CAN ALSO EMBED PRINTER CONTROL CODES IN TEXT IN THE SAME MANNER TO ALLOW FOR A CHANGE OF PRINT SIZE ON THE SAME LINE.

THE NEXT COMMAND COULD ALMOST BE CONSIDERED A SEPARATE PROGRAM NEVER MIND A SEPARATE COMMAND. MAIL MERGE IS A FACILITY GENERALLY FOUND AS AN ADD-ON TO A WORD PROCESSOR PROGRAM OR AS A SEPARATE PROGRAM IN IT'S OWN RIGHT. AJEDIT HAS MAIL MERGE AS A STANDARD PART OF THE PROGRAM AND IT WORKS IN THE FOLLOWING MANNER.
YOU FIRST CREATE A "MAILING LIST" WITH A SUITABLE FILE NAME. NEXT YOU WRITE YOUR "STANDARD" LETTER AND FOR THE ADDRESS AND NAME YOU SUBSTITUTE A SPECIAL COMMAND CHARACTER. YOU EMBED THE SPECIAL MAIL MERGE COMMAND IN THE TEXT AND THE FILE NAME OF THE MAILING LIST. WHEN YOU ISSUE THE PRINT COMMAND EACH TIME THE PROGRAM MEETS THE SPECIAL COMMAND CHARACTER IT READS THE NEXT PIECE OF INFORMATION FROM THE MAILING LIST FILE SPECIFIED.

COMMENTS ON THE TEXT ENTER MODE

THE ABSENCE OF REPEAT KEY AND "EDIT WHILE TYPING" IS A MAJOR DRAW BACK AS FAR AS I AM CONCERNED. MOST AUTHORS EDIT AND SHUFFLE TEXT ABOUT AS THEY GO AND IT IS AS MUCH A PART OF TYPING IN THE TEXT AS ENTERING THE WORDS THEMSELVES. THE EMBEDDED COMMANDS ARE EXCELLENT IN THEIR VARIETY AND OPERATION. AJEDIT HAS NO TYPE - AHEAD BUFFER AND IT IS ONLY TOO EASY TO DROP A CHARACTER AT THE END-OF-LINE WORD SHUFFLE. THE PROGRAM IS BETTER THAN PENCIL IN THIS RESPECT BUT IT IS STILL ANNOYING TO SEE THE ODD MISSING LETTER IN A WORD JUST TYPED.

THE EDIT MODE IS ENTERED

FROM THE TEXT ENTER MODE BY PRESSING DOWN ARROW. THERE ARE QUITE A NUMBER OF COMMANDS IN THIS MODE AND THEY ARE VERY MUCH AS THOSE USED TO EDIT IN BASIC. FOR EXAMPLE PRESSING H WILL HACK AND INSERT AND PRESSING K WILL KILL TEXT TO THE RIGHT OF THE CURSOR. THE EDIT MODE HAS A DIFFERENT CURSOR, A FLASHING GRAPHIC BLOCK, WHICH MAY BE MOVED ABOUT THE TEXT USING THE ARROW KEYS AND SPACE BAR. THE NEXT PAGE OF TEXT MAY BE LISTED (16 LINES) BY PRESSING P AND THE PREVIOUS PAGE BY PRESSING O. THERE ARE COMMANDS TO TAKE THE CURSOR TO THE START OR END OF TEXT. THE ROUTE TO THE COMMAND MODE IS BY PRESSING THE / KEY AND THE RETURN TO TEXT ENTER IS BY PRESSING ENTER OR WITH AN ACTUAL EDIT COMMAND. A LIST OF THE COMMANDS IS AS FOLLOWS ..

* MOVE CURSOR 1 CHARACTER TO RIGHT.
* MOVE CURSOR 1 WORD TO THE RIGHT.
* MOVE CURSOR ONE PLACE TO LEFT.
* MOVE CURSOR DOWN TO START OF NEXT LINE.
* MOVE CURSOR TO START OF PROCEEDING LINE.
* MOVE CURSOR TO STARTING LINE.
* MOVE CURSOR TO LAST LINE.
* DISPLAY FIRST 16 LINES.
* ADVANCE 16 LINES.
* DISPLAY PREVIOUS 16 LINES.
* CHANGE N (NUMBER) CHARACTERS.
* DELETE N CHARACTERS.
* HACK AND INSERT.
* INSERT.
* KILL TEXT TO RIGHT OF CURSOR.
* KILL N LINES OF TEXT.
* REPLACE N CHARACTERS.
* SEARCH LINE FOR CHARACTER.
* MOVE CURSOR TO END OF LINE AND ADD MORE TEXT.
* FIND A STRING IN TEXT.

COMMENTS ON THE EDIT MODE

THERE ARE CERTAINLY ADVANTAGES IN HAVING THE SAME EDIT COMMANDS AS APPEAR IN BASIC BUT, AS MOST OF US KNOW, THEY DO LACK THE EASE OF USE THAT FULL SCREEN EDITING (AS IN PENCIL FOR EXAMPLE) GIVES THE USER. THERE WAS A CHANGE IN RELEASE 1.4 TO ADD ANOTHER COMMAND KEY TO PERMIT INTRA-LINE TEXT ADDITION AS THE PREVIOUS VERSION DID NOT SUPPORT WRAP AROUND WHEN DOING TEXT INSERTION AND THIS COMPLICATES THE EDITING PROCEDURE. EVEN THOUGH THERE IS A DEGREE OF AWKWARDNESS IN THE EDIT MODE THE TEXT CAN BE MANIPULATED AS REQUIRED. ONE UNUSUAL FEATURE IS THE "TEXT-TIDY" COMMAND. THIS IS NECESSARY WHEN USING THE ADD TEXT WITH WRAP-AROUND AS THERE COULD BE A GAP AT THE END OF THE LINE AFTER THE LAST CHARACTER ENTERED. AFTER ISSUING THE TIDY COMMAND THE TEXT ON SCREEN IS NEATLY SORTED AND GIVES A BETTER IDEA OF WHAT THE FINAL OUTPUT WILL LOOK LIKE. AS IN THE TEXT ENTER MODE THE LACK OF THE REPEAT KEY FOR CURSOR MOVEMENT IS A HANDICAP.

CONCLUSIONS

AS I SAID WHEN I STARTED THE REVIEW, AJEDIT IS A MAJOR PIECE OF SOFTWARE AND IT IS VERY DIFFICULT TO ENSURE THAT YOU DO NOT MISS ANY IMPORTANT POINTS DURING THE REVIEW. I HOPE I HAVE COVERED THE VARIOUS COMMANDS IN NOT TOO BORING

A MANNER BUT IT IS ONLY BY LISTING THE VARIOUS COMMAND OPTIONS THAT I CAN GIVE SOME IDEA OF THE COMPLEXITY OF THIS PROGRAM. AJEDIT HAS JUST ABOUT EVERYTHING THAT IS REQUIRED IN A WORD PROCESSOR. IT HAS GOOD COMMUNICATION WITH THE DISK AND DOS. THE PRINTER CONTROL COMMANDS HELP THE USER TO GET THE BEST OUT OF HIS PRINTER. THE MAIL MERGE FACILITY IS EASY TO SET UP AND USE. AN EXCELLENT MANUAL GIVES HELP TO ALL, FROM THE NOVICE TO THE SEASONED WORD PROCESSOR USER. IF THE USER IS EXPERIENCED WITH THE EDIT FUNCTIONS OF BASIC THEN THERE IS NO NEED TO RE-LEARN A NEW SET OF EDIT COMMANDS. THE SAFE EXIT TO DOS COULD SAVE A LOT OF TEMPERS AND RE-TYPING.

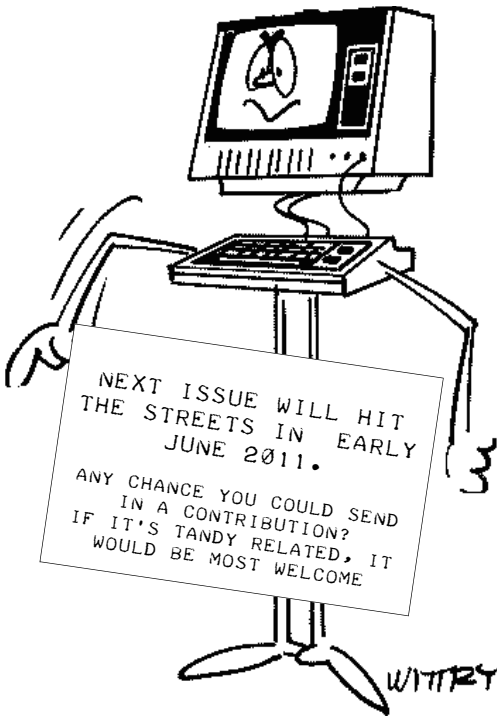
THERE ARE SOME MINUS POINTS AS I HAVE MENTIONED. THE LACK OF THE REPEAT KEY, TYPE-AHEAD AND EDIT WHILE TYPING MAKES LIFE THAT LITTLE MORE DIFFICULT IF YOU USE THE PROGRAM OFTEN.

TRAVELLING BETWEEN THE THREE MODES CAN BREAK THE "FLOW" OF USING THE PROGRAM. I AM NOT VERY ENAMOURED WITH THE "GOLF " CURSOR BEING MORE ACCUSTOMED TO THE USUAL GRAPHIC BLOCK. IF YOU WISH TO CHANGE THE CURSOR I HAVE PUT THE NECESSARY DETAILS AT THE END OF THE ARTICLE. I CONTACTED MOLIMERX MENTIONING THE LACK OF TYPE AHEAD AND SUGGESTED THAT AJEDIT COULD BE MADE MORE COMPATIBLE WITH LDOS AND COULD USE SOME OF THAT DOS'S FEATURES (AS DOES SCRIPSIT AND PENCIL) TO GIVE THE NECESSARY IMPROVEMENTS. I AM PLEASED TO REPORT THAT IN A RECENT LETTER, JOHN HARDING SAID THAT HE HAD CONTACTED THE PROGRAM'S AUTHOR TO LOOK INTO THE USE OF AJEDIT WITH LDOS. I'M LOOKING FORWARD TO SEEING THE OUTCOME!

MODIFYING AJEDIT'S "TEXT ENTER" CURSOR :-
USING PROZAP OR SUPERZAP, MODIFY RELATIVE FILE SECTOR 12H BYTE B8H (WHICH SHOULD BE BAH) INTO YOUR CHOICE OF CURSOR.

>> PLEASE REMEMBER TO WORK ON A BACKUP COPY THOUGH <<

--== o0o ==--



TIME (Part 1)

By Peter Stone

The inspiration for this article came about when I was trying to find some information I thought I'd bought with me when I'd moved, & then found out it must have been thrown away. I decided to write this article as an attempt to get my thoughts straight.

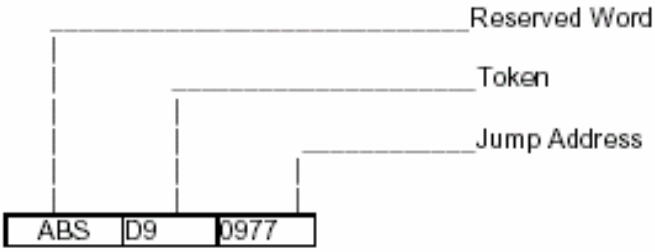
It all started, when, hunting around for a project for Trs8bit, I thought it would be interesting & a challenge to write some programs to display different types of clock, such as one displaying sidereal time alongside current time. The amount of calculation involved, showed that any attempt at a BASIC program would be (a) doomed to failure & (b) inaccurate. The main reason for this, is that the BASIC implementation in the TRS80, is a interpreter type rather than a compiler type. Basically the difference is that an interpreter translates each line of BASIC into machine code *every* time it is executed, whereas a compiler turns the entire BASIC program into machine code when it is first executed, & then runs this machine code. Alternatively a compiler program can be used to translate BASIC programs into machine code (/cmd) programs, which are then loaded & executed as normal.

At this point, it may be worthwhile to explain how a BASIC program is stored within a TRS80, so lets look at a simple BASIC program. The program below simply produces a one second delay.

```
10 FOR X = 1 TO 200
20 NEXT X
```

Within the TRS80 BASIC ROM, the various commands are treated as reserved words, & are assigned tokens in the range of 80H to F8H. When you type in a BASIC program, only this token is stored, not the complete words you type. Starting at location 1650H & extending to 1820H is a list of all the reserved words, in numerical order of the tokens. The first byte of each word is indicated by having bit 7 set, which is not used in ASCII code. there are two tables of jump addresses, located at 1608H to 164FH & 1822H to 1899H, plus a third area starting at 24B0H, these give the addresses where each command is executed. The table of level II reserved words is shown below, in alphabetical order.

The layout of the table is as follows;



Level II Reserved Words

ABS	D9	0977	GOSUB	91	1EB1	READ	88	21EF
AND	D2	25FD	GOTO	8D	1EC2	REM	8B	21EF
ASC	F6	2A0F	IF	8D	1EC2	RESET	82	0138
ATN	E4	15BD	INKEY$	C9	019D	RESTORE	90	1D91
AUTO	B7	2008	INP	DB	2AEF	RESUME	9F	1FAF
CDBL	F1	0ADB	INPUT	89	219A	RETURN	92	1EDE
CHR$	F7	2A1F	INSTR	C5	419D	RIGHT$	F9	291
CINT	EF	0A7F	INT	D8	0B37	RND	DE	14C9
CLEAR	B8	1E7A	KILL	AA	4191	RSET	AC	419A
CLOAD	B9	2C1F	LEFT$	F8	2A61	RUN	8E	1EA3
CLOSE	A6	4185	LEN	F3	2A03	SAVE	AD	41A0
CLS	84	01C9	LET	8C	1F21	SET	83	0135
CMD	85	4173	LINE	9C	41A3	SGN	D7	098A
CONT	B3	1DE4	LIST	B4	2B2E	SIN	E2	1547
COS	E1	1541	LLIST	B5	2B29	SQR	DD	13E7

CSAVE	BA	2BF5	LOAD	A7	4189	STEP	CC	2B01
CSNG	F0	0AB1	LOC	EA	4164	STOP	94	1DA9
CVD	E8	415E	LOF	EB	4167	STR$	F4	2836
CVI	E6	4152	LOG	DF	0809	STRING$	C4	2A2F
CVS	E7	4158	LPRINT	AF	2067	SYSTEM	AE	02B2
DATA	88	1F05	LSET	AB	4197	TAB(	BC	2137
DEF	BD	4158	MEM	C8	27C9	TAN	E3	15A8
DEFDBL	9B	1E09	MERGE	A8	418B	THEN	CA	---
DEFINT	99	1E09	MID$	FA	2A9A	TIME$	C7	4176
DEFSNG	9A	1E06	MKD$	EE	4170	TO	8D	---
DEFSTR	98	1E00	MKI$	EC	416A	TROFF	97	1DF8
DELETE	B6	2BC6	MKS$	ED	416D	TRON	96	1DF7
DIM	8A	2608	NAME	A9	418E	USING	8F	2CBD
EDIT	9D	2E60	NEW	BB	1B49	USR	C1	27FE
ELSE	95	1F07	NEXT	87	22B6	VAL	F5	2AC5
END	80	1DAE	NOT	CB	35C4	VARPTR	C0	24EB
EOF	E9	4161	ON	A1	1F6C	+	CD	249F
ERL	C2	24DD	OPEN	A2	4179	-	CE	2532
ERR	C3	24CF	OR	D3	25F7	*	CF	---
ERROR	9E	1FF4	OUT	A0	2AFB	/	D0	---
EXP	E0	1439	PEEK	E5	2CAA	↑(UPARROW)	D1	---
FIELD	A3	417C	POINT	C6	0132	>	D4	---
FIX	F2	0B26	POKE	B1	2CB1	=	D5	---
FN	BE	4155	POS	DC	27C5	<	D6	---
FOR	81	1CA1	PRINT	B2	206F	'	FB	---
FRE	DA	27D4	PUT	A5	4182	"	22	2866
GET	A4	417F	RANDOM	86	01D3	&	26	4194
						.	2E	0E67

If we save the example program listing above to disk, then examine the saved file using Superzap, we see the layout below.

```
DRV 00  FF56 6A0A 0081 2058 D531 20BD 2032 3030 .Vj....X.1...200
1   10  005E 6A14 0087 2058 0000 0004 0058 0000 .^j....X.....X..
1H  20  4988 048E 0400 FF00 FF00 FF00 FF00 FF00 I...............
    30  FF00 FF00 FF00 FF00 FF00 FF00 FF00 FF00 ................
```

Let's decode the file structure, & see how the tokens are used. Starting from the beginning of the file, the FFH indicates that this is a tokenised BASIC program, anything else will result in the 'DIRECT STATEMENT IN FILE' error message being displayed. The next two pairs of bytes, indicates the location of the first program line in RAM 6A56H, the next two pairs of bytes is the line number 000AH=10D. The next byte pair, 81H, is the token for the FOR BASIC statement, followed by 20H, which is the ASCII code for a space, followed by 58H, which is the ASCII for X. Next is D5H, the ASCII for =, followed by 31H, the ASCII for 1; 20H (space); BDH, the token TO; another space; 32H, the ASCII for 2; & 30H, twice, which is the ASCII for 0.

The next line starts with 00H, this is the End of Record marker. Basically each line of the BASIC program is considered a record. You should be able to decode the rest of the line, upto byte 18H, by using the table & explanation above. To assist you an ASCII equivalence label is included below.

Code			Code		
Dec	Hex	Char	Dec	Hex	Char
32	20	SPACE	64	40	@
33	21	!	65	41	A
34	22	"	66	42	B
35	23	#	67	43	C
36	24	$	68	44	D

37	25	%	69	45	E
38	26	&	70	46	F
39	27	'	71	47	G
40	28	(	72	48	H
41	29	)	73	49	I
42	2A	*	74	4A	J
43	2B	+	75	4B	K
44	2C	,	76	4C	L
45	2D	-	77	4D	M
46	2E	.	78	4E	N
47	2F	/	79	4F	O
48	30	0	80	50	P
49	31	1	81	51	Q
50	32	2	82	52	R
51	33	3	83	53	S
52	34	4	84	54	T
53	35	5	85	55	U
54	36	6	86	56	V
55	37	7	87	57	W
56	38	8	88	58	X
57	39	9	89	59	Y
58	3A	:	90	5A	Z
59	3B	:	91	5B	↑
60	3C	<	92	5C	↓
61	3D	=	93	5D	←
62	3E	>	94	5E	→
63	3F	?	95	5F	DEL

The main problem with an interpreter is, as I said above, it has to translate every command in the program as it is executed, which takes time, as each token has to be looked up & the appropriate sub-routine executed. This generally means that BASIC programs cannot be used for time critical tasks. A compiler however, would access a library of subroutines & replace each BASIC statement with the relevant code. While this would make for faster program execution, the problem is that a compiler makes no attempt at code optimisation. For example, if a program contained six FOR - NEXT loops, then there would be six machine code sub-routines in the program.

When it comes to machine code, timing is still something that has to be considered at times. Assembly language instructions include how many T states or clock cycles, each instruction takes. The Z80 processor in a TRS80, uses a clock circuit based on a 1.77MHz crystal, which gives a clock cycle of 0.565 µS (micro seconds). For an example, let's consider the LD instruction. A basic LD instruction such as LD B,D takes 4 T states, or 2.26 µS, whereas LD (HL),n takes 10 T states or 5.65 µS, & LD (IX+d),r takes 19 T states or 10.74 µS.

Going back to our BASIC example above, looking at *Microsoft BASIC decoded & other Mysteries*, the assembly language instructions that are executed by the FOR subroutine are shown below in an abbreviated form. To understand the sequence, start at the top left hand corner, read down, then move to the top of the next column. Also when you see two times for an instruction, separated by an /, the first is when the condition is met, & the second is when the condition isn't met.

M/C Instruction	T states (µS)	M/C Instruction	T states (µS)	M/C Instruction	T states (µS)
LD	7 (3.955)	LD	7 (3.955)	OR A	4 (2.26)
LD(nn)	13 (7.345)	LD	7 (3.955)	JR Z	12/7 (6.78/3.955)
CALL	17 (9.605)	LD A,(HL)	7 (3.955)	PUSH	11 (6.215)
EX (SP)	19 (10.735)	CP	4 (2.26)	LD	7 (3.955)
CALL	17 (9.605)	LD	7 (3.955)	CALL	17 (9.605)
POP	10 (5.65)	JR NZ	12/7 (6.78/3.955)	CALL	17 (9.605)
JR NZ	12/7 (6.78/3.955)	CALL	17 (9.605)	LD	7 (3.955)
ADD HL	15 (8.475)	PUSH	11 (6.215)	CALL	17 (9.605)
LD (SP)	6 (3.39)	CALL	17 (9.605)	POP	10 (5.65)
LD(nn)	16 (9.04)	CALL	17 (9.605)	EX DE	4 (2.26)
EX DE	4 (2.26)	CALL	17 (9.605)	RST	11 (6.215)

LD C	7 (3.955)	POP	10 (5.65)	LD DE	10 (5.65)
CALL	17 (9.605)	PUSH	11 (6.215)	PUSH	11 (6.215)
PUSH	11 (6.215)	PUSH	11 (6.215)	RET Z	11 (6.215)
CALL	17 (9.605)	LD	7 (3.955)	SUB	4 (2.26)
EX (SP)	19 (10.735)	RST	11 (6.215)	JP C	10 (5.65)
PUSH	11 (6.215)	LD	7 (3.955)	CP	4 (2.26)
LD HL	16 (9.04)	PUSH	11 (6.215)	JP NC	10 (5.65)
EX (SP)	19 (10.735)	PUSH	11 (6.215)	RLCA	4 (2.26)
RST	11 (6.215)	LD HL	16 (9.04)	LD	7 (3.955)
CP L	4 (2.26)	EX (SP)	19 (10.735)	LD	7 (3.955)
RST	11 (6.215)	LD	7 (3.955)	EX DE	4 (2.26)
JP Z	10 (5.65)	PUSH	11 (6.215)	LD HL	16 (9.04)
JP NC	10 (5.65)	INC	6 (3.39)	ADD HL	15 (8.475)
PUSH	11 (6.215)	CALL	17 (9.605)	LD C,(HL)	7 (3.955)
CALL	17 (9.605)	OR	4 (2.26)	INC	6 (3.39)
POP	10 (5.65)	CALL	17 (9.605)	LD	7 (3.955)
PUSH	11 (6.215)	LD (nn)	13 (7.345)	PUSH	11 (6.215)
JP P	10 (5.65)	LD (nn)	13 (7.345)	EX DE	4 (2.26)
CALL	17 (9.605)	LD	7 (3.955)	INC	6 (3.39)
EX (SP)	19 (10.735)	CP	4 (2.26)	LD	7 (3.955)
LD DE	10 (5.65)	JR Z	12/7 (6.78/3.955)	CP	4 (2.26)
LD A,(HL)	7 (3.955)	OR	4 (2.26)	RET NC	11 (6.215)
CP	4 (2.26)	JP NZ	10 (5.65)	CP	4 (2.26)
CALL Z	17/10(9.6/5.65)	INC	6 (3.39)	JP Z	10 (5.65)
PUSH	11 (6.215)	LD	7 (3.955)	CP	4 (2.26)
PUSH	11 (6.215)	INC	6 (3.39)	JR NC	12/7 (6.78/3.955)
EX DE	4 (2.26)	OR	4 (2.26)	CP	4 (2.26)
CALL	17 (9.605)	JP Z	10 (5.65)	JP NC	10 (5.65)
JR	12 (6.78)	INC	6 (3.39)	CP	4 (2.26)
CALL	17 (9.605)	LD	7 (3.955)	CCF	4 (2.26)
CALL	17 (9.605)	INC	6 (3.39)	INC	6 (3.39)
POP	10 (5.65)	LD	7 (3.955)	DEC A	4 (2.26)
PUSH	11 (6.215)	EX DE	4 (2.26)	RET	10 (5.65)
PUSH	11 (6.215)	LD (nn)	13 (7.345)		
LD BC	10 (5.65)	LD A, (nn)	13 (7.345)		

This takes between 803.99 µS, when conditions are met, & 768.915 µS when conditions aren't met.

The NEXT subroutine is shown below, & is in the same format as the one above.

M/C Instruction	T states (µS)	M/C Instruction	T states (µS)	M/C Instruction	T states (µS)
LD DE	7 (3.955)	JR	12 (6.78)	PUSH	11 (6.215)
CALL NZ	10 (5.65)	INC	6 (3.39)	LD E,	7 (3.955)
LD	7 (3.955)	INC	6 (3.39)	INC	6 (3.39)
CALL	17 (9.605)	INC	6 (3.39)	LD D,	7 (3.955)
JP NZ	10 (5.65)	INC	6 (3.39)	INC	6 (3.39)
LD SP	6 (3.39)	LD C,	7 (3.955)	EX (SP),	18 (10.17)
LD (nn)	13 (7.345)	INC	6 (3.39)	CALL	17 (9.605)
PUSH	11 (6.215)	LD B,	7 (3.955)	POP	10 (5.65)
LD A,(nn)	7 (3.955)	INC	6 (3.39)	POP	10 (5.65)
INC	6 (3.39)	EX (SP)	18 (10.17)	SUB	4 (2.26)
PUSH	11 (6.215)	LD E,	7 (3.955)	CALL	17 (9.605)
PUSH	11 (6.215)	INC	6 (3.39)	JR Z,	12/7 (6.78/3.955)
LD A,(	7 (3.955)	LD D,	7 (3.955)	EX DE,	4 (2.26)
INC	6 (3.39)	PUSH	11 (6.215)	LD (nn),	13 (7.345)
OR	4 (2.26)	LD	7 (3.955)	LD	7 (3.955)
JP M	10 (5.65)	LD	7 (3.955)	LD	7 (3.955)
CALL	11 (6.215)	CALL	17 (9.605)	JP	10 (5.65)

EX (SP),	19 (10.735)	LD A,	7 (3.955)	LD SP,	6 (3.39)
PUSH	11 (6.215)	CP	4 (2.26)	LD (nn)	13 (7.345)
CALL	17 (9.605)	JP Z,	10 (5.65)	LD (HL),	16 (9.04)
POP	10 (5.65)	EX DE,	4 (2.26)	LD A,	7 (3.955)
CALL	17 (9.605)	POP	10 (5.65)	CP	4 (2.26)
POP	10 (5.65)	LD (HL),	16 (9.04)	JP NZ,	10 (5.65)
CALL	17 (9.605)	DEC HL,	6 (3.39)	RST	11 (6.215)
PUSH	11 (6.215)	LD (HL),	16 (9.04)	CALL	17 (9.605)
CALL	17 (9.605)	POP	10 (5.65)	RST	11 (6.215)
				JR Z,	12/7 (6.78/3.955)

This takes between 441.265 µS, when conditions are met, & 435.615 µS when conditions aren't met.As you can see, this is a significant portion of time.

In machine code, things are different. A delay sub-routine looks like this.

```
        LD      B,5             ;NUMBER OF COUNTS TO LOOP
LP1     LD      HL,0FFFFH       ;VALUE TO DECREMENT
LP2     DEC     HL
        LD      A,H
        OR      L               ;HL-0?
        JP      NZ,LP2          ;IF NO, DEC AGAIN
        DJNZ    LP1             ;DEC, B--B=0?
        RET
```

As can be seen, a delay subroutine is simply achieved by loading a register or register pair with a value, & decrementing it until it reaches zero. If you figure out how long each instruction in the loop takes to execute, (as we did above for the BASIC tokens), we can calculate the delay time. In the program example above, there are two loops, one inside the other. One of the loops uses the HL register pair, & the other the single register B.

The inner loop, (lines 2 - 6 above), tests for zero by loading register A from H, ORing it with L & decrementing the count if the result of the OR if it is not zero. HL is loaded with FFFFH (65535), this provides the maximum delay. Computing the length of time the loop takes to execute is as follows

Instruction	T States
DEC HL	6
LD A,H	4
OR L	4
JP NZ,LP2	10
Total	24

Remembering that a T state is 0.565µS, then 24 T states takes 13.512µS to execute. this, multiplied by 65535, gives a time delay of 0.88556 seconds. The outer loop uses the B register, & the decrementing is done by the DJNZ instruction, which both decrements B & jumps to LP1 if it is non-zero. The JP statement at the end of the LP1 loop, could be replaced by a JR instruction, which takes 12 T states, instead of 10, which would give a time period of 0.96 seconds. It can be seen that by adjusting the values loaded into the B & HL registers, various time delays can be programmed quite simply.

I'll expand on this in the next article. In closing, if anyone has any topics they want me to cover, then please let me know by emailing me at peter_stone45@hotmail.com.

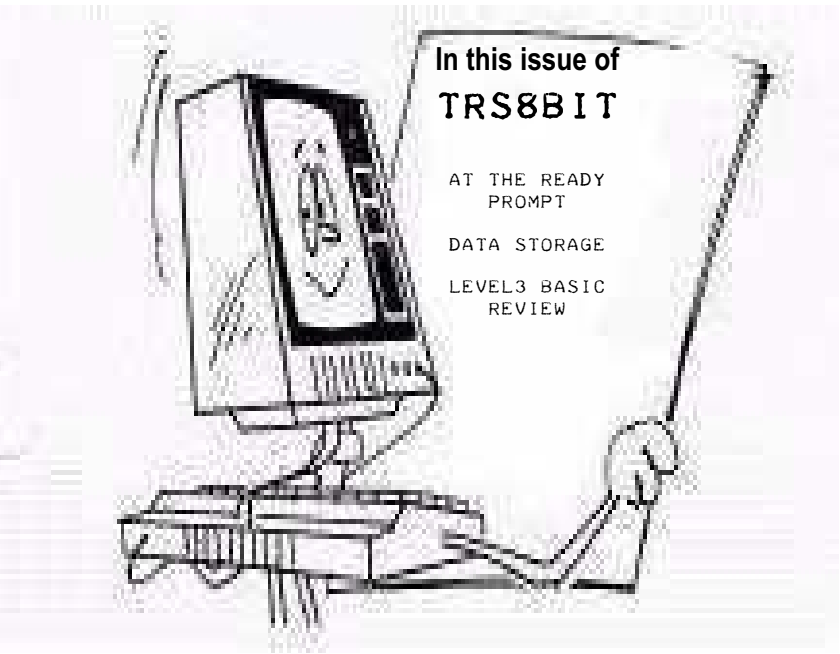
TRS8BIT



HI EVERYONE, AND WELCOME TO THE JUNE 2011 EDITION OF TRS8BIT.

I'LL START OFF WITH A BIT OF VERY SAD NEWS; THE RECENT DEATH OF RICK HANSON, FROM CLUB100.ORG. OUR CONDOLENCES GO TO HIS FAMILY AND MANY FRIENDS. HE WAS A REAL GENTLEMAN AND HAD WORKED, SUPPORTING THE MODEL 100'S SINCE ITS INCEPTION. HE'LL BE SADLY MISSED. I HOPE THE WEBSITE MANAGES TO CONTINUE, IT WOULD BE A MOST FITTING TRIBUTE TO HIM.

THE WEBSITE IS STILL ATTRACTING INTEREST, WHICH IS MOST PLEASING. THE 8K VISIT BARRIER HAVING JUST BEEN BROKEN WAS YET ANOTHER MILE-STONE. 10K IS THE NEXT BIG ONE! THANK YOU, EVERYONE, FOR YOUR INTEREST, SUPPORT AND INFECTIOUS ENTHUSIASM. I'M STILL ENJOYING EVERY MINUTE OF IT, EVEN AFTER 30 YEARS!



I SHALL PLACE A COPY OF THIS EDITION OF TRS8BIT 'FOR SALE' ON EBAY. ANYTHING FOR A BIT OF FREE PUBLICITY!

JOHN BENSON IN AUSTRALIA, HAS AT LAST FOUND THE INSTRUCTIONS FOR IS TC-8 CASSETTE INTERFACE. BY THE WAY, JOHN IS NOW LOOKING FOR TECHNICAL DETAILS OF THE TANDY HI-RES GRAPHICS BOARD, TANDY CATALOGUE NO. 26Ø-98ØØ. THIS, IT SEEMS, WAS ONLY AVAILABLE IN THE UK, SO IF ANYONE OUT THERE CAN ASSIST, PLEASE LET ME KNOW. I'VE ALSO ADDED A NUMBER OF NEW, TANDY RELATED, WEBSITES FOR YOU TO VISIT WHEN YOU GET A COUPLE OF SPARE MOMENTS. THERE'S A COUPLE IN AUSTRALIA AND ANOTHER, I'M NOT SURE WHERE, WHICH HAS A MASSIVE COLLECTION OF 'TANDY' RELATED CATALOGUES

MATTHEW REED HAS DONE A FANTASTIC JOB ON AMENDING LDOS SO THAT THE SYSTEM DATE NOW RUNS UNTIL 2Ø79. THE MODEL 1 SHOULD NOW MANAGE A CENTURY OF COMPUTING QUITE HAPPILY!! FREE UPGRADES ARE AVAILABLE FROM HIS WEBSITE.
I'VE ALSO HEARD FROM PETER STONE THAT MATTHEW IS AWARE OF THE NECESSARY CHANGES TO HIS EMULATOR TO ALLOW THE ACULAB FT TO RUN ON IT. I UNDERSTAND THAT IT'S A MATTER OF THE ROM SIZE BEING DULY CHANGED FROM WITHIN TRS-TOOLS WHICH WILL CREATE AN ADDITIONAL WORKLOAD FOR HIM. I'M REALLY LOOKING FORWARD TO SEEING THE FT RUNNING ON A M1 AGAIN. I'VE STILL HAD NO LUCK IN GETTING A FT TAPE

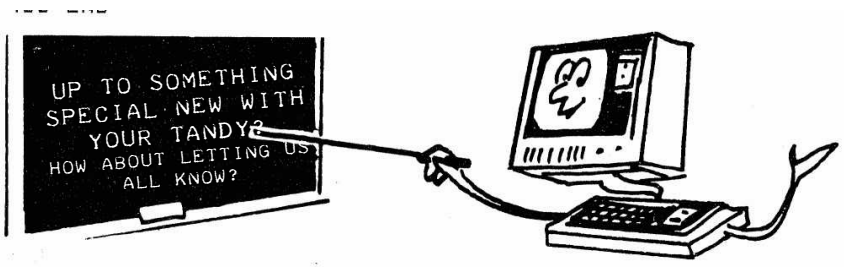
(CONTINUED ON PAGE 15)

AT THE
READY>
PROMPT



I NOTICED THAT IRA, WHILST SORTING THOUGH A RECENT SOFTWARE DONATION, CAME ACROSS A LITTLE PROGRAM WHICH CLAIMS TO IDENTIFY WHETHER YOUR MODEL I ROM WAS V1.Ø, 1.1, 1.2, 1.3, OR OTHER. THE ROMS OUT IN THE WILD ARE EITHER V1.2 OR V1.3. DOES ANYONE HAVE A MODEL I SYSTEM WITH V1.Ø OR V1.1?? THE PROGRAM IS AS FOLLOWS:

```
1Ø REM     PROGRAM TO FIND
VERSION OF MODEL I ROM
2Ø CLS
3Ø FOR I=11264 TO 12287
   :V=PEEK(I):S=S+V:NEXT I
   :X=S/16
4Ø A=(X-FIX(X))*16
   :Y=FIX(X)/16
   :B=(Y-FIX(Y))*256
5Ø V=(A+B)
6Ø IF V=176 PRINT"ROM 1.Ø"
   : GOTO 1ØØ
7Ø IF V=142 PRINT"ROM 1.1"
   : GOTO 1ØØ
8Ø IF V=1Ø  PRINT"ROM 1.2"
   : GOTO 1ØØ
9Ø IF V=162 PRINT"ROM 1.3"
   : GOTO 1ØØ
92 PRINT"VERSION OF ROM
UNKNOWN"
1ØØ END
```



DATA STORAGE

FROM AN ORIGINAL ARTICLE BY
DS PECKETT

IN THIS ARTICLE, I WILL TAKE A CLOSE LOOK AT WAYS OF USING CASSETTE TAPE MORE EFFECTIVELY AS A MICROCOMPUTER DATA STORAGE MEDIUM. ALTHOUGH CASSETTES ARE THE STANDARD WAY OF STORING DATA FOR SMALL MICROS, THEIR EFFECTIVE USE IS AN AREA WHICH DOES NOT USUALLY RECEIVED A LOT OF ATTENTION. MANY MICRO HANDBOOKS ARE UNJUSTIFIABLY WEAK ON THE TOPIC BUT, I SHALL PROVIDED YOU WITH A NUMBER OF TOOLS WHICH, WHEN COMBINED, WILL HELP YOU TO SOLVE PARTICULAR PROBLEMS. SINCE I USE A VIDEO GENIE, SOME OF THE INFORMATION WILL INEVITABLY BE BIASED TO THE LEVEL II BASIC OF THE GENIE AND TRS-8Ø. IN SOME WAYS, THIS MAY NOT BE SUCH A BAD THING SINCE THESE COMPUTERS ARE AMONG THE WORST AROUND WHEN IT COMES TO EFFECTIVE DATA STORAGE; IF YOU KNOW HOW TO SOLVE THEIR PROBLEMS, MOST OTHER COMPUTERS SHOULD BE PIECES OF SILICON CAKE!

A GOOD PLACE TO START IS THE BEGINNING, AND IN THIS CASE THAT MEANS LOOKING AT JUST WHY CASSETTES ARE SO WIDELY USED, AND AT SOME OF THEIR CHARACTERISTICS.

CASSETTES ARE USED BECAUSE MOST SMALL PERSONAL COMPUTERS - THE SORT YOU MIGHT HAVE IN YOUR HOME RATHER THAN YOUR OFFICE - GENERALLY COME WITH CASSETTE FACILITIES AS STANDARD. FAILING THAT, THE INTERFACE

IS THERE AND ALL YOU HAVE TO DO IS PROVIDE A CASSETTE MACHINE AS A NOT-SO-OPTIONAL EXTRA IF YOU WANT TO SAVE PROGRAMS AND/OR DATA. WHY ARE THEY PROVIDED ON SMALL MICROS? THAT'S EASY - BECAUSE THEY ARE CHEAPER THAN ANY OTHER LONG-TERM PROGRAM AND DATA STORAGE MEDIUM. COMPARE THE COST OF A REASONABLE CASSETTE DRIVE (£25?) WITH THAT OF FLOPPY DISC MECHANISMS, WHICH COST FROM £250. DON'T FORGET THAT TO USE THE DISC YOU WILL



PROBABLY ALSO REQUIRE EXTRA MEMORY, A DISC INTERFACE AND A DISC OPERATING SYSTEM (DOS) - TOTAL COST, SAY ANOTHER £300.

CASSETTES THEMSELVES ARE ALSO VERY CHEAP IN TERMS OF STORAGE CAPACITY WHEN COMPARED WITH OTHER POSSIBILITIES. A STANDARD C60 CASSETTE, USED WITH A COMMON MICRO, CAN STORE AROUND 225K; THAT'S AROUND THE SAME VOLUME AS A TYPICAL MINI-FLOPPY DISC FOR MAYBE A QUARTER OF THE PRICE.

OF COURSE, WE ALL KNOW OF, AND ARE OFTEN FRUSTRATED BY, THE FACT THAT TAPE HAS DISADVANTAGES. THE MAIN ONES ARE THAT TAPE IS SLOW AND THAT IT ONLY PROVIDES SEQUENTIAL STORAGE. IN TERMS OF SPEED, NOT MANY TAPE SYSTEMS CAN TRANSFER DATA AT MUCH MORE THAN 200 CHARACTERS PER SECOND WHEREAS A DECENT DISC SYSTEM CAN READ AND WRITE AT AROUND 20,000 CHARACTERS PER SECOND - A 100:1 SPEED ADVANTAGE. TO PUT THAT IN PERSPECTIVE, AND ALLOWING FOR DOS OVERHEADS, A GOOD TAPE SYSTEM CAN LOAD A 16K PROGRAM IN 80 SECONDS, WHILE AN AVERAGE DISC SYSTEM SHOULD LOAD THE SAME PROGRAM IN AROUND 3-5 SECONDS.

THE FACT THAT TAPE IS A SEQUENTIAL STORAGE MEDIUM

ONLY BECOMES APPARENT WHEN WE START TO STORE MORE THAN ONE PROGRAM OR SET OF DATA ON A SINGLE CASSETTE. UNLESS WE KNOW ROUGHLY WHERE ON THE TAPE A PARTICULAR ITEM OF DATA IS, AND CAN FIND IT VIA A TAPE COUNTER, THE COMPUTER MUST START AT THE BEGINNING AND SEARCH ALL THE WAY ALONG THE TAPE UNTIL IT FINDS WHAT IT IS LOOKING FOR - THIS COULD TAKE UP TO 45 MINUTES! A DISC SYSTEM, ON THE OTHER HAND, IS LIKE AN ORDINARY LP RECORD. JUST AS YOU CAN SET THE PICKUP ARM DIRECTLY TO TRACK 5, THE COMPUTER CAN SET THE FLOPPY DISC'S READ HEAD DIRECTLY TO THE POSITION OF THE DATA OR PROGRAM IT NEEDS. DISC READ TIME IS THEREFORE ALMOST INDEPENDENT OF WHERE ON THE DISC THE INFORMATION IS STORED; FOR THIS REASON, WE CALL DISCS 'RANDOM ACCESS' DEVICES.

AT HOME, HOWEVER, THERE IS NOT USUALLY THE SORT OF PRESSURE WHICH DEMANDS VIRTUALLY INSTANTANEOUS ACCESS; AFTER ALL, COMPUTING IS MEANT TO BE A HOBBY! WE CAN OFTEN POSITION THE TAPE BY HAND IN ORDER TO READ IN A PARTICULAR PROGRAM, OR ORGANISE A SET OF DATA SO THAT THE MICRO CAN READ IT IN THE ORDER IN WHICH IT IS NEEDED. PROBLEMS ARISE, HOWEVER, WHEN WE CANNOT PREDICT WHAT DATA THE PROGRAM WILL NEED AND WHEN IT WILL NEED IT.

WITH A RANDOM-ACCESS SYSTEM, THERE IS LITTLE DIFFICULTY; THE SYSTEM GOES STRAIGHT TO THE DATA. WITH TAPE, HOWEVER, UNLESS THE NEXT ITEM(S) OF DATA ARE PHYSICALLY AFTER THE LAST ONE(S), WE CANNOT LEAVE THE COMPUTER TO FIND INFORMATION AS IT NEEDS IT. WE HAVE TO MANUALLY REWIND THE TAPE TO THE CORRECT POINT, AS INSTRUCTED BY THE PROGRAM.

THIS IS, OF COURSE, A FUNDAMENTAL LIMITATION OF ANY SEQUENTIAL STORAGE MEDIUM. ALTHOUGH THERE ARE RELATIVELY LOW COST ALTERNATIVES, SUCH AS THE EXATRON 'STINGY FLOPPY', THERE IS, IN REALITY, LITTLE TO BEAT THE DISC, IN ITS VARIOUS GUISES, FOR RANDOM-ACCESS. THE ACULAB 'FLOPPY TAPE' COMES A VERY CLOSE SECOND WHEN USED IN CONJUNCTION WITH ACULAB'S ADVANCED BASIC 'XBAS' SOFTWARE.

DATA STORAGE ON CASSETTE

FIRST, THOUGH, LET'S TAKE A LOOK AT JUST HOW THE DATA IS PHYSICALLY RECORDED ON TO THE TAPE. INFORMATION IS STORED IN A TOTALLY SEQUENTIAL MANNER; FIG. 1 IS AN IMPRESSION OF A SINGLE BYTE WITH VALUE 8F HEX (143 DECIMAL OR 1000 1111 BINARY) STORED ON TAPE. WHAT CAN WE SEE? FIRST OF ALL, THE DATA ONLY OCCUPIES HALF OF THE WIDTH OF THE TAPE - CASSETTE DRIVES FOR CHEAP COMPUTERS ARE STANDARD DOMESTIC MONO MACHINES AND SO USE THE MONO RECORDING FORMAT. NEXT, AND MORE IMPORTANTLY, THE DATA IS STORED BIT-BY-BIT, WITH THE MOST SIGNIFICANT BIT (MSB) FIRST. (THE WAY IN WHICH THE 1'S AND 0'S ARE ACTUALLY CODED ON THE TAPE IS DIFFERENT FOR VIRTUALLY VERY COMPUTER).

THAT THEN, IS THE STORAGE OF A SINGLE BYTE. FIGURE 2 SHOWS HOW A SINGLE BLOCK OF DATA MAY BE STORED ON TAPE. THERE ARE FOUR DISTINCT PARTS: THE FIRST IS AN IDENTIFYING SEQUENCE, CONTAINING NO DATA, BUT IS NEEDED TO GIVE THE COMPUTER SOMETHING TO LOCK ONTO AND TO STABILIZE ITS CIRCUITS BEFORE IT STARTS READING THE DATA. THE SECOND ITEM IS A SINGLE BYTE, OR MORE LIKELY A GROUP OF A FEW BYTES,

MAIN GENIE DEALER

Premier Publications now offer the FULL range of Genie equipment, including expander boxes, disk drives and video monitors for the Genie and TRS80 I/III range. Samples prices . . .

GENIE I with cassette deck	£319.95 inc
GENIE II with numeric pad	£334.95 inc
40 track disk drive (EG400)	£219.95 inc
32K Expander Box (EG3014)	£214.99 inc
12inch black & white monitor	£69.95 inc
9inch high quality monitor	£92.00 inc
Parallel Printer Interface	£37.95 inc

See us at Stand 536 Computer Fair

PREMIER WORD PROCESSOR PACKAGES

Premier offer a wide range of WP packages for the hobbyist and small businessman, starting from as little as £370. All use our phenomenally successful WORD4WORD word processor which was our top-selling business program in 1981. Most of the packages use the CENTRONICS 737, a superb printer which is capable of correspondence quality output with three type faces, true descenders, sub/superscripts, underlining AND proportional spacing. The WORD4WORD supplied makes full use of the above features, and gives true right justification in proportional mode, whether the text is a letter or a series of columns. All disk-based packages are supplied with DOSPLUS.

WP1 W4W + GENIE II	£369.95
WP2 W4W + GENIE II + CENTRONICS 737 PRINTER + all cables	£759.95
WP3 W4W + 48K GENIE + 40 track disk drive + DOSPLUS + 737 Printer	£1245.00
WP4 as WP3, but twin drives	£1449
WP5 as WP4, but with twin double density disk system	£POA
WP6 W4W, GENIE II, disks and a daisywheel printer. Available MAY	£TBA

WORD4WORD

STUNNING NEW TRS80/VG WORD PROCESSOR

Why pay £75 to £150+ for a W.P? W4W gives you all the facilities you could wish for in WP for a senislbe price. Amongst the features included are:

FULL SCREEN EDITING & TWO WAY SCROLLING
FULL TEXT INSERT/DELETE ON SCREEN
GLOBAL SEARCH & AMEND
TEXT SEARCH — TOTAL TEXT MOBILITY
TEXT FORMATTING TO SCREEN OR PRINTER
CASSETTE/DISK STORAGE OF TEXT
VISION LOAD OF STORED TEXT
STANDARD LETTER ROUTINES
OVERTYPE CORRECTION
SINGLE KEY WORD DELETION
PAGE LAYOUT & NUMBERING
TOTAL PRINTER CONTROL
TEXT HIGHLIGHTING FOR U/L RECOGNITION
TEXT BLOCK MANIPULATION
COMPREHENSIVE TAB & TABLE GENERATION

Many other features are included in W4W. We would need several pages to do justice to this superb product. W4W is comparable to most purpose-built systems. COMMISSIONED AND DEVELOPED BY PREMIER WITH THE HOBBYIST AND SMALL BUSINESSMAN IN MIND.
Please state machine type and printer when ordering. W4W can cope with the Centronics 737/9 and Tandy proportional spacing modes. Price CASSETTE £33.95, DISK/FLOPPY TAPE £37.95.

SCREDT

This fabulous screen editor has the following features:

- twin cursors for easy editing
- two speed copying from second to main cursor
- complete mobility of new cursor
- user-definable cursor locations
- auto-repeat cursor movement keys
- 27 new functions
- useable with most assemblers

SCREDT also allows the user to save a segment of screen information to memory for later retrieval/editing. SCREDT is a machine code program needing less than 3K of user RAM. It is compatible with Premier's TOOLKiT II, and locates at the top of memory.

Specify Model I, II GENIE I, II and memory size when ordering. Cassette £14.95. DISK (specify DOS) £16.95. Floppy Tape £17.95.

MICROTRAIN FOR TRS80/VG

At last you can combine two of the most popular hobbies in the country — railways and computing. MICROTRAIN, a brand-new quality simulation from Premier, allows you to set up a network of tracks, points, stations, tunnels, bridges etc and then run trains to your own timetable! Signalling is provided either automatically or to your design.

MICROTAIN is a screen-based simulation. With one keystroke you can draw and signal lines many scale-miles long. Trains can be run simultaneously and the speed and length of each train is user-selectable. A cursor is used to move the whole screen display left or right for visual inspection of any part of the network. You can use any of the machine's graphics on-screen to build up scenery. Designs can be stored onto tape/disk for future retrieval.

MICROTRAIN is a machine code program and will run on any Genie or TRS80. State machine when ordering.
CASSETTE £14.95 inc. DISK (specify DOS) £16.95 inc.

TOOLKIT 2 FOR TRS80 I & III/VG

TOOLKIT 2 features 17 new easy-to-remember command words and a machine code monitor which greatly enhance an already powerful BASIC

REPLACE	replace any string, word or variable
VARS	gives a list of variables on screen
TRACE	see line contents as executed
ABBREV	26 BASIC words become single-key entries
RENUMBER	operates from any start in any increment
BLANK	removes unwanted spaces and LET statements
VTAPE	true Vision load plus APPENDing
MC	a full machine code monitor
VARTRACE	lists lines as executed plus variables
PAGE	controlled list scrolling
OLD	retrieves lost programs!
FIND	anyting in a BASIC listing
REMKIL	kills REMS!
DUPL	copy existing line to new line
LFIND, LVARS, LREPLACE	all work to printer
PRICES	Cassette £29.95 Disk (Specify DOS) £32.95

POSTAGE & PACKING: Software 75p per order. Eproms/Discs 95p per order
GENIE HARDWARE POA
ALL PRICES INCLUDE VAT

BARCLAYCARD VISA

Premier Publications

We welcome Access

208 Croydon Road, Anerley, London SE20 7YX. Tel: 01-659 7131

IDENTIFYING THE START OF THE DATA AREA. THEN COMES THE DATA ITSELF, FOLLOWED BY ANOTHER DISTINCTIVE BYTE OR GROUP OF BYTES TO INDICATE THAT ALL THE DATA HAS BEEN TRANSFERRED AND THAT THE COMPUTER SHOULD STOP READING AND SWITCH THE DRIVE OFF. VIRTUALLY ALL COMPUTER SYSTEMS USE THIS SORT OF SCHEME, ALTHOUGH THE DETAILS OBVIOUSLY DIFFER FROM MACHINE TO MACHINE. TRS-80 AND VIDEO GENIES' USE THE FOLLOWING FORMAT FOR DATA TAPES (OTHER SORTS OF TAPES HAVE OTHER FORMATS):
A. 255 BYTES, EACH OF VALUE ZERO, AS A SYNCHRONIZING LEADER.
B. A SINGLE BYTE OF VALUE A5 HEX TO MARK THE START OF THE DATA.
C. THE DATA ITSELF, WITH INDIVIDUAL ITEMS SEPARATED BY ASCII COMMAS (2C HEX).
D. A SINGLE CARRIAGE RETURN CHARACTER (0D HEX).-,/)

THE LEVEL II DATA BLOCK ITSELF CAN CONTAIN A MAXIMUM OF 249 BYTES, INCLUDING ANY COMMAS USED AS SEPARATORS. IT IS IMPORTANT TO REALISE THAT THIS FORMAT IS ALWAYS USED FOR DATA TAPES, WHETHER THE DATA IS JUST ONE BYTE OR THE FULL ALLOWANCE OF 249 BYTES. EVERY TIME A BLOCK OF DATA IS WRITTEN TO TAPE, THERE IS AN AUTOMATIC OVERHEAD OF 257 BYTES OF LEADER, SYNC BYTES, ETC. AT THE SYSTEM'S STANDARD 500 BAUD TRANSFER RATE, THIS OVERHEAD TAKES APPROXIMATELY FOUR SECONDS TO WRITE. (A 'BAUD' RATE IS A MEASURE OF HOW FAST DATA IS SENT ALONG A COMMUNICATIONS LINK - FOR OUR PURPOSES, IT IS THE SAME AS BITS/SEC'; I.E. '500 BAUD' MEANS 500 BITS/SEC.) THE TRS-80 MODEL III HAS A FASTER DATA TRANSFER RATE OF 1500 BAUD, BUT STILL USES THE SAME FORMAT.

YOU CAN SEE THAT, EVEN IF WE USE THE FULL 249 BYTE ALLOWANCE EVERY TIME, ONLY 49.2% OF THE TAPE ACTUALLY CONTAINS DATA. IN THE WORST CASE, WRITING SINGLE BYTES TO TAPE ONLY 99.6% OF THE TAPE IS FULL OF MATERIAL THAT ACTUALLY CONTRIBUTES NOTHING TO WHAT WE ARE TRYING TO SAVE. OUR FIRST THOUGHTS ON MAKING A MORE EFFECTIVE USE OF TAPE FOR DATA STORAGE SHOULD OBVIOUSLY CONCERN THE REDUCTION OF THIS OVERHEAD. UNFORTUNATELY, THE LEVEL II DATA RATE AND TAPE FORMAT, LIKE THOSE USED BY ALL MICROS, ARE CONTROLLED BY THE COMPUTER'S OPERATING SYSTEM AND ARE NORMALLY BURIED IN THE DEPTHS OF THE MICRO'S ROM. THE ONLY WAY TO ALTER THEM IS TO DELVE INTO THE SYSTEM'S MACHINE CODE, WRITING NEW CASSETTE READ AND WRITE ROUTINES TO SUIT YOUR PARTICULAR PURPOSES. FOR MOST PEOPLE, THIS SORT OF SOLUTION IS NOT REALISTIC, AND SO WE HAVE TO PUT UP WITH WHATEVER THE COMPUTER'S DESIGNERS HAVE DEEMED AS APPROPRIATE.

IF WE WANT TO MAKE MORE EFFECTIVE USE OF TAPE STORAGE, THE ONLY WAY IS TO GET AS MUCH DATA AS POSSIBLE IN TO EACH BLOCK WE WRITE. THAT, IN TURN, MEANS USING THE FULL ALLOWANCE (249 BYTES IN THIS CASE REMEMBER) AND PACKING WHATEVER WE ARE TRYING TO SAVE INTO THAT ALLOWANCE AS EFFICIENTLY AS POSSIBLE. FOR NOW I WILL STICK TO STRING STORAGE ONLY. NUMBER ARE HANDLED IN A DIFFERENT MANNER AND IF THERE'S ANY INTEREST, I'LL PRODUCED ANOTHER ARTICLE FEATURING JUST NUMBER HANDLING.

EFFICIENT STRING STORAGE

SUPPOSE THAT WE HAVE A PROGRAM WHICH HANDLES LISTS OF ONE SORT OR ANOTHER - A SIMPLE DATA BASE SYSTEM TO KEEP A RECORD OF WHO SENT YOU CHRISTMAS CARDS LAST YEAR, SAY. IN A CASE LIKE THIS, THE STRINGS REPRESENTING THE DATA WILL PROBABLY BE STORED IN AN ARRAY - CALL IT DT$(). THE EASIEST WAY OF STORING THAT DATA ON TAPE IS BY A ROUTINE OF THE FORM

```
 9999 REM ** STORE DATA TO
TAPE
10000 FOR I=1 TO N:REM ** N
ITEMS
10010 PRINT#-1,DTS(I)
10020 NEXT
10030 REM ** CONTINUE
```

UNFORTUNATELY, LIKE MOST 'EASIEST WAYS', THIS ROUTINE IS A VERY INEFFICIENT. WHY? BECAUSE EVERY TIME WE USE PRINT * - 1.... A NEW 255-BYTE HEADER, ETC, IS WRITTEN TO TAPE. CERTAINLY, THE DATA IS SAVED, BUT MOST OF THE BYTES ACTUALLY ON THE TAPE ARE OF NO DIRECT USE TO US. WE HAVE TO FIND A WAY OF PACKING THE DATA ONTO TAPE AS FULLY AS POSSIBLE. THERE ARE TWO OPTIONS HERE: A. IF EVERY ITEM OF DATA HAS THE SAME LENGTH. B. IF EACH DATA ITEM'S LENGTH IS UNPREDICTABLE.

PACKING UNIFORM DATA
THE VERY EASIEST CASE ARISES WHEN WE KNOW AT THE TIME OF WRITING THE PROGRAM HOW LONG EACH DATA STRING WILL BE. SUPPOSE THAT WE KNOW WE CAN PACK 'P' ITEMS INTO 249 BYTES, WE CAN THUS MODIFY THE PROGRAM FRAGMENT ABOVE TO BECOME:

```
 9999 REM ** STORE DATA TO
TAPE
10000 FOR I=1 TO N STEP
P:REM ** N ITEMS
10010 PRINT#-
1,DT$(I),DT$(I+1),...,
DT$(I+P-1):REM ** SAVE P
ITEMS
10020 NEXT
10030 REM ** CONTINUE
```

WE SIMPLY MODIFY THE PRINT # - 1, TO INCLUDE THE CORRECT NUMBER OF DATA ITEMS.
THAT'S TOO EASY THOUGH. LET'S STICK WITH THE FIXED-SIZE DATA STRINGS, BUT ASSUME THAT THE SIZE IS ACTUALLY FIXED BY THE PROGRAM. THIS MEANS THAT WE CANNOT DEFINE IN ADVANCE THE NUMBER OF ITEMS WE WILL WRITE. THAT, IN TURN, MEANS WE CANNOT USE A MULTIPLE-ITEM STATEMENT LIKE THAT IN LINE 10010 ABOVE, BUT MUST WRITE A SINGLE STRING, AS NEAR TO 249 BYTES LONG AS POSSIBLE, TO TAPE EACH TIME. LISTING 1 IS A SUBROUTINE TO DO JUST THAT JOB. IT SHOULD BE CLEAR HOW IT WORKS AND, LIKE ALL THESE SUBROUTINES, IT USES VARIABLES BEGINNING WITH 'Z' AS 'LOCAL VARIABLES'. IF YOU AVOID Z-VARIABLES ELSEWHERE, YOU CAN USE MY OFFERINGS IN YOUR OWN PROGRAMS.
THE ONLY SIGNIFICANT POINT TO NOTE IN THE LISTING IS LINE 10040, WHICH MAKES SURE THAT THERE ARE ENOUGH STRINGS LEFT TO COMPLETELY FILL ZA$. IF THERE ARE NOT, IT FORCES A REDUCED-LENGTH FINAL SAVE; IN THIS WAY, WE AVOID TRYING TO ACCESS POSSIBLY NON-EXISTENT POSITIONS IN DT$() AT LINE 10060.
THAT'S GOOD - WE ARE SAVING DATA TO TAPE AS EFFECTIVELY AS WE CAN. HOW DO WE GET IT BACK AGAIN? LISTING 2 IS THE ANSWER. AGAIN, THE LISTING SHOULD BE FAIRLY CLEAR. LINE 11020 EXTRACTS THE TOTAL

The Route to Fast and Easy Pl



Do Away With Slow, Messy Spreadsheets . . .

Planning and decision-making are what being a manager is all about. But they can also be a manager's biggest headache. Especially when they involve complex spreadsheets. So many entries, so many variables — many dependent on one another. Working with pencil and paper can quickly become a messy job. Even an electronic calculator only helps with the arithmetic. Change anything and you have to refigure and re-enter all the related numbers.



. . . or Expensive Time-Share Systems



So, computerize it. Sure, but that's not always the ideal solution. Time-sharing is expensive and the computer may not be available when you need it most. Even if your company has its own mainframe system you may have a long wait.

The answer? TRS-80 with VisiCalc. You get the speed of computerized analysis, personal control of your own worksheet and no hourly or monthly fees!

A New Kind of Tool for Business Planners

If what you need is a quick and accurate method of working with the data for budgets, cost projections, financial statements, production schedules, market predictions or engineering problems, then what you need is VisiCalc. VisiCalc is a computer program for business-people who aren't computer experts and don't care to be, but who are responsible for important spreadsheet planning and decisions.

How does VisiCalc do it? By turning your TRS-80 computer into a giant "electronic worksheet" that can be up to 63 columns wide or 255 rows deep.

Electronic Worksheet — Fast, Accurate, Neat!

You enter your data just once. But where you would normally calculate an entry, you enter the formula for calculating that number. Instead of hours of figuring, erasing, copying and shuffling from sheet to sheet, with VisiCalc you begin immediately to analyze your information.

Every time you enter a new piece of data, VisiCalc instantly and automatically refigures all related entries and totals, and then displays the results on your screen. Add a line printer and produce a report in seconds.

"What if . . ." Questions Become a Snap

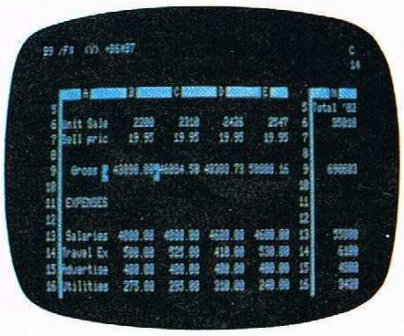
Raw materials, wages, transportation may go up, but production improvements may cut unit costs. What if you pay cash for that new piece of equipment — or lease it? Just plug in a number or two and see the effect for a whole year. Or five years.

anning—TRS-80 with VisiCalc



The Planner's Dream: A Window on the Future

VisiCalc is your "crystal ball," a window through which you can judge the future effect of today's decisions. Because of its speed and accuracy, you spend less time "pencil-pushing" and more time managing, and that can be the critical edge in a competitive world. Once your spreadsheet or model is set up, you can store it on diskettes for updating or review any time in the future.



Screen splits horizontally or vertically to allow you to watch results in one area while working in another.

Easy to Use—No "Computerese"

Best of all, VisiCalc is designed for business people. The manual is written in down-to-earth English rather than computer jargon. It's so easy to learn you can teach yourself and begin seeing results the same day—a budget analyst for a major newspaper did just that.

In Use by Many Managers . . . Right Now!

VisiCalc is already helping many companies both large and small. For example, the research arm of a major corporation uses VisiCalc for budgetary decision-making, allocation of resources, comparison of performance to projection, personnel ratings—even for the allocation of space within a new building.

A smaller company uses VisiCalc to speed and simplify its complicated billing procedures.

A giant manufacturing firm began using VisiCalc and found that analysts, relieved of spreadsheet drudgery, were enjoying their work again. At the same time, demands for typing were significantly reduced.

How to Get Relief From Planning Headaches

It's easy. All you need is a TRS-80 Model I, II, or III microcomputer and the VisiCalc program, plus a line printer to give you hard copy when you've got the results you want.

Planning and analysis will never be the same.

Specifications and Special Functions

Accuracy: 11 digit in conventional values, integers or dollars and cents. *Capacity:* 63 columns wide, 255 rows deep (actual capacity will vary depending on the model of TRS-80 used and the size of individual elements in the spreadsheet). *Graphing capacity:* Simple bar graphs. *Fast formula extension:* Automatic replication. *Business Functions:* SUM (list)—sum of all values in list (all or part of a row or column); MIN or MAX (list)—smallest or largest value in list; COUNT (list)—number of non-blank entries in list; AVERAGE (list)—average of non-blank entries in list; NVP—net present value of cash flow in specified range; LOOKUP—find value in table. *Engineering Functions:* SIN, COS, TAN, ASIN, ACOS, ATAN, ABS, LOG, LN, PI, EXP and SQRT.

NUMBER OF READS FROM TAPE WHICH THE ROUTINE MUST MAKE; IT SHOULD WORK CORRECTLY UNLESS THERE ARE MORE THAN 1000 ITEMS ON THE TAPE. LINE 11070 ACTUALLY EXTRACTS THE DATA; IT USES THE MID$() FUNCTION, (PRESENT IN MOST MICROSOFT-TYPE BASICS).

VARIABLE LENGTH DATA

THE SORT OF ROUTINES I HAVE JUST DESCRIBED WILL DO THE JOB BUT SOMETIMES WE HAVE TO SAVE STRINGS OF DIFFERENT, UNPREDICTABLE LENGTHS. SINCE THE OBJECT OF THE EXERCISE IS STILL TO MAKE THE BEST USE OF TAPE STORAGE, WE HAVE TO ARRANGE TO:

A. FILL UP THE AVAILABLE SPACE AS MUCH AS POSSIBLE.
B. NOT OVERFLOW THE AVAILABLE SPACE.
C. RECOVER THE DATA AFTERWARDS.

FILLING UP THE SPACE IS EASY - SIMPLY MEASURE THE SPACE AVAILABLE AND SEE IF THE DATA STRING WHICH COMES NEXT WILL FIT. IF IT WILL, JOIN IT ONTO THE 'TAPE' STRING WE ARE PREPARING; IF IT WON'T, WRITE WHAT WE HAVE TO TAPE AND START A NEW TAPE STRING.

THE PROBLEM COMES WHEN WE HAVE TO GET THE DATA BACK AFTERWARDS. IN THIS CASE, SINCE WE DO NOT KNOW HOW LONG EACH STRING IS, WE MUST READ IT ONE CHARACTER AT A TIME. IN TURN, THIS IMPLIES THAT WE CAN IDENTIFY THE END OF EACH ITEM. I FIND THE BEST WAY IS TO USE A SEPARATOR BETWEEN EACH ITEM; THE SEPARATOR MUST ITSELF BE A CHARACTER, BUT ONE WHICH IS NOT LIKELY TO BE IN ANY OF THE DATA STRINGS. A GOOD ONE TO USE IS CHR$(127), WHICH IS NOT NORMALLY FOUND ON MICRO KEYBOARDS.

WITH THESE CONSTRAINTS, LOOK AT LISTING 3, WHICH SHOWS A WAY OF SAVING VARIABLE-LENGTH STRINGS TO TAPE.

THE ROUTINE IS, ONCE AGAIN, STRAIGHTFORWARD. ZL IS USED TO KEEP TRACK OF HOW MUCH SPACE IS AVAILABLE IN THE DUMMY STRING - ITS INITIAL VALUE IS 248, NOT 249, TO ALLOW FOR THE SEPARATOR WHICH WILL BE AT THE END OF THE LAST ITEM IN EACH BLOCK OF DATA WRITTEN TO TAPE. THE CODE IN LINES 12110 AND 12120 TAKES CARE OF THE FACT THAT, WHEN THE ROUTINE LEAVES ITS MAIN LOOP, THERE MAY STILL BE SOME UNWRITTEN DATA BECAUSE THE FINAL ZA$ WAS NOT FULL. IT ALSO CHECKS TO SEE IF THERE IS ANYTHING IN ZA$ AND, IF THERE IS, WRITES IT TO TAPE.

SO, WE HAVE GOT THE DATA ONTO THE TAPE, ALL WE HAVE TO DO NOW IS GET IT BACK AGAIN! LISTING 4 IS ONE SOLUTION TO THE PROBLEM. THE KEY LINES IN THIS SUBROUTINE ARE:

13050, 13060. THESE LINES LOOK FOR A SEPARATOR, AND JUMP PAST THE CONCATENATION ROUTINE IF ONE IS FOUND. EACH CHARACTER IS READ INTO THE DUMMY ZP$ TO SPEED UP AN IN-HERENTLY SLOW PROCESS.

13100. THIS LINE DECIDES WHETHER THERE IS ANY MORE DATA IN THE EXISTING ZA$ AND IF THERE IS MORE DATA TO READ. IF NECESSARY, IT FORCES ANOTHER READ FROM TAPE AND REINITIALIZES Z2 TO 0.

13140. LINE 13140 INCREMENTS THE POINTER PAST THE SEPARATOR AND ADVANCES THE ZEROED Z2S TO THEIR STARTING VALUES OF 1.

PROBLEMS

THAT LOOKS PRETTY GOOD BUT, IN LEVEL II BASIC AT LEAST, THERE ARE STILL TWO PROBLEMS LEFT. YOU WILL RECALL THAT WE ARE TRYING TO PACK DATA INTO SINGLE, MAXIMUM-LENGTH STRINGS AND WRITE THEM TO

TAPE. WHAT HAPPENS IF THE DATA CONTAINS COMMAS OR COLONS ':'?
THE SYSTEM TREATS THESE AS INPUT SEPARATORS AND, IF IT IS LOOKING FOR A SINGLE STRING (INPUT#-1,ZA$), STOPS AS SOON AS IT READS A COMMA OR A COLON. IF THAT HAPPENS TO BE THE FIFTH CHARACTER OF A 240-BYTE STRING, TOUGH - YOU JUST GET THE FIRST FOUR CHARACTERS, AN 'EXTRA IGNORED' ERROR MESSAGE AND A PROGRAM CRASH!
THE WAY OUT OF THIS CORNER IS TO ENCLOSE ZA$ IN QUOTES WHENEVER IT IS WRITTEN TO TAPE. YOU CAN DO THIS WITH CODE OF THE FORM:
PRINT#-1,CHR$(34)
+ZA$+CHR$(34) QUOTES ARE REPRESENTED BY THE ASCII CODE '34'. IF YOU DO THIS, THE QUOTES WILL BE WRITTEN TO TAPE (AND SO YOU ONLY HAVE 247 BYTES TO PLAY WITH EACH TIME), BUT WILL NOT BE READ BACK. HOWEVER, ALL THE COMMAS AND COLONS WILL BE READ CORRECTLY.
JUST A MINUTE THOUGH - WHAT IF ZA$ ITSELF INCLUDES A QUOTE? THE TAPE READ OPERATION WILL STOP WHEN IT REACHES THE EMBEDDED QUOTE AND WILL NEVER REACH THE TERMINATING ONE. AT BEST, THIS WILL RUIN THE DATA INPUT BUT THE MOST LIKELY RESULT WILL BE ANOTHER PROGRAM CRASH.

THERE IS A WAY OUT OF THIS PROBLEM TOO. FORGET ABOUT QUOTES - REMEMBER THAT WE ARE SAVING CHARACTERS IN THE FORM OF STRINGS. USUALLY, WE WILL BE SAVING THE NORMAL TEXT CHARACTERS WHICH HAVE ASCII VALUES FROM 32 TO 125. HOWEVER, A BYTE CAN CONTAIN VALUES UP TO 255. WE CAN THEREFORE OFFSET EACH CHARACTER IN ZA$ BY ADDING 128 TO IT BEFORE WRITING IT TO TAPE AND STRIPPING 128 FROM IT WHEN WE READ IT BACK. THIS APPROACH SLOWS DOWN THE READ/WRITE ROUTINES, BUT IT DOES ALLOW US TO SAVE ANY NORMAL CHARACTER ON TAPE.
HOW TO DO IT? LISTING 5 IS A SUBROUTINE TO ADD 128 TO THE ASCII VALUE OF EACH CHARACTER IN ZA$; IF YOU INSERT GOSUB 2000 AS LINE 10075 IN LISTING 1 AND AS LINES 12035 AND 12105 IN LISTING 3, THAT SHOULD DO THE TRICK.
TO RECOVER THE DATA, THERE ARE TWO POSSIBILITIES. IF YOU ARE USING THE LISTING 2 APPROACH, USE ANOTHER SUBROUTINE WITH THE FORMAT OF LISTING 1 BUT WITH LINE 20030 REPLACED BY:
20030 ZB$=ZB$+CHR$(ASC (MID$(ZA$,Z9, 1))-Z8)
IF, HOWEVER, YOU ARE USING THE VARIABLE-LENGTH RECORD APPROACH OF LISTING 4, THE ANSWER IS RATHER EASIER. SIMPLY RE-WRITE LINE 13050 AS:
13050 ZB$=CHR$(ASC (MID$(ZA$,Z2,1))-128)
REMEMBER THAT OFFSETTING IS SLOW. IF YOU CAN GUARANTEE NO PROBLEMS WITH COMMAS, COLONS AND QUOTES, DON'T USE IT.

CONCLUSIONS
AND THERE WE HAVE IT - WAYS OF MAKING THE BEST USE OF TAPE STORAGE FOR STRING HANDLING. I'VE DESCRIBED SUBROUTINES TO PACK FIXED- AND VARIABLE-LENGTH STRINGS TO TAPE, AND READ THEM BACK AGAIN. BECAUSE OF THE WAY THAT LEVEL II BASIC WORKS, THERE IS A GREAT DEAL OF FIDDLING ABOUT WITH INDIVIDUAL CHARACTERS IN STRINGS, WHICH INEVITABLY SLOWS THINGS DOWN A BIT. NEVERTHELESS, IT IS STILL AN MASSIVE ADVANCE ON THE 'OBVIOUS' APPROACH OF WRITING EACH ITEM OF DATA TO TAPE IN SPLENDID ISOLATION WHICH WASTES A LOT OF TIME AND TAPE.

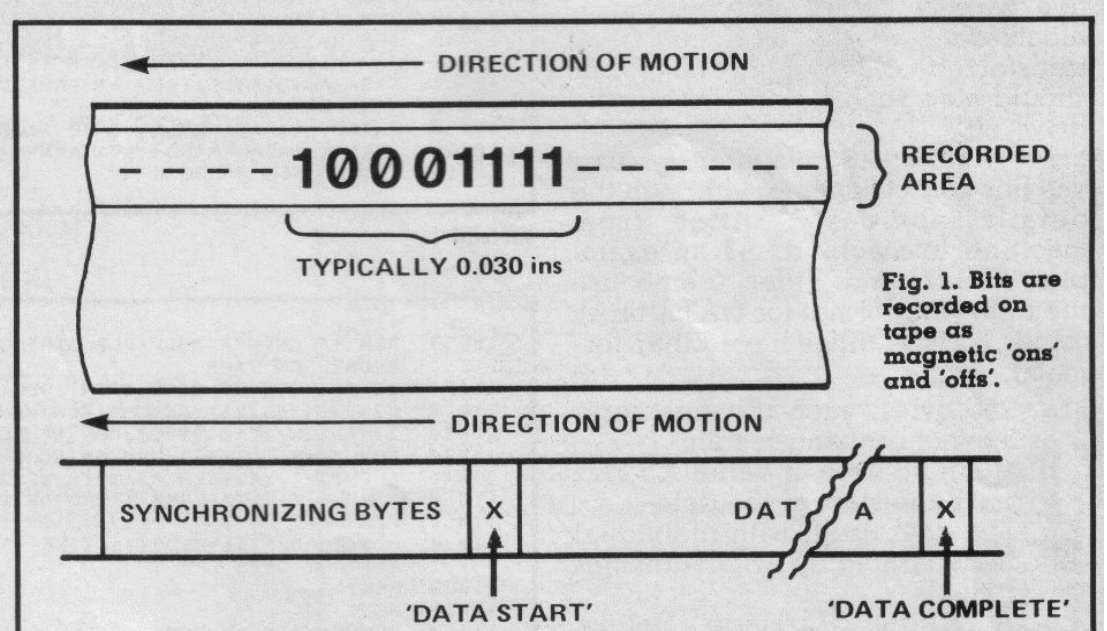


Fig. 1. Bits are recorded on tape as magnetic 'ons' and 'offs'.

Fig. 2. The format of block storage on tape showing the start and stop information.

```
 9999  REM ** PACK MULTIPLE STRINGS INTO A SINGLE STRING
       ON TAPE
10000  Z1=LEN(DT$(1)):REM ** GET LENGTH OF EACH ITEM
10010  Z2=INT(249/Z1):REM ** HOW MANY CAN WE PACK?
10020  FOR Z3=1 TO N STEP Z2
10030  ZA$="":REM CLEAR DUMMY STRING
10039  REM ** ENOUGH TO FILL THE STRING?
10040  IF (N-Z3)>Z2 THEN Z4=Z2 ELSE Z4=N-Z3+1
10050  FOR Z5=Z3 TO Z3+Z4-1:REM ** LOOP Z4 TIMES
10060  ZA$=ZA$+DT$(Z5):REM ** PACK DUMMY STRING
10070  NEXT Z5
10080  PRINT#-1,ZA$:REM ** SAVE PACKED STRING
10090  NEXT Z3:REM ** DO IT AGAIN
10100  RETURN:REM ** DONE
```

Listing 1.

```
10999  REM ** UNPACK MULTIPLE STRINGS FROM A SINGLE ONE
       STORED ON TAPE
11000  Z1=[LENGTH OF EACH ITEM]:REM ** GET ITEM LENGTH
11010  Z2=INT(249/Z1):REM ** HOW MANY IN EACH TAPE STRING
11020  Z3=INT(N/Z2+0.9999):REM ** NO OF BLOCKS ON TAPE
11030  FOR Z4=0 TO Z3-1:REM ** LOOP Z3 TIMES
11040  INPUT#-1,ZA$:REM ** THIS IS A LEVEL II COMMAND
11050  Z5=LEN(ZA$)/Z1:REM ** NO OF ITEMS IN THIS STRING
11060  FOR Z6=1 TO Z5
11070  DT$(Z4*Z2*Z6)=MID$(ZA,((Z6-1)Z1)+1,Z1):REM **
       EXTRACT DATA
11080  NEXT Z6
11090  NEXT Z4
11100  RETURN:REM ** DONE
```

```
11999  REM ** PACK VARIABLE LENGTH STRINGS
12000  Z1=248:REM ** RECORD OF SPACE LEFT
12010  ZA$="":REM ** CLEAR DUMMY STRING
12020  FOR Z2=1 TO N
12030  IF LEN(DT$(Z2))<=Z1 THEN 12070:REM ** IS THERE
       ROOM?
12040  PRINT#-1,ZA$:REM ** NO ROOM - SAVE IT
12050  ZA$="":REM ** CLEAR IT AGAIN
12060  Z1=248:REM ** RE-INITIALISE COUNTER
12070  ZA$=ZA$+DT$(Z2)+CHR$(127):REM ** ADD DATA AND
       SEPARATOR
12080  Z1=Z1-LEN(DT$(Z2))-1:REM ** ADJUST COUNTER
12090  NEXT Z2:REM ** BACK FOR MORE
12099  REM ** AT END, SEE IF ANYTHING LEFT
12100  IF Z1=248 THEN 12120
12110  PRINT#-1,ZA$:REM ** THERE WAS - SAVE IT
12120  RETURN:REM ** DONE
```

Listing 3.

```
12999  REM ** UNPACK VARIABLE LENGTH STRINGS FROM TAPE
13000  INPUT#-1,ZA$:REM ** GET FIRST STRING
13010  Z1=LEN(ZA$)-1:REM ** HOW MANY CHARACTERS?
13020  Z2=1:REM ** POINTER TO CHARACTERS
13030  FOR Z3=1 TO N
13040  DT$(Z3)="":REM ** INITIAL VALUE
13050  ZP$=MID$(ZA$,Z2,1):REM ** CHARACTER POINTED TO
13060  IF ZP$=CHR$(127) THEN 13100:REM ** SEPARATOR?
13070  DT$(Z3)=DT$(Z3)+ZP$:REM ** ADD CHARACTER TO STRING
13080  Z2=Z2+1:REM ** INCREMENT POINTER
13090  GOTO 13050:REM ** GET ANOTHER CHARACTER
13100  IF Z2<Z1 OR Z3=N THEN 13140:REM ** IS THERE MORE
       TO READ?
13110  INPUT#-1,ZA$:REM ** IF SO, READ ANOTHER STRING
13120  Z1=LEN(ZA$)-1
13130  Z2=0:REM ** NEXT LINE WILL INCREMENT POINTER
13140  Z2=Z2+1:REM ** GO PAST START OR SEPARATOR
13150  NEXT Z3:REM ** GET NEXT DT$()
13160  RETURN:REM ** DONE
```

Listing 4.

```
19999  REM ** ADD 128 TO EACH CHARACTER IN ZA$
20000  Z8=128:REM ** CONSTANT
20010  ZB$="":REM ** NEW DUMMY
20020  FOR Z9=1 TO LEN(ZA$)
20030  ZB$=ZB$+CHR$(Z8+ASC(MID$(ZA$,Z9,1))):REM **
       ACTUALLY DO IT
20040  NEXT Z9
20050  ZA$=ZB$:REM ** RESTORE ZA$
20060  RETURN:REM ** DONE
```

Listing 5.

LEVEL 3 REVIEW

NIGEL DIBBEN

LEVEL 3 BASIC IS WRITTEN BY THE AUTHORS OF LEVELS 1 AND 2 (BILL GATES ØF MICROSOFT) AND PROVIDES THREE BROAD FUNCTIONS:

1. USE OF COMMANDS THAT ARE OTHERWISE AVAILABLE ONLY ON DISC : (USR FN, TIMES$ &H, LINETNPUT,MID$ ETC)
2. NEW INSTRUCTIONS AND COMMANDS FOR GRAPHICS AND FOR BASIC RENUMBERING, RS232 OUTPUT ETC.
3. IMPROVED OPERATING; SYSTEM WITH REDUCED LOAD PROBLEMS, SINGLE KEY WORD ENTRY, KEYBOARD DEBOUNCE, LONG ERROR MESSAGES AND RESET VIA BREAK.

TO THE USER, ESPECIALLY WITH AN EXPANSION INTERFACE, THE ADDITIONAL COMMANDS MEAN THAT SOME OF THE FACILITIES OTHERWISE RESTRICTED TO DISC ONLY ARE AVAILABLE AT A FRACTION OF THE COST. THESE COMMANDS ARE AS LISTED ON PAGE 1Ø4 OF THE REFERENCE MANUAL. THIS IS PARTICULARLY TRUE IN THE CASE OF TIMES$ WHICH IS PHYSICALLY PRESENT IN THE EXPANSION INTERFACE BUT UNATTAINABLE TO THE NON-DISC-OWNER. THE GRAPHIC COMMANDS THAT ARE NEW TO BASIC DO NOT FUNDAMENTALLY DO ANYTHING THAT COULD NOT BE LABORIOUSLY CONTRIVED IN BASIC. HOWEVER, AFTER WATCHING LEVEL 3 COVER THE SCREEN WITH X'S OR FLASH PAST OF IT IN MILLISECONDS INSTEAD OF 'MAXI' SECONDS, YOU SOON APPRECIATE THE DIFFERENCE.

THE OPERATIONAL SYSTEM CHANGES ARE A CERTAIN IMPROVEMENT ESPECIALLY THE SHIFT KEY ENTRIES OF WHICH 26 ARE SET BY THE PROGRAMME (SHIFT A TO SHIFT Z) THESE CAN ALL BE CHANGED BY THE USER SO THAT FOR EXAMPLE:

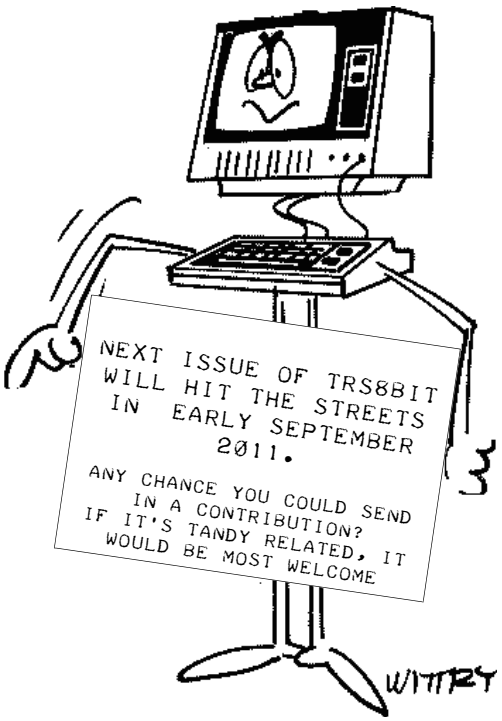
```
LSET C= "CONT"
+ CHR$(13)
```

WILL MEAN THAT EVERY TIME SHIFT C IS PRESSED, THE WORD CONT WILL APPEAR AND THE COMPUTER WILL CONTINUE. INTERESTINGLY, ONE CAN MAKE UP GRAPHIC WORDS IN THIS WAY. FOR EXAMPLE:

```
LSET A=
CHR$(191)+
CHR$(140)
+CHR$(191)
```

WILL DISPLAY AND ASSIGN: WHENEVER SHIFT A IS PRESSED. RESET THROUGH BREAK MEANS THAT TAPES CAN BE STOPPED DURING LOAD AND SAVE OPERATIONS. THUS, AN INCORRECT TAPE CAN BE STOPPED LOADING WITH THE INTERFACE, THE USE OF BREAK DOES NOT LOSE THE BASIC PROGRAMME - A THOROUGHLY USEFUL NIECE OF SOFTWARE

TO LOOK ON THE OTHER SIDE, THERE ARE NATURALLY SOME DISADVANTAGES: THE FIRST IS



THAT LEVEL 3 RESIDES FROM 4300H TO 57E0H SO CANNOT BE USED WITH 4K AND LEAVES THE 16K USER WITH ONLY 10K OF STORE. EVEN THE ECONOMIES IN LEVEL 3 CAN NEVER RECOUP THE LOST 6K. EXTRA MEMORY IS THEREFORE DESIRABLE.
SECONDLY, WHILE LEVEL 2 PROGRAMMES RUN IN LEVEL 3, PROGRAMMES WITH LEVEL 3 FUNCTIONS IN THEM WILL NOT RUN IN LEVEL 2 (OR, I BELIEVE TRSDOS BASIC). THERE IS HOWEVER NO PROBLEM IN USING LEVEL 3 TO WRITE, EDIT AND SAVE A PROGRAM WITHOUT SPECIAL FUNCTIONS AND THEN RUNNING IT IN LEVEL 2.
THIRDLY THERE IS A DISC LEVEL 3 SUPPLIED ON THE CASSETTE BUT THIS IS STILL SUPPLEMENTAL TO LEVEL 2 AND NOT TO DOS BASIC. LASTLY, A PERSONAL POINT, RENUMBERING LINES WILL NOT ECONOMISE ON SPACE IN PROGRAMME TEXT AS EVERY LINE NUMBER IS ASSIGNED 5 BYTES OF WHICH THE BLANKS WOULD HAVE TO BE EDITED OUT; ALSO, PRINT STATEMENTS CONTAINING ENTER BYTES (I.E.: ASCII '13') ARE MESSED UP BY THE RENUMBER ROUTINE. BEWARE!!

IN SUMMARY, WHILST THERE ARE DISADVANTAGES, THERE ARE BY FAR OUTWEIGHED BY THE ADVANTAGES AND, IN NO SMALL DEGREE, BY A GOOD REFERENCE BOOKLET AND QUICK REFERENCE CARD COVERING ALL LEVEL 2 AND LEVEL 3 COMMANDS, INSTRUCTIONS, ERROR CODES, CONTROL CHARACTERS ETC. IF YOU ARE EXPANDING SLOWLY, YOU WILL PARTICULARLY APPRECIATE SPENDING ABOUT 1/10TH OF THE COST OF A DISC UNIT FOR AT LEAST 1/2 THE ADVANTAGES.

--== o0o ==--

(CONTINUED FROM PAGE 1)
TO READ THOUGH. UNTIL I CAN GET MY HANDS ON SOME TECHNICAL INFORMATION, I'M LOATHED TO DO TOO MUCH POKING ABOUT.

EBAY, AS USUAL, HAS BEEN BUSY OVER THE LAST 3 MONTHS. (THERE'S A M4, ONLY 28 MILES AWAY JUST BEING RE-LISTED, I HAVEN'T MENTIONED IT TO DEE, YET!) I DON'T THINK I DARE SPEND £99 ON A M4, NO MATTER HOW NICE IT LOOKS IN THE PHOTO!
VARIOUS M1'S AND M4'S HAVE BEEN SOLD, WITH M1'S FETCHING AROUND THE £50 MARK.

QUITE A BIT OF SOFTWARE HAS BEEN FOR SALE TOO. ONE ARTICLE I NOTICED BEING OFFERED UP FOR SALE, WITH A STARTING PRICE OF £4.99, WHICH CAUGHT MY EYE WAS A COPY OF MICROSOFT'S LEVEL 3 BASIC.
I REMEMBERED THAT NIGEL DIBBEN WROTE A REVIEW OF L3 SO I'VE INCLUDED IT IN THIS EDITION, JUST TO REFRESH OUR MINDS. THIS FIRST APPEARED, BELIEVE IT OR NOT, IN NATGUG NEWS, 32 YEARS AGO.

THERE IS ALSO A COMPREHENSIVE ARTICLE ON DATA STORAGE BY DS PECKETT WHICH I HOPE YOU ENJOY.

WELL I THINK THAT ABOUT WRAPS UP THIS ISSUE. IF YOU WOULD LIKE TO CONTRIBUTE AN ARTICLE, PLEASE DON'T HOLD BACK, ANYTHING WHICH IS TANDY RELATED WOULD BE MOST WELCOME.

BY FOR NOW
DUSTY





Genie III...

your third and greatest wish!

The new Genie III surpasses even its predecessors by harnessing all the benefits of modern, professional microcomputing in one unit.

Versatility.

Software availability is exceptionally good – in fact the Genie III has more software available than any of its competitors. And even if you don't want to use a ready made business package, with the Genie III you can write your own! The Genie III is fully operational in either CP/M, or level II basic. It has a built-in screen, 64K RAM, and dual disk drive with 730K storage capacity, all incorporated in its compact and attractive casing.

Speed.

The new Genie III is an obedient slave in the office, quickly and quietly taking care of all your routine work, from printed invoices, mailing lists and letters to stock control.

Economy and simplicity.

It all costs less than a new office junior, yet it's simple enough to operate, and comes complete with a teaching programme to show you how.

Reliability.

Your Genie III is reliable too, manufactured to the most stringent standards with top quality components. And even in the unlikely event that your Genie should ever temporarily fail to please you are assured of very fast, competent service because the Genie III is only available from our specially selected SUPERDEALERS. Dealers whose technical expertise and experience in the fields of installation, software availability and repair is proven.

Software.

Finally, it's compatibility with CP/M or Level II BASIC means there is more Software available for Genie III than any other comparable system!

Please send me FREE, 16 page colour brochure on the Genie Computer System. I enclose 25p postage.

Name _____

Address _____

_____

Telephone _____ CT ·82

LOWE electronics

Chesterfield Road,
Matlock,
Derbyshire DE4 5LE.

Telephone: 0629 4995.
Telex: 377482 Lowlec G.

TRS8BIT



WELCOME TØ THE SEPTEMBER 2Ø11 EDITION!

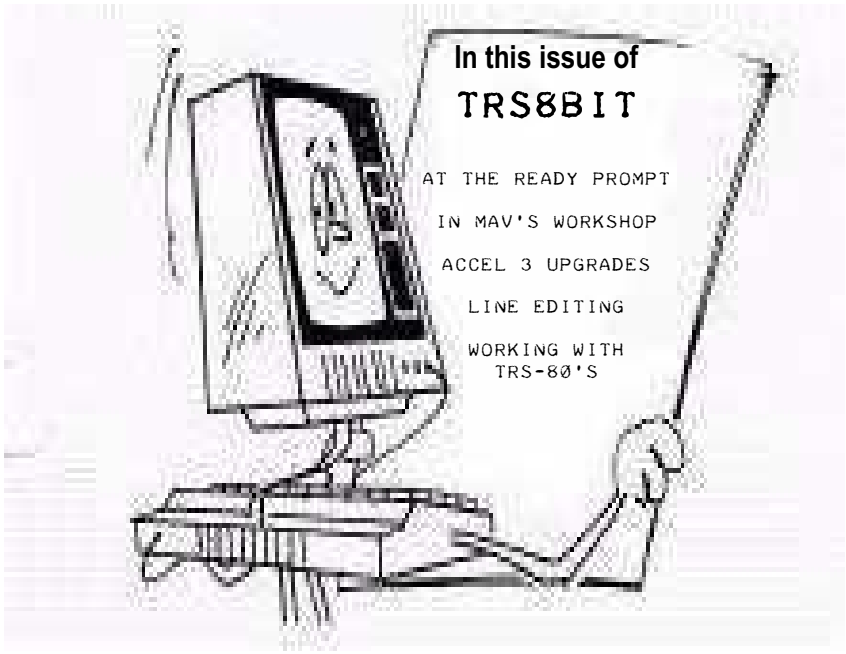
ONCE AGAIN, I HAVE TO START OFF AN EDITION OF TRS8BIT WITH SAD NEWS. IT IS WITH MUCH REGRET I HAVE TO INFORM YOU ALL OF THE DEATH OF ROY T BECK. ROY WAS BEST KNOW TO MOST OF US IN THE U.K. THROUGH HIS MANY EXPERT ARTICLES PRODUCED FOR TRSTIMES, IN THE 199Ø'S. OUR CONDOLENCES GO TO HIS FAMILY AND MANY FRIENDS.

WHAT A SUMMER THIS HAS BEEN FOR ANYONE IN THE TRS-8Ø WORLD. PRICES ON EBAY, UK, SEEM TO HAVE GONE BALLISTIC IN EVERY SENSE OF THE WORD! MODEL 1'S AND VIDEO GENIE'S FETCHING OVER £3ØØ. ALSO, SEE IAN MAVRIC'S ARTICLE ON PAGE 2, A NON-WORKING MODEL 1 IN AUSTRALIA, FETCHING OVER $A45Ø!
I FIND IT RATHER SAD, AND HOPE THAT OUR INTERESTS WILL NOT BE TAKEN OVER BY 'SILLY MONEY' PEOPLE TOGETHER WITH THOSE TRYING TO PANDER TO THE IDEA OF 'GOOD INVESTMENTS' AND QUICK PROFIT MAKING IDEAS.
IAN SEEMS TO BE WORKING HARD AT KEEPING TANDYS' UP AND RUNNING IN AUSTRALIA. THERE'S A LINK TO HIS WEBSITE, 'THE RIGHT STUFF' ON OUR SITE IF YOU WISH TO USE IT.
IS THERE ANYONE IN THE UK WHO IS OFFERING A SIMILAR SERVICE? IF SO, PLEASE LET US ALL KNOW. I'LL BE MORE THAN HAPPY TO PUBLISH CONTACT DETAILS FOR YOU.
I'VE ALSO NOTICED THAT 5 MODEL 3'S HAVE APPEARED FOR SALE RANGING FROM £1ØØ TO £19Ø, BUY-IT-NOW. IT WILL BE INTERESTING TO SEE IF THEY ATTRACT ANY INTEREST.

HAVE YOU VISITED MATTHEW READ'S WEBSITE RECENTLY? I TRY AND KEEP A CLOSE EYE ON IT IN CASE THERE IS ANY FURTHER NEWS OF THE ACULAB RUNNING ON HIS EMULATOR.
I NOTICED THAT HE'S POSTED THREE VERY INTERESTING INTERVIEWS WITH PEOPLE WHO WERE AT THE FORE-FRONT OF TANDY DEVELOPMENT IN THE 197Ø'S/8Ø'S. THEY ARE WITH WILLIAM DEMAS, DAN GOOKIN AND KEVIN TSCHUDI. REALLY FASCINATING STUFF AND WELL WORTH A CHECK-OUT.

ONE OF THE MOST INTERESTING EVENTS OVER THE LAST FEW MONTHS HAS

(CONTINUED ON PAGE 16)



AT THE
READY>
PROMPT



HERE'S A LITTLE BIT OF NONSENSE I CAME ACROSS THE OTHER DAY WHICH HAD COMPLETELY SLIPPED MY MIND OVER THE YEAR. GOOD-OLD SCHOOL BOY STUFF!!
A COUPLE OF QUICK WAYS TO MAKE A NUMBER EXACTLY DIVISIBLE BY NINE.

1) ADD UP THE INDIVIDUAL NUMBERS THAT MAKE UP ANY NUMBER AND SUBTRACT THEM. E.G. 5843: THE INDIVIDUAL NUMBER 5,8,4 AND 3 ADD UP TO 20. SUBTRACT 20 FROM 5843 MAKING 5823. 5823 IS EXACTLY DIVISIBLE BY 9!
2) SUBTRACT FROM THE ORIGINAL NUMBER IT'S OWN NUMBERS IN REVERSE. THE ANSWER WILL BE EXACTLY DIVISIBLE BY 9.

---== o0o ==---

TRS80 HARDWARE (MODEL 1)

A/D CONVERTER.

4 Analogue inputs. 2 Flat inputs. Analogue output plugs into keyboard. Software included. Board only: £48.00. Complete in case: £60.00. P&P £1.50.

JOYSTICK CONTROLLER

Uses Atari Joystick, plugs into keyboard. Software included. £24.00. P&P £1.00

I/O PORT.

6 solid state and 2 Relay contact outputs. 8 T.T.L. Inputs. LED indication of outputs. £72.00. P&P £2.00

LIGHT PEN.

Plugs into Cassette socket. BUILT IN AMPLIFIER. Software included. £22.50. P&P £1.00.

All prices include VAT.

T. GARLAND & SON LTD.

14A Kenworthy Lane, Northendon, Manchester M22 4EJ
Phone: 061 998 4207.

IN MAV'S WORKSHOP

BY

IAN MAVRIC

TONE-LOC'S MODEL 1'S

ABOUT A MONTH AGO, I WAS HAVING A YARN OVER A BEER WITH JOHN BENSON ABOUT THE STATE OF TRS-80 AFFAIRS IN THE WORLD IN GENERAL AND AUSTRALIA IN PARTICULAR, HAVING JUST WITNESSED A MODEL 1 16K LEVEL II KEYBOARD UNIT (I.E. THE COMPUTER) SELL FOR $A461.20 ON EVIL-BAY IN MELBOURNE[1]. WHAT WAS MORE INCREDIBLE WAS THAT THE MACHINE DIDN'T RUN (THE DISPLAY PHOTO SHOWED A SCREEN FULL OF @ SYMBOLS) AND IT DIDN'T EVEN COME WITH THE MONITOR, WHICH WAS IN A 2ND AUCTION AND SOLD FOR $A181.38[2] BOTH CAME WITH DUST COVERS AND COSMETICALLY WERE ABOVE AVERAGE CONDITION BUT WE REMAINED BEWILDERED BY THE SHEER AMOUNT SOMEONE WAS WILLING TO SHELL-OUT FOR THIS NON-RUNNING M1. WHAT DOES THAT MAKE RUNNING M1S WORTH? MY JOKE WAS THAT IT WOULDN'T BE TOO LONG BEFORE I WAS ABLE TO ASK THE SAME PRICES TANDY WAS ASKING IN 1978 FOR THE MACHINES AT THIS RATE, SINCE I SELL MY EQUIPMENT FULLY TESTED AND GUARANTEED FOR 90S DAYS, JUST LIKE WHEN IT WAS NEW. BENSON'S RETORT WAS THAT THE BUYER WOULD PROBABLY FIND ME TO REPAIR THE COMPUTER, SINCE I'M THE ONLY ONE LEFT IN AUSTRALIA WHO CAN REPAIR A MODEL 1. I THOUGHT LITTLE OF IT UNTIL I RECEIVED AN EMAIL FROM THE OWNER, A FEW WEEKS LATER, ASKING IF I'D BE ABLE TO HAVE A LOOK AT IT AND FIX IT. I ENTHUSIASTICALLY AGREED, SINCE I'D LIKE TO SEE THE MACHINE AND MEET THE NEW GENERATION OF VINTAGE

COMPUTER COLLECTOR WHO ARE PREPARED TO DIG DEEP TO BUY THESE MACHINES.

SYSTEM 1: FUNKY COLD MEDINA

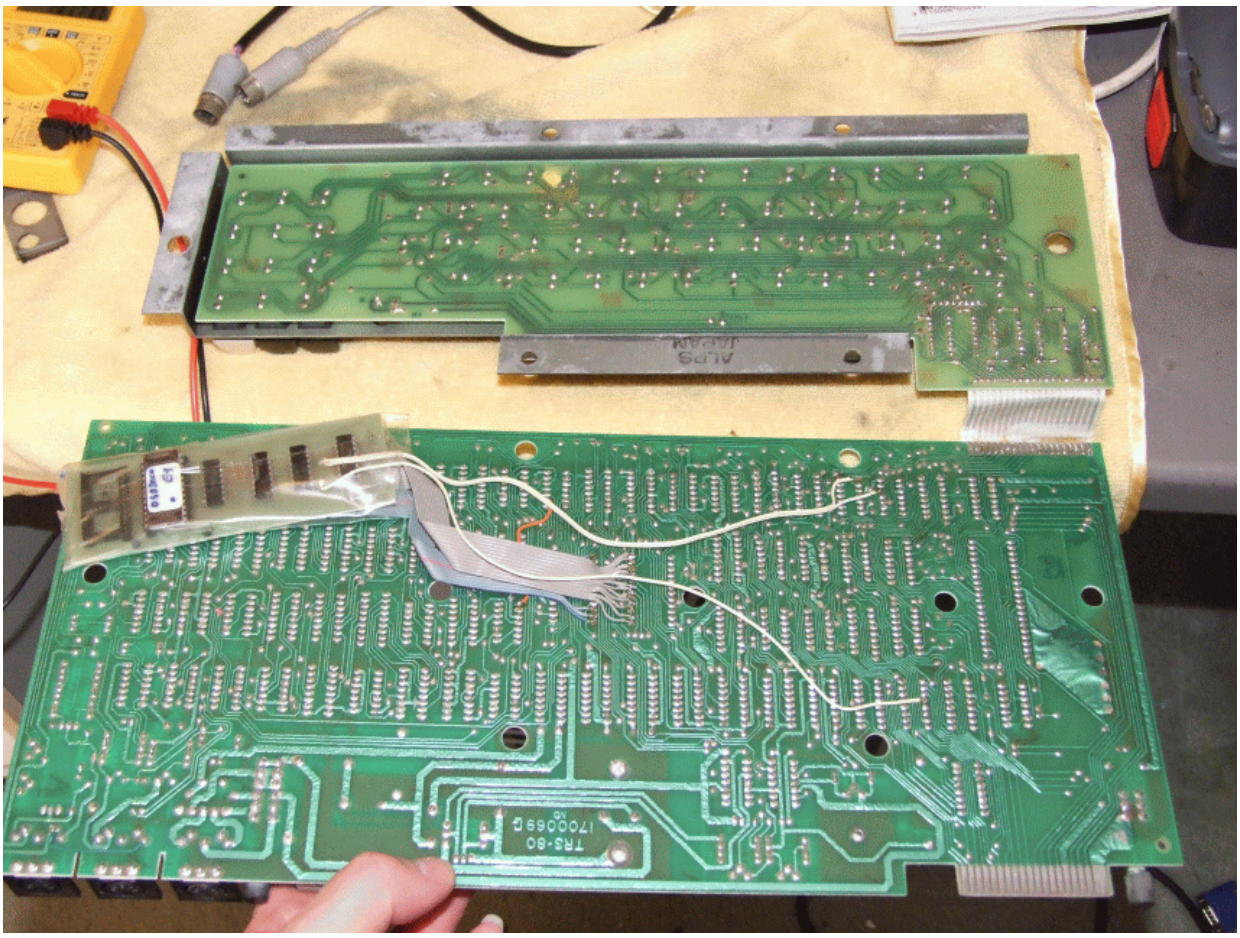
ABOUT A WEEK LATER A MIDDLE-AGED ASIAN GUY SHOWED UP WITH THE MODEL 1 UNDER HIS ARM. I ASKED ABOUT HIS INTEREST IN THESE MACHINES AND HE DESCRIBED HOW THERE WERE WAS A TOP-10 COLLECTABLE VINTAGE HOME COMPUTERS LIST SOMEWHERE ON THE NET AND THE MODEL 1 WAS ON IT. HE WAS ALSO AFTER AN APPLE II AS WELL, AND A COMMODORE PET 2001 AND A SINCLAIR ZX-80. I SAID I COULDN'T HELP WITH THOSE BUT I COULD GET THE MODEL 1 UP AND RUNNING AGAIN, SO HE LEFT IT WITH ME AND I HAD THE MACHINE ON MY TEST BENCH IN A FEW MINUTES. IT DISPLAYED A 32-COLUMN @@@ SCREEN WITH NO REFRESH STROBES, BUT THE PICTURE WAS SHARP AND STABLE, SO AT LEAST I KNEW THE VIDEO SIDE WAS OK. POKING AROUND WITH THE LOGIC PROBE I FOUND THE BUS WAS FALLING OVER AT EITHER Z35 OR Z64 (BOTH 74LS157 DEVICES) SO I REPLACED THEM AND WAS GREETED WITH A WELCOME @9@9 SCREEN WITH STROBE PULSES[3], SAME AS YOU GET WITH THE ROMS REMOVED... ONLY THING WAS THEY WERE STILL IN PLACE. WHAT WOULD STOP ROMS FROM STARTING UP? I REASONED THAT IF THEY WEREN'T CHIP-SELECTED BY THE MEMORY BUS THEN THEY WOULD APPEAR ABSENT FROM THE SYSTEM, AND Z74 (A 74LS00 DEVICE) DOES THAT, SO I CHANGED THAT DEVICE AND VOILA, THE NEXT POWER UP I WAS GREETED WITH THE FRIENDLY MEM SIZE? PROMPT[4]. A NEW KEYBOARD CABLE AND WE WERE BACK IN BUSINESS[5]. THE POWER SWITCH WOULDN'T SWITCH OFF (OR ON, FOR THAT MATTER - IT WAS STUCK IN THE ON POSITION) SO I ASKED OLD-MATE JOHN IF HE HAD ANY NOS SPARES AND SENT ONE STRAIGHT DOWN. ANOTHER 1/2 HOUR LATER AND THE POWER SWITCH WAS REPLACED - THEY ARE A P.I.T.A. TO REMOVE WITH A SOLDERING IRON AND SOLDER-SUCKER - AND THE COMPUTER WAS RUNNING AGAIN LIKE NEW.

I CALLED ITS OWNER AND TOLD HIM IT WAS BACK UP AND RUNNING, AND MADE A TIME TO PICK IT UP. HE ALSO TOLD ME THAT HE HAD ANOTHER MODEL 1 THERE WHICH HE DIDN'T BRING AS HE THOUGHT IT WOULD BE HARDER TO FIX. ALWAYS UP FOR A CHALLENGE I SAID BRING IT ON OVER - HOW HARD COULD IT BE?

SYSTEM 2: WILD THANG

THIS ONE HAD A SCREEN FULL OF RANDOM CHARACTERS BUT SEEMED TO RESPOND TO KEYBOARD INPUT. I ASKED IF IT WAS LIKE IT HAD SPELLING ERRORS? HE WASN'T SURE WHAT I MEANT BUT I RECKON IT WAS A PRETTY COMMON FAULT MODEL 1S DEVELOP WHERE ONE OR MORE OF THE VIDEO SRAMS (A 2102 DEVICE) FAIL (A BIT GETS STUCK) SO THE INFORMATION BEING FED TO Z29, THE CHARACTER GENERATOR, IS FED THE WRONG INFORMATION AND SO YOU SEE SPELLING ERRORS. MEM SIZE? CAN BECOMES <E< 5!2E? BUT OTHERWISE THE COMPUTER RUNS PROPERLY[6].

HE BROUGHT IT IN AND IT WAS EXACTLY THE PROBLEM I THOUGHT IT WAS, SO POKING AROUND THE VIDEO MEMORY WITH THE LOGIC PROBE I COULDN'T DECIDE WHETHER IT





WAS Z46 OR Z62 THAT WAS CAUSING GRIEF SO I CHANGED Z46 FIRST AND WAS IMMEDIATELY GREETED UPON POWER UP BY MUCH LESS GARBAGE ON THE SCREEN BUT STILL SPELLING ERRORS. MORE PROBING AND I STILL WASN'T HAPPY WITH Z62 SO I REPLACED THAT AS WELL AND PRESTO MEM SIZE? AND A HAPPY MODEL 1 AGAIN. TWO KEY SWITCHES HAD STOPPED WORKING AND THE OBLIGATORY KEYBOARD CABLE NEEDED CHANGING, AND THE COMPUTER WAS FIXED.

IT HIGHLIGHTS SOMETHING ABOUT HOW OLD THESE COMPUTERS ARE GETTING - BACK IN THE 80S AND 90S WHEN I REPAIRED MODEL 1S I'D NEVER SEEN MORE THAN ONE OF THESE 2102S FAIL, BUT NOW I'M STARTING TO SEE MODEL 1S WITH TWO OR MORE FAILED 2102S.

THE HAPPY OWNER CAME BY AND PICKED UP HIS MODEL 1. NO IDEA WHAT THEY ARE DOING AT THE MOMENT BUT I WOULD IMAGINE SITTING AROUND LOOKING RETRO AS PART OF A COMPUTER COLLECTION AND OCCASIONALLY RUNNING BLACKJACK OR GALAXY INVASION PLUS.

OH, AND WHY DID I SAY THESE WERE TONE-LŌC'S MODEL 1S? THAT'S THE EBAY NAME OF THE OWNER OF THESE MACHINES, OBVIOUSLY BORROWED FROM THE 1980S AMERICAN RAP ARTIST. DID THE ASIAN VINTAGE COMPUTER ENTHUSIAST HAVE ANYTHING IN COMMON WITH TONE-LŌC? NOT THAT I COULD SEE. HE HAD GOOD TASTE IN COMPUTERS THOUGH.

UNTIL NEXT TIME,
MAV

REFERENCES:

1. EBAY ITEM # 120721832836

2. EBAY ITEM # 120721835018

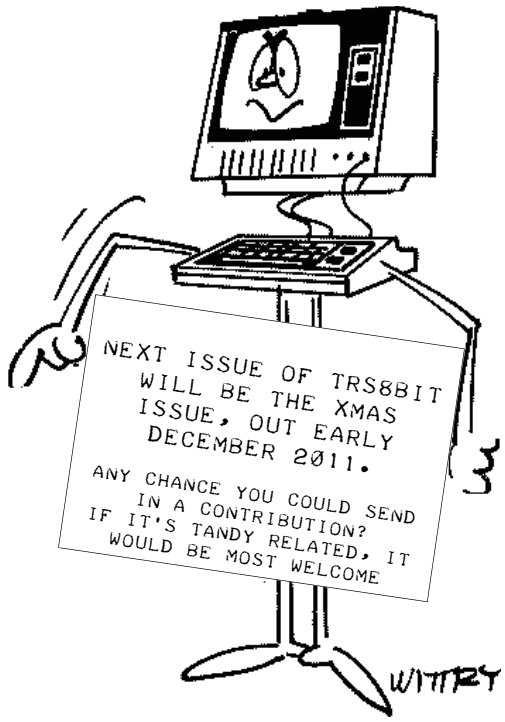
3. TRS-80 MICRO COMPUTER TECHNICAL REFERENCE HANDBOOK (RS CAT. NO. 269-2103) PAGE 26 "RAM ADDRESSING"

4. TRS-80 MICRO COMPUTER TECHNICAL REFERENCE HANDBOOK (RS CAT. NO. 269-2103) PAGES 21-22 "ROM DECODING"

5. THE CUSTOM TRS-80 & OTHER MYSTERIES BY D.B. KITSZ, PAGES 237-238 "A NEW KEYBOARD CABLE"

6. TRS-80 MICRO COMPUTER TECHNICAL REFERENCE HANDBOOK (RS CAT. NO. 269-2103) PAGE 71 "SPELLING ERRORS"

---=== o0o ===---



ACCEL 3

UPGRADE IMPROVEMENTS

IMAGINATIVELY NAMED ACCEL3. THE NEW PROGRAM IS APPARENTLY A COMPLETE RE-WRITE OF ACCEL2 AND INCORPORATES SOME NEW FEATURES.

ACCEL3 WILL COMPILE NON STRUCTURED FOR. . NEXT LOOPS MAKING IT POSSIBLE TO COMPILE PROGRAMS CONTAINING JUMPS OUT OF LOOPS, CONDITIONAL NEXT STATEMENTS AND SO FORTH. ACCEL2 DIDN'T DO THIS OFTEN LEADING TO CHANGES BEING MADE TO A PROGRAM BEFORE IT COULD BE COMPILED. THE SNAG IS THAT THE EXTRA CODE TO HANDLE UNSTRUCTURED LOOPS SLOWS UP COMPILED PROGRAMS - FOR. NEXT STATEMENTS USED WITH THE NEW COMPILER ARE ABOUT HALF AS FAST AS THEY WERE UNDER ACCEL2. SIMILARLY ACCEL3 NOW COMPILES REFERENCES TO ARRAYS WITH MORE THAN ONE DIMENSION BUT THE SPEED OF ACCESS TO ONE-DIMENSIONAL ARRAYS HAS SUFFERED.



ACCEL3 COMPILES SOME FUNCTIONS THAT ACCEL2 USED TO LEAVE FOR THE BASIC INTERPRETER TO HANDLE. IN PARTICULAR, FLOATING-POINT FOR. NEXT LOOPS AND FUNCTIONS SUCH AS INT AND SQR ARE NOW COMPILED INTO ROM CALLS. THE COMPILATION OF THE FUNCTIONS (E.G. SIN, ETC) DOESN'T REALLY SPEED THEM UP SINCE THEY TAKE MUCH LONGER TO PROCESS THAN TO INTERPRET BUT IT DOES MEAN THAT EXPRESSIONS USING THEM CAN BE COMPILED. THIS WOULD SPEED UP THE MULTIPLICATION IN X=SIN(X)*3.1416, FOR EXAMPLE. THE USR(N) FUNCTION, USED TO CALL A MACHINE-CODE ROUTINE, IS NO LONGER COMPILED. ACCEL3 WILL ALSO COMPILE PROGRAMS WHICH USE VARIABLE BOUND ARRAYS, SUCH AS - 2Ø INPUT N DIM A$(N,2). THE ACCEL3 SALES LITERATURE CLAIMS IT IS FASTER AND GENERATES MORE COMPACT CODE THAN ACCEL2, BUT THE DIFFERENCE IN PERFORMANCE DOES NOT SEEM TO BE THAT GREAT; THE NEW COMPILER NO LONGER HAS THE DISC COMMANDS / SAVE, /RUN AND / LOAD. ACCEL3 ALLOWS COMPILED PROGRAMS TO BE SAVED, RUN AND LOADED JUST AS IF THEY WERE NORMAL BASIC ALTHOUGH THEY WILL NOT WORK UNLESS THE RUN-TIME ROUTINES OF ACCEL3 ARE IN MEMORY. EVEN THE CASSETTE COMMANDS, CSAVE AND CLOAD, CAN NOW BE USED TO STORE AND RETRIEVE COMPILED PROGRAMS.

THE PRE-RELEASE VERSION OF THE ACCEL3 COMPILER HAS BEEN TESTED USING IT TO SPEED UP A FEW WELL KNOWN PROGRAMS - IT EVEN FOUND ONE OR TWO UNNOTICED SYNTAX ERRORS! MORE THAN HALF OF THE PROGRAMS COMPILED FIRST TIME AND MOST OF THE REST COULD BE COMPILED ONCE A FEW LINES WERE SHORTENED OR EXPRESSIONS SIMPLIFIED.

LINE EDITING

HERE'S A LITTLE PROGRAM FOR THE LEVEL 2 16K FANS, WHO I REGRET, HAVE BEEN SOMEWHAT NEGLECTED OVER THE LAST COUPLE OF ISSUES.

THERE'S QUITE A BIT OF HISTORY WITH THIS LITTLE PROGRAM. YES, IT WOULD CLASS AS A ONE-LINER WITH VERY LITTLE BOTHER, BUT I'VE LEFT IT AS 4 LINES, TO MAKE IT EASIER TO FOLLOW.

INITIALLY, IT STARTED OUT AS A PROGRAM CALLED 'AUTO EDIT' WRITTEN BY DAN ROLLINS, WHICH APPEARED IN 80MICRO. IT WAS THEN PATCHED BY JOHN COUCH FOR NATGUG NEWS AND THEN SIMPLIFIED, AS BELOW.

THE PROGRAM ALLOWS THE AUTOMATIC EDITING OF SUCCESSIVE PROGRAM LINES WITHIN BASIC. BY PRESSING THE 'SHIFT<ENTER>' KEYS THE COMPUTER GOES INTO EDIT MODE FOR THE NEXT LINE.

THE 23 BYTES OF MACHINE CODE ARE RELOCATABLE AND IN THE VERSION BELOW, FOR A TAPE-BASED MODEL 1, ARE LOADED INTO PART OF RAM RESERVED FOR DOS. IT ACTUALLY LINKS THE MACHINE CODE WITH THE KEYBOARD INPUT ROUTINE (IN ROM), VIA A DOS EXIT AT 16815 DECIMAL. THIS PROTECTS THE CODE AND AVOIDS IT BEING OVERWRITTEN BY BASIC.

ONCE THE PROGRAM HAS BEEN RUN, IT CAN BE DELETED, BUT IN THE EVENT OF A SYSTEM CRASH (I.E. MEMORY SIZE?) THE POKE STATEMENTS IN LINE 70 WILL HAVE TO BE RE-ENTERED.

BY REPEATED USE OF THE 'SHIFT<ENTER>' KEYS THE PROGRAM CAN ALSO BE USED AS A SINGLE-STEP LIST COMMAND, WHICH CAN BE VERY USEFUL AT TIMES!

I THINK THIS PROGRAM WOULD BE WORTH CONVERTING TO RUN IN DISK BASIC. DO YOU FANCY HAVING A GO? IF ANYONE REQUESTS, I'LL HAVE A GO MYSELF FOR THE NEXT ISSUE.

--== oØo ==--

TRS80 HARDWARE (MODEL 1)

A/D CONVERTER.

4 Analogue inputs. 2 Flat inputs. Analogue output plugs into keyboard. Software included.
Board only: £48.00. Complete in case: £60.00. P&P £1.50.

JOYSTICK CONTROLLER

Uses Atari Joystick, plugs into keyboard. Software included. £24.00. P&P £1.00

I/O PORT.

6 solid state and 2 Relay contact outputs. 8 T.T.L. Inputs. LED indication of outputs. £72.00. P&P £2.00

LIGHT PEN.

Plugs into Cassette socket. BUILT IN AMPLIFIER. Software included. £22.50. P&P £1.00.

All prices include VAT.

T. GARLAND & SON LTD.

14A Kenworthy Lane,
Northendon, Manchester M22 4EJ
Phone: 061 998 4207.


```
10 DATA 58,128,56,31,208,237,91,236,64,19,205,44
20 DATA 27,197,225,35,35,205,63,27,195,102,46
50 FOR X=16477 TO 16499:READA:POKEX,A:NEXT
70 POKE16815,195:POKE16816,93:POKE16817,64
```

WORKING WITH TRS80'S

BY PETER STONE

I SUPPOSE THAT ANYONE WHO BOUGHT A COMPUTER IN THE LATE 70S, EARLY 80S, INVARIABLY GOT ASKED THE QUESTION, "WHAT CAN YOU DO WITH IT?" ATTEMPTING TO EXPLAIN, POSSIBLY WITH DEMONSTRATIONS OF PROGRAMS, USUALLY LEFT THE QUESTIONER WALKING AWAY SHAKING THEIR HEAD OVER WHAT THEY THOUGHT WAS AN EXPENSIVE WASTE OF MONEY.

I WAS NO EXCEPTION TO THIS PHENOMENA. MY FIRST EXPOSURE CAME WHEN I ATTEMPTED TO SHOW SOME WORK COLLEAGUES A TRS80 MODEL 1. A DEMONSTRATION OF A GRAPHICS PROGRAM, FOLLOWED BY A GAMES PROGRAM INVOKED THE RESPONSE, "SO?" I SEEMED TO BE THE ONLY ONE WHO COULD SEE THE WONDERFUL THINGS THIS MACHINE COULD DO.

IN TRUTH, I WASN'T UNSURPRISED AT THE RESPONSE. AS I GOT INTO MY EARLY TEENS, I GOT INTERESTED IN ELECTRONICS, & DECIDED ON THIS AS A CAREER. DURING MY TIME AT MY FIRST COMPANY, I BECAME THE RECIPIENT OF THE MONTHLY INDUSTRIAL ELECTRONICS MAGAZINES THAT ARRIVED BECAUSE OF THE COMPANIES BRIEF, BUT ABORTIVE, FORAY INTO THE WORLD OF ELECTRONIC DIMMERS. THESE MAGAZINES HAD ARTICLES IN THEM ABOUT MICROPROCESSORS, & HOW THE SAME HARDWARE WITH DIFFERENT SOFTWARE COULD DO A WIDE RANGE OF TASKS, THIS LED TO ME TAKING COURSES IN DIGITAL TECHNIQUES & LOGIC DESIGN, SO SHAPING MY FUTURE CAREER, THOUGH JUST HOW MUCH I DIDN'T KNOW. WHILE I WAS AT THIS COMPANY, I TRIED TO EXPLAIN THE WONDERS OF WHAT MICROPROCESSORS COULD DO, BUT GOT LAUGHED AT. I WAS ALSO PLAYING WITH ELECTRONIC CIRCUITS, & THESE WOULD INVARIABLY EITHER NOT WORK FIRST TIME, OR NOT WORK THE WAY I'D PLANNED THEM, LEADING TO MORE LAUGHTER AT MY EXPENSE.

I WAS 28 WHEN PERSONAL COMPUTERS BECAME AVAILABLE TO THE GENERAL PUBLIC, & THE MODEL 1 BECAME MY PERSONAL CHOICE OF COMPUTER SIMPLY BECAUSE I'D BEEN READING THE FIRST EDITIONS OF PERSONAL COMPUTER WORLD, (PCW), & PRACTICAL COMPUTING, SO I'D READ THE ADVERTS FOR THE COMMODORE PET, THE APPLE II & THE TRS80 MODEL 1. I'D SEEN BOTH AN APPLE & A COMMODORE IN LOCAL SHOPS, WITH LARGE "DO NOT TOUCH" SIGNS ON THEM, & NO-ONE AROUND WHO APPEARED INTERESTED IN THE MACHINES OR SHOWING ME WHAT THEY COULD DO. HOWEVER I WALKED INTO MY LOCAL TANDY FRANCHISE STORE, & STOOD WATCHING A DEMONSTRATION ON THE SCREEN. THE SYSTEM ON DISPLAY CONSISTED OF KEYBOARD, MONITOR, EXPANSION INTERFACE, DISK DRIVE & LINE PRINTER. THE STORE MANAGER CAME OVER & STARTED TALKING. I EXPLAINED THAT I'D BEEN READING ABOUT THE MACHINE & ITS POSSIBILITIES & WAS INTERESTED IN IT. I ALSO EXPLAINED THAT I COULDN'T AFFORD ONE AT THE TIME & WOULD NEED TO SAVE THE DEPOSIT FOR FINANCE. THE REASON FOR THIS, WHICH I DIDN'T EXPLAIN, WAS THAT I

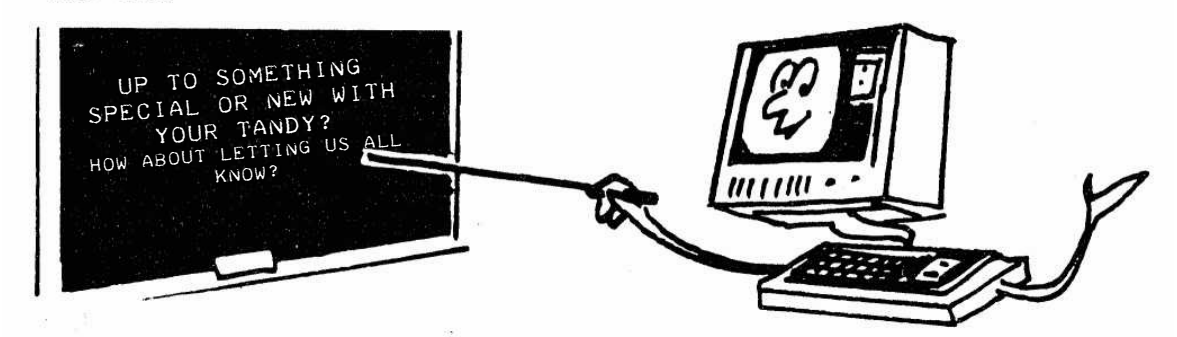
USED TO BUY THE AMERICAN MAGAZINE POPULAR ELECTRONICS, & TRS80 ADVERTS GAVE A PRICE OF $807. HOWEVER, THE UK PRICE WAS £807, ALMOST DOUBLE!

THIS DIDN'T STOP HIM ENCOURAGING ME TO SIT DOWN & HAVE A PLAY ON THE MACHINE. I SAT AT THE KEYBOARD & WONDERED WHAT TO DO. I'D GOT THE LATEST EDITION OF PCW WITH ME, & I REALISED THERE WAS A PROGRAM CALLED SUBCHASE FOR THE MODEL 1 IN IT, WRITTEN BY AJ HARDING, SO I SAT DOWN & ENTERED IT. I MANAGED TO GET IT ALL ENTERED INTO THE SYSTEM, & SAVED TO FLOPPY DISK. I RAN THE PROGRAM & WATCHED THE DISPLAY GO CRAZY. THE PROGRAM WAS FOR LEVEL 1 BASIC & OF COURSE I WAS USING LEVEL 2. THIS MEANT THAT THE PROGRAM USED STATEMENTS LIKE PRINT AT INSTEAD OF PRINT@. THERE WAS A GUY IN THE STORE WHO OFFERED ME A DEBUGGED VERSION OF THE PROGRAM, (HE'D BOUGHT HIS OWN MACHINE), BUT I DECLINED, I WANTED TO GET THE PROGRAM WORKING BY MYSELF BECAUSE I'D SPENT ALL THAT TIME ENTERING IT, PLUS, I WAS TRULY FASCINATED BY THE MACHINE & WHAT IT COULD DO & WANTED TO LEARN HOW TO USE IT, ALSO I FELT THAT BY BEING DETERMINED TO SORT THINGS OUT FOR MYSELF, I'D GOT THE MANAGERS APPROVAL. THAT NIGHT WHEN THE STORE CLOSED, I LEFT WITH A COPY OF THE LEVEL 2 MANUAL, THE TRSDOS 2.3 MANUAL & A LISTING OF MY PROGRAM. THROUGHOUT THE FOLLOWING WEEK, I PRACTICALLY WORE THE MANUALS OUT & WENT BACK TO THE STORE THE FOLLOWING SATURDAY WITH A HEAVILY MODIFIED LISTING & MORE KNOWLEDGE. I'LL DRAW A VEIL OVER THE FOLLOWING MONTHS, BUT I EVENTUALLY DID GET THE PROGRAM WORKING. I KNEW THAT ME USING THE MACHINE WAS TO THE ADVANTAGE OF THE STORE MANAGER, AS HE KNEW THAT SOMEONE ACTUALLY USING THE MACHINE DREW INTERESTED PEOPLE IN.

I SEARCHED AROUND FOR ANOTHER PROGRAMMING PROJECT, AS I WAS NOW TOTALLY BITTEN BY THE COMPUTER BUG. I'D GOT THE LATEST COPY OF POPULAR ELECTRONICS WITH ME, & IT HAD A PROGRAM THAT DREW A STAR ON THE SCREEN, & I STARTED PROGRAMMING. ONE OF MY FONDEST MEMORIES COMES FROM THIS TIME. THE SHOP WAS ON THE CORNER OF A MAIN ROAD & A SIDE STREET, SO THE MACHINE WAS MOVED INTO THE SIDE WINDOW, FOR BETTER DISPLAY. LATE ONE AFTERNOON WHILE I WAS WORKING ON A MODIFICATION TO THE ORIGINAL PROGRAM, I GOT AN ITCH ON THE BACK OF MY NECK, I USUALLY GET THIS WHEN I'M BEING WATCHED. I TURNED MY HEAD, & WAS AMAZED TO SEE AROUND TWENTY FIVE ENTRANCED MEN & BOYS, (NO FEMALES), WATCHING ME PROGRAM, IT WAS BOTH A SHOCK & AN ADRENALIN RUSH. AFTER A YEAR, THE PRICE OF A LEVEL 2 SYSTEM CAME DOWN TO £570(ISH), WHICH MEANT THAT THE MONEY I'D GOT SAVED WAS ENOUGH OF A DEPOSIT TO GET A MACHINE ON FINANCE, WHICH I DID, SO STARTING MY LOVE-IN WITH COMPUTERS.

PLEASE DON'T THINK THAT I NOW IMMEDIATELY STARTED WORKING AT THE LOCAL POLYTECHNIC. I'D GOT A JOB AS AN ELECTRONICS SYSTEMS

TEST ENGINEER AT A COMPANY THAT BUILT SPECIALISED SCALES & WEIGHING EQUIPMENT. HOWEVER I STILL WENT INTO MY LOCAL TANDY FRANCHISE STORE, AS IT STOCKED BOOKS & PROGRAMS OTHER THAN THE OFFICIAL TANDY ONES. ONE DAY I WENT TO THE STORE & NOTICED THERE WAS AN OLDER GUY USING THE MODEL 1. TRUTHFULLY WITH HIS SHABBY MAC & UNKEMPT HAIR, HE LOOKED THE SORT OF PERSON WHO YOU WOULD CROSS THE ROAD TO AVOID IF YOU MET HIM IN THE STREET. HOWEVER, I FOUND OUT THAT HE'D GOT HIS OWN MODEL1, & WAS EXTREMELY INTELLIGENT. LIKE ME HE WOULD COME INTO THE STORE OF A SATURDAY AFTERNOON, & SAW A JOB ADVERTISED IN THE LOCAL PAPER FOR A MICRO-COMPUTER TECHNICIAN AT THE LOCAL POLYTECHNIC, SO I DECIDED TO APPLY. I HAPPENED TO MENTION THIS IN PASSING THE FOLLOWING SATURDAY, & THOUGHT NO MORE ABOUT IT. A FEW WEEKS LATER I GOT A LETTER INVITING ME TO AN INTERVIEW & DULY ATTENDED. THE INTERVIEW FOLLOWED THE USUAL PATTERN. WE WERE SHOWN AROUND FIRST, THEN THE INTERVIEW. AS WE WERE BEING SHOWN THE MICRO-COMPUTING LABS, IMAGINE MY SURPRISE WHEN WE WERE INTRODUCED TO THE OLDER GUY WHO CAME INTO THE TANDY SHOP ON SATURDAY. APPARENTLY HE WAS THE SENIOR TECHNICIAN OF



AFTER WE WERE JOINED BY A THIRD YOUTH, STARTED TO USE THE PLACE AS A SORT OF CLUB HOUSE. THE MANAGER DIDN'T MIND AS IT GAVE HIM A VERY KNOWLEDGEABLE GROUP, TO HELP SELL MACHINES. AFTER TWO YEARS, THE WEIGHING EQUIPMENT COMPANY STARTED CHANGING CONTRACTS & WORKING CONDITIONS FOR THOSE OF US WORKING IN THE ELECTRONIC SYSTEMS DIVISION, IN A DICTATORIAL & BULLYING WAY. AFTER BEING TOLD THAT I COULD EITHER ACCEPT THE NEW CONTRACT OR BE SACKED, I DECIDED THAT I WOULD ACCEPT THE CONTRACT, BUT LOOK FOR ANOTHER JOB.

A COUPLE OF MONTHS LATER, I THE DEPARTMENT. I WAS THE LAST TO LEAVE THE OFFICE, & AS I DID SO HE GAVE ME A THUMBS UP, NEEDLESS TO SAY, I GOT THE JOB.

UPON STARTING AT THE POLY, I GOT GIVEN THE TOOLS OF THE TRADE, WHICH WAS BASICALLY A COUPLE OF 'MASTER' FLOPPY DISKS, SOME BLANKS & SOMETHING TO STORE THEM IN. MY FIRST TASK BEING TO GENERATE SOME WORKING MASTERS. THE FOLLOWING MONTH WAS DEVOTED TO GETTING TO KNOW THE SYSTEMS & THE LAYOUT OF THE POLY. MY OWN SYSTEM AT THIS TIME WAS STILL CASSETTE BASED, & IT WAS OBVIOUS THAT IT NEEDED UPGRADING, IF ONLY I'D KNOWN WHERE THIS WOULD LEAD!

I SEEM TO RECALL THAT I HAD TO DO A SIX MONTH PROBATION PERIOD, BEFORE BEING TAKEN ON PERMANENTLY, NEEDLESS TO SAY I COMPLETED THE PROBATION PERIOD SUCCESSFULLY. WITHIN A MONTH, THE SENIOR TECHNICIAN STARTED SAYING I NEEDED TO GET MYSELF AN EXPANSION INTERFACE & A PAIR OF FLOPPY DISK DRIVES IF, THAT IS, I WANTED TO BE A DRIVING FORCE IN THE DEPARTMENT. I THOUGHT ABOUT IT, & AFTER LOOKING IN MAGAZINES, I FOUND A PAIR OF TWIN FLOPPY DRIVES MANUFACTURED BY CUMANA FOR AROUND £100. THE NEXT BREAK CAME WHEN TANDY STARTED SELLING MODEL 1 EXPANSION INTERFACES WITH NO MEMORY FOR £125. A BANK LOAN ALLOWED ME TO PURCHASE THE EQUIPMENT, & SO I WAS EQUIPPED, SO I THOUGHT, TO MAKE A FORTUNE. THE MEMORY PROBLEM WAS SOLVED BY PURCHASING SIXTEEN 4116 MEMORY CHIPS FROM A SUPPLIER IN AN ELECTRONICS MAG, ONCE THE 90 DAY WARRANTY HAD EXPIRED. NOT THAT I WAS WORRIED IF THE UNIT DEVELOPED A FAULT, AS PART OF MY JOB WAS FIXING THE EQUIPMENT WHEN IT BROKE.

THE MAIN LAB CONSISTED OF 16 TRS80 MODEL 1 COMPUTERS. MY FIRST TASK, AS FAR AS I CAN REMEMBER, WAS UPGRADING THE EXPANSION INTERFACES FROM 16K TO 32K, GIVING A FULL 48K FOR PROGRAMMING. EACH OF THE COMPUTERS HAD A BOX ATTACHED WHICH PROVIDED A NUMBER OF DIFFERENT INTERFACES. THERE WERE 8 SWITCHES & LEDS TO PROVIDE DIGITAL I/O, A SLIDER POTENTIOMETER, A TRANSISTOR SETUP AS A TEMPERATURE MEASURING DEVICE, OR A PAIR OF INPUT PINS EITHER OF WHICH COULD BE COUPLED TO AN A/D CONVERTER TO ALLOW FOR MEASURING VOLTAGES, & A D/A CONVERTER CONNECTED TO EITHER A MOTOR, A METER OR A PAIR OF TERMINALS TO PROVIDE AN ANALOGUE VOLTAGE. AT THE BOTTOM OF THE UNIT WAS A PLUG BOARD WHICH ALLOWED CIRCUITS TO BE CONSTRUCTED. THE VARIOUS CIRCUITS HAD BEEN MAPPED TO 4 PORTS WHICH COULD BE ACCESSED VIA THE OUT, PORT,VALUE & INP(PORT) BASIC STATEMENTS. ANOTHER OF MY REGULAR DUTIES WAS TO TEST THAT EACH OF THE BOXES WORKED. ONE OF THE LECTURERS HAD WRITTEN A BASIC PROGRAM TO HELP TEST THE BOXES. THE THING WAS THE PROGRAM WAS BUGGY, QUIRKY & GENERALLY UNRELIABLE! AFTER USING THE PROGRAM FOR THE FIRST TIME, I IMMEDIATELY SET ABOUT RE-WRITING IT. A NEW BASIC VERSION WAS READY BY THE NEXT DAY, & A MACHINE CODE VERSION WITHIN 10 DAYS. THE MAIN CHALLENGE WAS CODING A ROUTINE THAT WOULD TAKE THE VALUE FROM THE A/D CONVERTER & DISPLAY IT IN A FORMATTED VALUE ON THE SCREEN, WITH LEADING ZEROS REPLACED BY SPACES, BUT A ZERO BEING SHOWN WHEN IT WAS ACTUALLY PART OF THE DISPLAYED NUMBER.

THE 16 COMPUTERS IN THE MAIN LAB WEREN'T THE ONLY TRS80 MODEL 1S IN THE BLOCK WHERE I WORKED. THE CHEMISTRY DEPARTMENT HAD 10, WHICH WHEN CONNECTED TO VARIOUS PIECES OF EQUIPMENT, PROVIDED DATA LOGGING FOR THE EXPERIMENTS, PLUS THERE WERE OTHERS ABOUT THE DEPARTMENT.

OTHER TASKS CARRIED OUT DURING THOSE FIRST MONTHS INVOLVED FITTING LOWER CASE

MODS TO ALL THE COMPUTERS, REPLACING ALL THE SYSTEM DISKS USED, & WIRING THE MAIN LAB UP WITH WALL PLATES TO PROVIDE POWER, (+12, +5, -5 & -12 VOLTS), & PINS 2,3 & 7 OF THE RS232 LEADS ALLOWING FOR FLEXIBILITY IN CONNECTING EQUIPMENT UP. HOWEVER, ONE OF THE MOST IMPORTANT JOBS I DID, (AT MY SUGGESTION), WAS FITTING IDC CONNECTORS TO ALL THE EDGE CONNECTORS ON BOTH THE MAIN UNITS & THE EXPANSION INTERFACES, SOLVING A PROBLEM OF BOTH PROGRAMS & TEMPERS BEING LOST IF KIT GOT SLIGHTLY MOVED, GIVING A MASSIVE INCREASE IN SYSTEM RELIABILITY.

I SUPPOSE, GIVEN THE "SAMENESS" OF THE PCS WE USE TODAY, THOSE NOT IN THE KNOW MIGHT THINK THAT THE USES & ABILITIES FOR AN EIGHT BIT COMPUTER WOULD BE VERY LIMITED. HOWEVER THE TRUTH WAS VERY DIFFERENT. A FAVOURITE SAYING OF MINE, USUALLY IN REPLY TO SOMEONE SAYING, "YOU CAN'T DO THAT WITH A COMPUTER", WAS, "I WISH YOU'D TOLD ME THAT BEFORE I'D DONE IT!" THIS WAS USUALLY TRUE AS I WAS ENCOURAGED TO EXPERIMENT AS PART OF MY WORK. EXAMPLES OF THINGS THAT TRS8ØS WERE USED FOR WITHIN THE DEPARTMENT, WERE A COMPUTERISED BOOKING SYSTEM, A FILE & PRINT TRANSFER SYSTEM & NIGHT TIME ENTRY CONTROL SYSTEM. THESE USUALLY INVOLVED HARDWARE MODIFICATIONS TO THE BASIC SYSTEM, USUALLY CARRIED OUT THROUGH THE EXPANSION PORT. THE COMPUTERISED BOOKING SYSTEM USED A SINGLE FLOPPY BASED SYSTEM, & MAINTAINED THE BOOKINGS FOR TWO ROOMS OF 16 MACHINES, FOR TWO WEEKS. TO PREVENT ANY TAMPERING, THE SYSTEM WAS ON THE INSIDE OF A WINDOW, & COMMUNICATION WAS THROUGH A HEX KEYPAD, & CUSTOM MENUS, WHICH ALLOWS THE ENTRY OF THE STUDENTS ID, & A FOUR DIGIT PASSCODE TO PREVENT SOMEONE DELETING A BOOKING. THE PROGRAM WAS A MARVEL OF HOW TO PACK BOTH DATA & OPERATING SYSTEM ON THE DISK & HOW TO FIT EVERYTHING IN MEMORY. THE COMPUTERISED ENTRY SYSTEM CAME LATER, IN RESPONSE TO A NEED TO LET STUDENTS HAVE NIGHT ACCESS TO THE LAB BLOCK, & AN AVERSION TO TECHNICIANS HAVING "UNNECESSARY" OVERTIME, (5 NIGHTS A WEEK). THIS INVOLVED ANOTHER HEX KEYPAD ON THE MAIN DOOR, COUPLED TO A SOLENOID KEEP ON THE DOOR. THE PARALLEL DATA FEED TO/FROM THE HEX KEYPAD RAN SOME 15 METRES TO A TRS8Ø IN OUR OFFICE. PEOPLE WHO NEEDED ACCESS, HAD TO REGISTER A SIX DIGIT NUMBER WITH US, TO WHICH THE

TRS 80 & VG

Software to make life easier

DISK-R.P.M. Measure the rotational speed of your disk drives and reveal variations that can be a cause of unexplained errors. Shows current and average speeds. Operates under any D.O.S. 32-48k Mod. I or III or V.G. **£14.50** incl. p&p. + V.A.T.

FASTER Analyse your Basic programmes while they run, FASTER will show you simple changes that can reduce run times by up to 50%. Accounting systems, Financial models, Engineering, Scientific, Games all run faster. Large complicated programmes improve the most. 16-48k Mod. I or III or V.G. Tape or Disk **£17.50** incl. p&p. + V.A.T.

QUICK-Compress Small (276 bytes), fast (process 800 lines in less than 3 seconds) utility removes blanks and remarks from your Basic programmes. Produces smaller, faster programmes, and doesn't alter the original logic. 16-48k Mod. I or III or V.G. Tape or Disk **£8.90** incl. p&p. + V.A.T.

SPECIAL: FASTER + QUICK-Compress. **£22.00** incl. p&p. + V.A.T.

Specify Mod I or III or V.G. and Tape or Disk when ordering.

E.A. International, 8 High Street, Suite 201 Meldreth, Royston, Herts. (0763) 60189

Catalogue £1.00 (Includes vouchers value £5.00)

SYSTEM ADDED 2 RANDOM DIGITS, THIS WAS THEN STORED. THERE WAS A SERIAL LINK FROM THIS ACCESS SYSTEM, TO THE BOOKING SYSTEM, & THE WHOLE THING RAN IN THE FOLLOWING MANNER. A STUDENT WOULD COME ALONG AT NIGHT & ENTER HIS 8 DIGIT CODE INTO THE KEYPAD. THE ACCESS SYSTEM WOULD CHECK THAT THE ENTERED CODE WAS VALID, & FROM ITS DATABASE, WOULD GET THE STUDENT ID NUMBER. THEN IT WOULD CHECK THAT THE STUDENT HAD A VALID BOOKING, FROM MEMORY, BETWEEN 15 MINUTES BEFORE, TO 5 MINUTES AFTER, & IF VALID, ALLOWED ACCESS. THE POSSIBILITY OF THEFT OR DAMAGE, OF WHICH WE NEVER SUFFERED FROM I MIGHT ADD, WAS MINIMISED BY EXPLAINING TO THOSE WHO USED THE SYSTEMS, THAT IN THE EVENT OF ANYTHING HAPPENING, THE LIST OF USERS BOOKED ON THE SYSTEM WOULD BE HANDED TO THE RELEVANT AUTHORITIES FOR INVESTIGATION. HOWEVER, A CCTV CAMERA, PART OF A LECTURERS FAILED ROBOTIC TABLE TENNIS EXPERIMENT, COUPLED TO A LONG PLAY VIDEO RECORDER GAVE US A VISUAL BACKUP. THE FILE TRANSFER & PRINT SYSTEM BASICALLY ALLOWED FOR THE STUDENTS TO EITHER PRINT THEIR WORK OFF, OR AS A BACKUP TO THE WORK KEPT ON THEIR FLOPPY DISKS, DO A TRANSFER OF THEIR FILES, IN ASCII TO, OR FROM, THEIR ACCOUNTS ON THE PRIME MAINFRAMES.

BESIDES MAINTAINING THE COMPUTER EQUIPMENT, & BUILDING VARIOUS PIECES OF EXPERIMENTAL ELECTRONIC HARDWARE, DESIGNED BY EITHER ME, OR THE SENIOR TECHNICIAN, THIS COULD BE SOMETHING AS SIMPLE AS PLAIN CURIOSITY, OR SOMETHING MORE STRUCTURED FOR EITHER STUDENT PROJECTS OR FINAL YEAR DEGREE STUDENTS PROJECTS, THIS WAS A FAVOURITE ACTIVITY FOR ME. FINAL YEAR PROJECTS I REMEMBER, INCLUDED COMPUTER NETWORKING & COMPUTER CONTROLLED EMBROIDERY EQUIPMENT. THE NETWORKING ONE, INVOLVED CONNECTING THREE TRS80S TOGETHER USING THE SIMPLISTIC RS232 INTERFACE WE PROVIDED AT THE WALLS SOCKET. THE "SWITCH" WAS A SIMPLE PASSIVE SETUP USING DIODES. THE THREE TRS80S ACTED AS FILE SERVER, WORKSTATION & PRINT SERVER RESPECTIVELY, SO THAT BESIDES WRITING THE BASIC DRIVERS FOR THE COMPUTERS, COLLISION & CONTROL SOFTWARE HAD TO BE DEVELOPED, TO ROUTE THE DATA SUCCESSFULLY BETWEEN THE SYSTEMS. THE COMPUTER CONTROLLED EMBROIDERY EQUIPMENT PROJECT CAME ABOUT FROM A LOCAL BUSINESSMAN, HE MANUFACTURED EMBROIDERED T-SHIRTS, TABARDS & JACKETS FOR CLUBS & BUSINESSES. THE PROBLEM WAS THAT THE EMBROIDERY MACHINES WERE CONTROLLED BY PUNCHED PAPER TAPE, & WAS EXPENSIVE, I SEEM TO RECALL THAT A NEW, 10 CHARACTER NAME TAPE COST £400 TO PRODUCE, & A REPLACEMENT TAPE WAS NEARLY AS COSTLY.

THE FIRST THING TO DO WAS TO COPY TAPES, WHICH WAS DONE ON A TYPE33 TELETYPE - WE HAD A NUMBER KNOCKING AROUND, & THEY WERE ON THE "TO BE DISPOSED OF" LIST, EVENTUALLY, THEY ALL WENT TO GOOD HOMES, I USED ONE AS A PRINTER FOR A WHILE. THE NEXT THING WAS TO SEE

HOW THE TAPES ACTUALLY WORKED & DO A "PROOF OF CONCEPT". THIS INVOLVED TAKING A TAPE, & VIA A TELETYPE & A LONG SERIAL LEAD, & GETTING A FILE ONTO A TRS80 FOR ANALYSIS. THE ACTUAL TAPE FORMAT HAD BEEN HELPFULLY PRODUCED IN THE MANUAL FOR THE EMBROIDERY MACHINES, SO ANALYSING THE TAPE WAS RELATIVELY EASY. AS WE DIDN'T HAVE AN ACTUAL MACHINE TO TEST THINGS OUT ON, WE USED THE NEXT BEST THING A TANDY PLOTTER THAT USED 8.5" WIDE PAPER, POSSIBLY ONE OF THE BEST PIECES OF KIT TANDY EVER SOLD.

FROM THIS IT WAS A SHORT STEP TO ACTUALLY DESIGNING EMBLEMS ON A GRAPHICS TABLET, ATTACHED TO A TRS80 OF COURSE, SAVING THE RESULTANT FILE ON A FLOPPY & TESTING IT ON THE PLOTTER. THE FACT THAT THE BUSINESSMAN WENT OUT & BOUGHT TWO MODEL1S, A GRAPHICS TABLET & PLOTTER SHOWED HOW MUCH HE VALUED THINGS. FINAL YEAR PROJECTS THAT WERE SPUN OF FORM THIS INCLUDED ROBUST PROGRAMS TO DO THE DESIGN REQUIRED & PRODUCE THE OUTPUT ON THE PLOTTER. ONE FINAL PROJECT, DIFFERENT TO THE REST, INVOLVED TAKING THE TRS80 & VIA THE SERIAL PORT, DEVELOP THE CIRCUITRY TO DRIVE THE EMBROIDERY MACHINES DIRECTLY, ELIMINATING THE PAPER TAPES ENTIRELY.

AS FAR AS ORDINARY STUDENT ASSIGNMENTS WENT, THE FOLLOWING MAY SERVE AS AN EXAMPLE OF THE SORT OF THING THE STUDENTS HAD TO DO, IT CONSISTED OF THREE MODULES, & HAD TO BE COMPLETED OVER TWO TERMS, WITH EACH STAGE BEING DOCUMENTED & HAD TO INCLUDE A PROGRAM LISTING. THE FIRST PART OF THE ASSIGNMENT INVOLVED WRITING A PROGRAM, SPECIFICALLY A SUBROUTINE, EITHER IN MACHINE CODE OR BASIC, DEPENDING ON THE STUDENTS LEVEL OF COMPETENCE, THAT ACTED AS A SIMPLE DUMB TERMINAL & ALLOWED COMMUNICATION BETWEEN TWO COMPUTERS OVER THE RS232 LEAD. THE NEXT STAGE INVOLVED SOLVING A COMPLEX EQUATION. THE FIRST PART OF THE FORMULAE HAD TO BE SOLVED ON COMPUTER A. THE RESULT OBTAINED WAS SENT VIA THE SERIAL LINK TO COMPUTER B, & THE VALUE USED TO COMPLETE A SUB CALCULATION, THEN THIS RESULT WAS SENT BACK TO COMPUTER A TO COMPLETE THE SOLVING OF THE REST OF THE FORMULAE. THE OBVIOUS THING WAS THAT THE SUBROUTINE WRITTEN IN THE FIRST MODULE WOULD FORM THE BASIS OF THE PROGRAM NEEDED IN THE SECOND MODULE, THOUGH THIS WASN'T IMPLICITLY STATED IN THE DOCUMENTATION. THE THIRD MODULE WAS THE MOST COMPLEX. THE SCENARIO WAS THAT THE STUDENT HAD TO DEVELOP TWO PROGRAMS. COMPUTER B, USING THE NOMENCLATURE ABOVE, WAS SUPPOSED TO BE LOCATED IN A NUCLEAR REACTOR, & THE ATTACHED INTERFACE BOX WAS THE ACTUAL REACTOR. USING THE D/A CONVERTER & A RESISTOR, WHICH WAS CONNECTED VIA A HEATSINK TO THE TEMPERATURE MEASURING TRANSISTOR, WHICH WAS MONITORED BY THE A/D CONVERTER. THE IDEAL "TEMPERATURE" WAS A VALUE OF 200, BELOW THIS THE REACTION WOULD SLOW DOWN, &

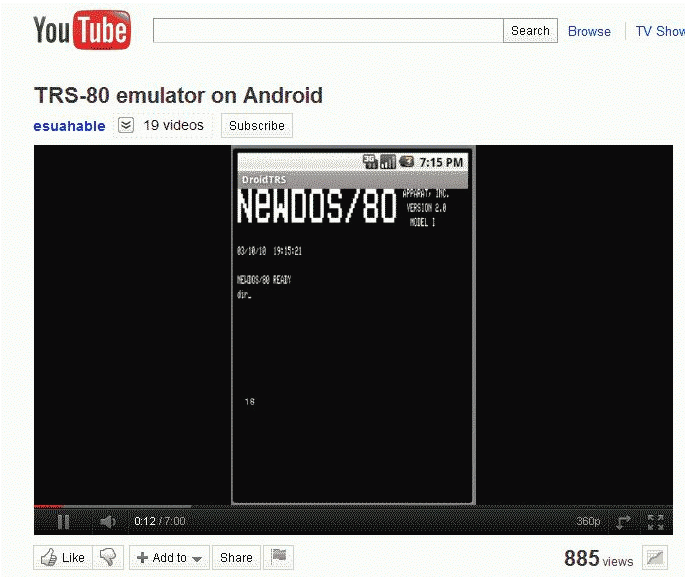
ABOVE THIS, THE REACTOR WOULD GO CRITICAL & EXPLODE. THIS MEANT THAT THE STUDENTS HAD TO GET INVOLVED WITH THREE TERM TEMPERATURE CONTROL. ANYONE INTERESTED, MIGHT LIKE TO TRY & DEVELOP THEIR OWN SUBROUTINES, BASED ON THE INFO I PROVIDED IN TRS8BIT VOLUME 3 NO 4 & VOLUME 4 NO 1.

MY OWN EXPERIMENTS INVOLVED A WIDE RANGE OF THINGS, WHATEVER TOOK MY FANCY. AN EXAMPLE WAS USING A TRS80 AS A SERVER, FOR A BBC MODEL B RUNNING IN MODE7, WHICH EMULATED A PRESTEL SYSTEM. ONE FINAL "WHAT CAN YOU DO WITH IT?" MOMENT FROM ME, INVOLVED SOME OF MY FATHERS FRIENDS WHO PAID A VISIT ONE NIGHT. I'D MOVED MY TRS80 DOWNSTAIRS & WAS USING A 110BAUD ACOUSTIC COUPLER TO COMMUNICATE WITH MY BOSS, IT WAS AN EXPERIMENT IN MANY THINGS, TESTING THE QUALITY OF THE PHONE SYSTEM, HOW FEASIBLE WAS IT, & OTHER STUFF LIKE THAT. MY FATHERS FRIENDS WONDERED WHAT I WAS DOING, & I EXPLAINED. THE COMMENT WAS MADE THAT YOU COULD SIMPLY PICK UP THE PHONE & TALK. I POINTED OUT, THAT AN ELECTRONIC MESSAGE COULD BE RECEIVED BY SOMEONE IN AUSTRALIA WITHIN 12 HOURS OF SENDING IT. I HADN'T REALLY THOUGHT ABOUT IT, BUT I THINK I WAS OUT BY QUITE A BIT, AS EMAILS THESE DAYS CAN BE SENT WITHIN SECONDS. I WAS BASING MY ORIGINAL COMMENT ON THE PACKET SWITCHED STREAM NETWORK THAT WE'D HAD INSTALLED AT THE POLY, & HAD BEEN EXPERIMENTING WITH.

---=== o0o ===---



(CONTINUED FROM PAGE 1)
BEEN LOOKING OUT TO SEE IF THERE WAS ANY POSSIBILITY OF PORTING A MODEL 1 EMULATOR TO THE ANDROID OPERATING SYSTEM. (I 'WON' AN EL-CHEAPO TABLET ON EBAY AND WAS QUITE IMPRESSED WITH THE BUILD AND DISPLAY QUALITY). I THEN CAME ACROSS A LINK ON, OF ALL PLACES, YOU-TUBE SHOWING NEWDOS80 RUNNING ON A MOBILE 'PHONE. THERE'S A LINK FOR YOU TO USE ON OUT WEBSITE.



THE APP WAS CALLED TRSDROID BUT THERE WERE NO DETAILS OF WHO HAD WRITTEN IT, OR ANY CONTACT DETAILS. THE SMALL AMOUNT OF INFORMATION THAT WAS AVAILABLE, MENTIONED THAT TIM MANN'S UNIX EMULATOR HAS BEEN USED AS A BASIS FOR THE EMULATOR, SO I PESTERED TIM WITH AN EMAIL ASKING IF HE HAD ANY FURTHER INFORMATION RE DEVELOPMENT AND PROGRESS WITH AVAILABILITY. TIM VERY KINDLY REPLIED IMMEDIATELY AND SENT A COPY OF THE ORIGINAL EMAIL HE HAD RECEIVED.

I CONTACTED THE AUTHOR OF TRSDROID, FRANK CERVENKA AND HE EMAILED ME AND STATED THAT HE HOPES TO HAVE AN APP AVAILABLE WITHIN THE NEXT 2-3 MONTHS. HE WILL KEEP US INFORMED OF DEVELOPMENTS SO WATCH THE WEBSITE FOR FURTHER INFORMATION. AS SOON AS IT'S AVAILABLE, I'LL POST IT UP. WHAT AN XMAS PRESENT EH! A MODEL 1 RUNNING ON AN ANDROID 'PHONE/TABLET!!
I HOPE FRANK GETS SOME SUPPORT AND ENCOURAGEMENT WITH HIS APP, I, FOR ONE, CAN'T WAIT TO GET MY HAND ON A COPY.

AND FINALLY, WHAT HAVE WE GOT FOR YOU IN THIS ISSUE?

AT THE READY PROMPT HAS A COUPLE OF SCHOOL-BOY IDEAS ABOUT THE NUMBER NINE!

THERE'S A FASCINATING ARTICLE FROM IAN MAVRIC, IN AUSTRALIA, DETAILING HIS WORK ON RESTORING AND RE-CREATING TANDY'S. HE'S PROMISED FURTHER DETAILS FOR THE XMAS ISSUE. IT SEEMS HE'S WORKING ON A 4, LOW PROFILE DRIVE, M4. YOU CAN USE THE LINK ON THE WEBSITE TO CHECK OUT WHAT HE'S UP TO.

I CAME ACROSS A SUPER LITTLE ADVERT FOR ACCEL 3 AND FOUND A REVIEW TO GO WITH IT. I HOPE YOU LIKE THE ADVERTS I'M FEATURING, BY THE WAY, I TRY AND FIND THE 'NOT SO WELL KNOWN' ONES!

FOR THE 16K LEVEL 2'ERS (IF THERE ARE ANY, BY THE WAY) I'VE FOUND A LITTLE SOMETHING FOR YOU WHICH HELPS WHEN EDITING.

AND FINALLY, PETER STONE, ONCE AGAIN, CAPTURES OUR INTEREST WITH HIS PROGRESS INTO THE TANDY ERA.

TAKE CARE EVERYONE

DUSTY



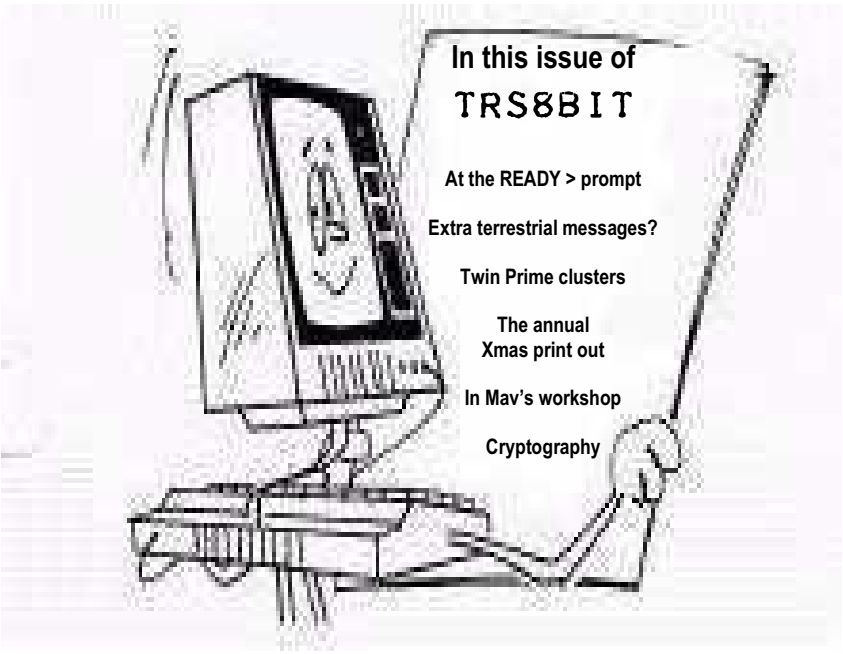
TRS8BIT



FIRST AND FOREMOST DEE AND I WOULD LIKE TO WISH YOU ALL -
A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR.

MANY THANKS FOR YOUR INTEREST SUPPORT AND ENCOURAGEMENT WITH TRS8BIT AND THE WEBSITE OVER THE LAST 5 YEARS. WE HAD OVER 90 DOWNLOAD OF THE JUNE EDITION OF THE NEWSLETTER AN ALL TIME RECORD!

QUITE OUT OF THE BLUE I HAD A PHONE CALL FROM LEIGHTON DAVIES. (HE WAS THE M1 LIBRARIAN FOR MANY YEARS). IT WAS GREAT TO HEAR FROM HIM. IT MUST BE 20 ODD YEAR THAT WE LAST MET UP. HE HAS A LARGE AMOUNT OF TANDY EQUIPMENT WHICH HE IS PUTTING UP FOR SALE ON EBAY. KEEP AN EYE OUT THERE'S A M2 M12 AND A M16 (WITH ALL THE BITS ATTACHED) THAT SHOULD BE APPEARING IN THE NEXT FEW MONTHS.



I'VE BEEN KEENLY WATCHING THE HIT COUNTER ON THE WEB SITE AND WITH ANY LUCK WE SHOULD TOP THE 10K MARK EARLY IN THE NEW YEAR. WE HAD A 'FREE' ADVERT ON IRA'S WEB SITE WHICH SEEMS TO HAVE GENERATED ADDITIONAL INTEREST.
I'VE NOTICED THAT IRA HAS SETUP A 'REPAIRS REGISTER' GIVING DETAILS OF ANYONE WHO CAN ASSIST IN HARDWARE REPAIRS. AS HE'S U.S. BASED IT'S NOT OF GREAT HELP TO PEOPLE IN THE U.K. BECAUSE OF POSTAGE COSTS ETC. BUT I'M MORE THAN HAPPY TO SETUP A U.K. REGISTER ON THE WEB SITE AND IN TRS8BIT IF THERE'S ANYONE OUT THERE WISHING TO PUT THEIR NAME FORWARD!
I KNOW MAV'S VERY KEEN TO ESTABLISH A TRS-80 BUSINESS OUT IN AUSTRALIA.

DOES ANYONE FANCY A 1 DAY WORKSHOP OR JUST A BIT OF A GET-TOGETHER? I'M QUITE HAPPY TO TRY AND ORGANISE SOMETHING IF SAY 5 OR 6 PEOPLE LIKE THE SOUND OF IT. PLEASE EMAIL ME IF YOU ARE INTERESTED.

I THINK ONCE AGAIN I'VE MANAGED TO COLLATE A FANTASTIC AND VARIED GROUP OF ARTICLES FOR YOUR DELIGHT. I HOPE YOU AGREE AND FIND THEM ENJOYABLE. ALL THAT'S LEFT FOR ME TO DO IS ONCE AGAIN WISH YOU ALL A VERY MERRY XMAS AND I LOOK FORWARD TO YOUR CONTINUED SUPPORT IN 2012 OUR 6TH YEAR!



BYE FOR NOW

DUSTY

AT THE
READY>
PROMPT



HERE'S A LITTLE

SNIPPET TO EXTRACT THE INDIVIDUAL DIGITS IN A THREE CHARACTER NUMBER WITHOUT USING ANY STRING FUNCTIONS.
FOR EXAMPLE, THIS WILL PLACE THE INDIVIDUAL CHARACTERS OF THE NUMBER 864 INTO THE VARIABLES B(1), B(2) AND B(3).

```
10 A=864
20 B(1)=INT(A/100)
30 B(2)=INT(A/10)-B(1)
40 B(3)=A-B(1)*100-B(2)*10
50 PRINT B(1);B(2);B(3)
```

FROM MICHAEL JAMES.
ONE OF THE FIRST (PSEUDO) RANDOM NUMBER GENERATORS, THE MID SQUARE METHOD, WAS SUGGESTED BY VON NEUMANN IN 1951. IT IS EASY TO USE BUT GENERATES FAIRLY LOW QUALITY RANDOM NUMBERS. IT HAS A TENDENCY TO PRODUCE NUMBERS LIKE 00YY AND XY00 PERIODICALLY, BUT IS EASY TO USE AND UNDERSTAND.

1. SPECIFY THE NUMBER OF DIGITS TO BE GENERATED, SAY FOUR.
2. CHOOSE ANY STARTING VALUE, SAY 5069.
3. SQUARE THE STARTING VALUE-25694761
4. THE NEXT RANDOM NUMBER IS THE MIDDLE FOUR DIGITS-6947
5. STEPS 3 AND 4 ARE REPEATED.

A SHORT BASIC PROGRAM FOR THE MID SQUARE METHOD FOLLOWS, IN THE HOPE YOU'LL HAVE SOME XMAS FUN EXPERIMENTING WITH IT

```
10 INPUT"STARTING VALUE";A
20 L=LEN(STR$(A))
30 P=10↑(INT(L/2))
40 Q=10↑L
50 A=INT(A*A/P)
60 A=A-INT(A/Q)*Q
70 PRINT A;
80 GOTO 50
```

EXTRA TERRESTRIAL MESSAGES?

E.T. FONHUME

HELLO DUSTY, HERE'S AN AMAZING DISCOVERY WHICH I THOUGHT OTHER TRS8BIT READERS MIGHT BE INTERESTED IN!!

CONSIDER THE MESSAGE "GREETINGS EARTHLINGS" ... IF YOU CONVERT THE LETTERS INTO MORSE CODE YOU GET

GREETINGS
DASH DASH DOT
DOT DASH DOT
DOT
DOT
DASH
DOT DOT
DASH DOT
DASH DASH DOT
DOT DOT DOT

EARTHLINGS
DOT
DOT DASH
DOT DASH DOT
DASH
DOT DOT DOT DOT
DOT DASH DOT DOT
DOT DOT
DASH DOT
DASH DASH DOT
DOT DOT DOT

TAKE DOT TO BE "ZERO" AND DASH TO BE "ONE" THE MORSE CODE REPRESENTATION GIVES THE BINARY NUMBERS

GREETINGS
11Ø Ø1Ø Ø Ø 1 ØØ 1Ø
11Ø ØØØ

EARTHLINGS
Ø Ø1 Ø1Ø 1 ØØØØ Ø1ØØ
ØØ 1Ø 11Ø ØØØ

NOW CONVERT THESE BINARY NUMBERS TO DECIMAL. (YOU CAN USE ANY OF THE AVAILABLE WEBSITES ON THE NET TO DO JUST THAT)

11ØØ1Ø ØØ1ØØ1Ø11ØØØØ
= 41Ø8ØØ
ØØ1Ø1Ø1ØØØØØ1ØØØØ1Ø11ØØØØ
= 2754648

NOW IF YOU CHECK THE VALUE WITHIN THE PI CALCULATIONS ON THE INTERESTING WEB SITE

WWW.ANGIO.NET/PI/BIGPI.CGI

IT COMES UP WITH THE ASTONISHING RESULT THAT THESE VALUES ACTUALLY OCCUR IN THE CALCULATION OF PI !!!

THE STRING 41Ø8ØØ (GREETINGS) OCCURS AT POSITION 56Ø□852 COUNTING FROM THE FIRST DIGIT AFTER THE DECIMAL POINT.

THE STRING 2754648 (EARTHLINGS) OCCURS AT POSITION 11□655□1Ø7 COUNTING FROM THE FIRST DIGIT AFTER THE DECIMAL POINT.

CLEARLY EXTRA TERRESTRIAL BEINGS ARE ATTEMPTING COMMUNICATION WITH PLANET EARTH THROUGH MATHEMATICAL CONSTANTS !!!!

KEEP UP THE GOOD WORK WITH THE NEWSLETTER AND THE WEBSITE□ THEY'RE BOTH BRILLIANT !
**
THANKS FOR YOUR ARTICLE AND KIND WORDS ET□ THIS REALLY TICKLED MY IMAGINATION. MY HOME PHONE NUMBER IS THERE TOO! A WORRYING THOUGHT EH?

HAVE YOURSELF A WONDERFUL CHRISTMAS.

DUSTY
**

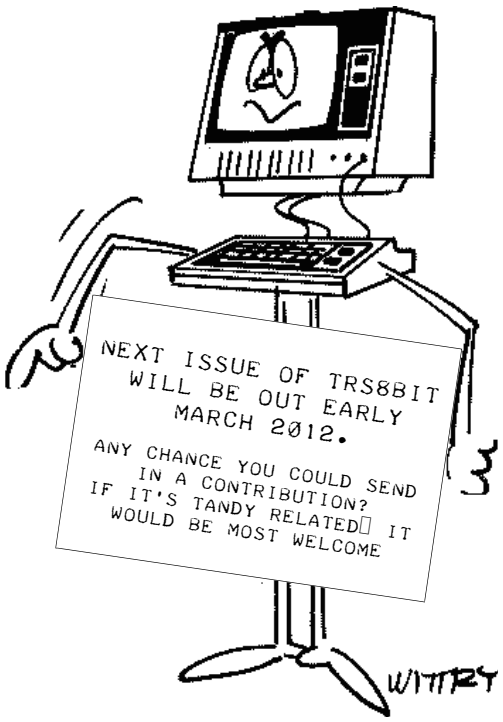
INTRODUCTION TO TWIN PRIME CLUSTERS

ROGER HARGRAVE

PEOPLE HAVE BEEN FASCINATED BY PRIMES - INTEGER NUMBERS GREATER THAN 1 WHICH ARE ONLY DIVISIBLE BY THEMSELVES AND 1 - SINCE THE DAYS OF ANCIENT GREECE. EUCLID PROVED THAT THEIR NUMBER IS INFINITE□ AND OVER THE YEARS MATHEMATICIANS HAVE DISCOVERED AND PROVED THAT EVER LARGER NUMBERS ARE PRIME. FROM TIME TO TIME THE MEDIA REPORT THAT ANOTHER□ BIGGER EXAMPLE HAS BEEN FOUND AND PROVED TO BE PRIME□ USING LARGE COMPUTERS AND SOPHISTICATED TECHNIQUES. THIS HAS BECOME AN ACTIVITY WAY BEYOND THE EXPERTISE AND EQUIPMENT AVAILABLE TO MOST OF US□ AND WE HAVE TO ACCEPT THE WORD OF PROFESSIONAL MATHEMATICIANS THAT THE RECORD HAS BEEN BROKEN. HOWEVER□ THERE IS ONE ASPECT OF PRIMES WHERE AN INTERESTED AMATEUR CAN BEGIN TO GAIN SOME INSIGHT WITH THE AID OF A PERSONAL COMPUTER AND LITTLE MORE THAN THE MATHEMATICS LEARNT AT SCHOOL. THIS IS THE MYSTERY OF THEIR DISTRIBUTION. IT IS TOO MUCH TO HOPE THAT A GENERAL EXPLANATION WILL BE FOUND. THAT HAS ELUDED THE BEST BRAINS TACKLING THE PROBLEM OVER MORE THAN TWO MILLENNIA. NEVERTHELESS□ THERE ARE CERTAIN SYMMETRICAL GROUPINGS OF PRIMES WHOSE ANALYSIS CAN ELUCIDATE□ NOT A GENERAL RULE OR FORMULA THAT WILL INVARIABLY PRODUCE EXAMPLES OF SUCH GROUPS□ BUT AT LEAST SOME OF THE CONDITIONS WHICH MUST BE MET FOR SUCH EXAMPLES TO EXIST.

TWIN PRIMES

THE SIMPLEST OF THESE GROUPS ARE CALLED TWIN PRIMES THEY CONSIST OF TWO PRIMES WITH A DIFFERENCE OF 2, SUCH AS 11 AND 13, OR 29 AND 31. THE PRIMES IN EACH OF THESE SETS ARE SEPARATED BY A COMPOSITE NUMBER - ONE WHICH IS THE PRODUCT OF TWO OR MORE PRIMES - WHICH FOR WANT OF A BETTER NAME I CALL A PIVOTAL COMPOSITE, PIVOTAL BECAUSE IT IS THE PIVOT OF A SYMMETRICAL GROUP OF PRIMES. THE TERM PIVOTAL COMPOSITE, THOUGH DESCRIPTIVE, IS INCONVENIENTLY LONG, SO IN WHAT FOLLOWS IT WILL BE SHORTENED TO PIVCOM . IT IS WELL-KNOWN THAT ALL TWIN PRIMES ARE OF THE FORM 6N-1 AND 6N+1. ALL PRIMES EXCEPT 2 ITSELF ARE ODD NUMBERS, AND THE TWO ADJACENT NUMBERS CAN ONLY BE PRIME IF THE PIVCOM IS DIVISIBLE BY THREE, SINCE OTHERWISE ONE OF THEM WOULD BE A MULTIPLE OF THREE. THEREFORE EVERY PAIR OF TWIN PRIMES HAS A MULTIPLE OF 6 AS ITS PIVCOM. NOTE THAT ALTHOUGH IT IS NECESSARY FOR THE PIVCOM TO BE A MULTIPLE OF 6, IT IS NOT SUFFICIENT IN ALL CASES. FOR EXAMPLE, IF N=4, 6N=24 AND 6N+1=25, WHICH IS NOT A PRIME. (AS A MATTER OF INTEREST, THE LOWEST VALUE OF N FOR WHICH NEITHER OF THE ADJACENT NUMBERS IS PRIME IS 20.)



PRIME QUADRUPLETS

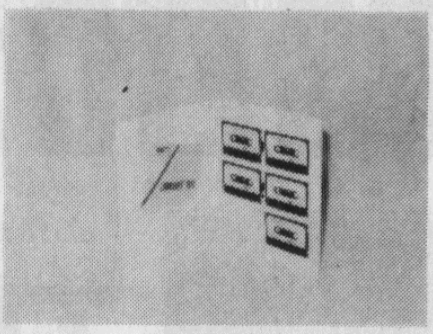
A LARGER SYMMETRICAL GROUP IS THE PRIME QUADRUPLET. THESE ARE MADE UP OF TWO SETS OF TWIN PRIMES AS CLOSE TOGETHER AS THEY CAN BE. HERE IT TURNS OUT THAT THE PIVCOM HAS TO BE A ODD MULTIPLE OF 15 AND THE DIFFERENCES OF THE PRIMES IN THE GROUP FROM THE PIVCOM ARE -4, -2, +2, +4, THE REASON WHY THIS IS SO CAN BE UNDERSTOOD BY USING THE CONCEPTS OF MODULUS AND RESIDUE . WHEN ONE INTEGER IS DIVIDED BY ANOTHER (CALLED A MODULUS), IT EITHER DIVIDES EXACTLY TO GIVE A THIRD INTEGER, OR ELSE THERE IS A RESIDUE OR REMAINDER. FOR EXAMPLE, 7 DIVIDES 35 EXACTLY TO GIVE 5, BUT 38 DIVIDED BY 7 GIVES 5 AND A RESIDUE OF 3. SO WE CAN SAY 38 TO THE MODULUS OF 7 GIVES A RESIDUE OF 3, OR IN SHORT 38(MOD 7) = 3. TWO NUMBERS WHICH GIVE THE SAME RESIDUE TO A MODULUS ARE SAID TO BE CONGRUENT WITH RESPECT TO THAT MODULUS. DEVELOPING THIS IDEA A LITTLE FURTHER, IF BOTH A MODULUS AND THE RESIDUE OF A CERTAIN NUMBER IN RESPECT OF THAT MODULUS ARE DIVISIBLE EXACTLY BY A

(CONTINUED ON PAGE 7)

LIBRARY 100

for the

TRS-80



A complete library of 100 programmes for the TRS-80 Level II. The 5 cassettes are attractively packaged in a plastic binder which also holds the full documentation. All programmes were specifically written for the Library and, as a bonus, include a Tiny version of the Pilot language, to give your TRS-80 the added dimension of a new language. Less than 50p a programme !!

Finance: Present Value of a Future Sum—Simple Interest for Days— Future Value of a Present Sum—Amortization Schedule—Interest Rate: Compound Interest—Interest Rate: Installment Loan—Days Between Dates—Term of an Installment Loan—Present Value of a Series of Payments—Real Estate Capital Investment—Nominal and Effective Interest Rates—Internal Rate of Return—Future Value of Regular Deposits—Regular Deposits for Future Value—Depreciation Amount: Rate; Salvage Value; Schedule—Bond Present Value—Bond Yield to Maturity—Sale—Cost—Margin—Day of the Week—Moving Ad.

Education: Multiplication & Division—Add—Subtract—Fraction & Decimal—States & Capitals—States & Order of Entry—States & Abbreviation—Inventors & Inventions—World Capitals & Countries—Urban Areas & Population—Authors & Books—Presidents & Order—States & Largest City—Basenum.

Graphics: Left Right—Random Ad—Graphic—Blocks—Fireside— Snow—Step Ad—Step Ad 2—Launch—Ratrace—War Game—Weird—Herring—Blinker—Snoopy.

Home: Message Board—Expense Account—Nutrition—Mileage— Remember—Phone Codes—Night Check Off—Drunkometer—Perpetual Calendar—Babysitter—Calculator—Bartender—Christmas List—Vacation Check Off—Conversion.

Games: Speedy—Odd One—R. Roulette—Star Blazer—Search— Spyship—Tiger Shark—Jumble 2—Sting Ray—Stars—Sketch—Flipper—Scissors—Horse—Doomsday—Craps—Jumble 1—Mem. Quiz Letters—Mem. Quiz Numbers—Wheel of Fortune—Decision—Unjumble—Fifteen—Towers—Life—Star Trek—Race Track—Count—Roachrace—Gypsy.

£46.95 (inc)

A. J. HARDING

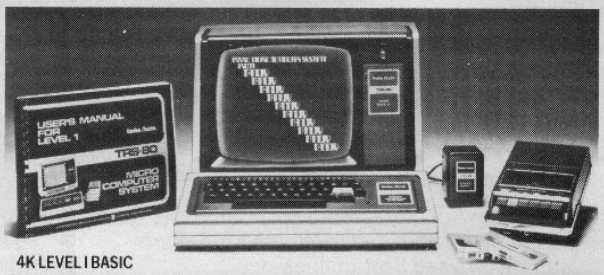
28 Collington Ave., Bexhill, E. Sussex. (0424) 220391.

The TANDY TRS 80 the ultimate in personal computers -right off the shelf.

4K LEVEL I BASIC

Low cost breakthrough!
*TRS-80 Micro-computer
*12" video display
*Easy 'typewriter' keyboard.
*Power transformer
*Cassette tape recorder
*4K RAM, Level-1 BASIC
*232 page manual
*2 game cassettes
26-1001

£499

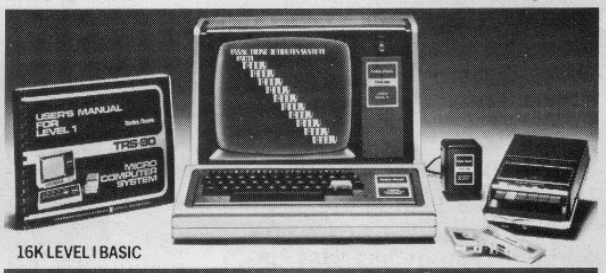


4K LEVEL I BASIC

16K LEVEL I BASIC

The same hardware as the 4K system but with 16K RAM it gives greater data storage and programming capacity. 26-1003

£728

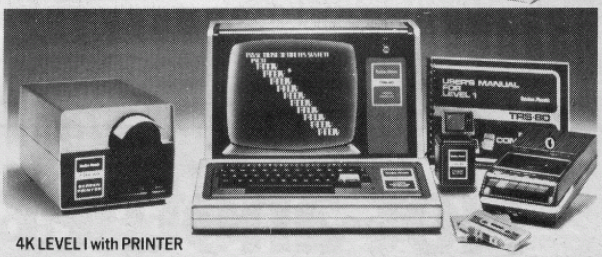


16K LEVEL I BASIC

4K LEVEL I with PRINTER

Our basic system but with a hard-copy screen printer. At the touch of a button the screen display is printed out on electrostatic paper as a permanent visual record. (26-1001/26-1151)

£978

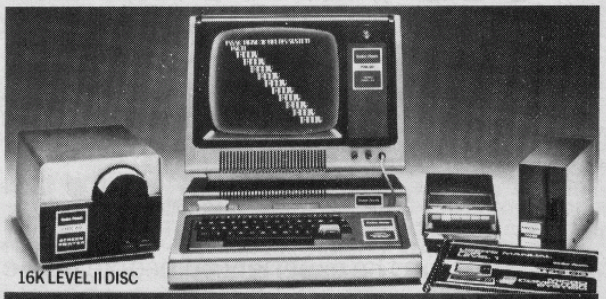


4K LEVEL I with PRINTER

16K LEVEL II DISC

Professional 16K RAM operation with mini-disc, screen printer and expansion interface. (26-1006/26-1151/26-1140/26-1160)

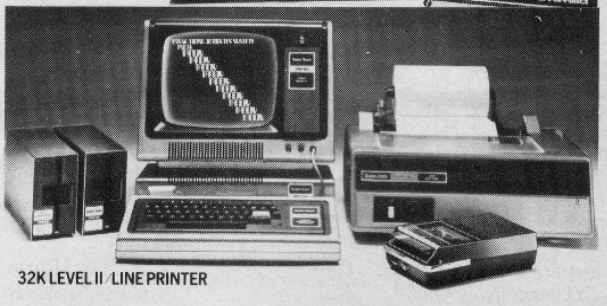
£1914


16K LEVEL II DISC

32K LEVEL II/LINE PRINTER

This is the ultimate! 32K RAM, 2 mini-discs, line printer, expansion interface, 16K memory option. A full business system.
(26-1006/26-1160/26-1161/26-1150/26-1140/26-1101).

£3058


32K LEVEL II/LINE PRINTER

See the TRS-80 micro-computer breakthrough at most Tandy stores and dealerships or write now for further data to:-

Tandy Corporation, Computer Division, Bilston Road, Wednesbury, Staffordshire WS10 7JN.

TANDY

The biggest name in little computers.

(CONTINUED FROM PAGE 4)
THIRD NUMBER, THEN THE NUMBER ITSELF MUST ALSO BE DIVISIBLE BY THE THIRD NUMBER. FOR EXAMPLE, 51 (MOD 9)=6; BOTH 9 AND 6 ARE DIVISIBLE BY 3; THEREFORE 51 MUST ITSELF BE DIVISIBLE BY 3.
NOW, WE ARE LOOKING FOR A PIVCOM WHICH CAN SIT BETWEEN TWO SETS OF TWIN PRIMES AS CLOSELY AS POSSIBLE AND EQUIDISTANT FROM THEM. THERE ARE TWO POSSIBLE SORTS OF PIVCOM FOR A SYMMETRICAL GROUP OF PRIMES. IF THE PIVCOM IS AN ODD NUMBER, THEN THE DISTANCE FROM THE CENTRE OF EACH PRIME MUST BE A MULTIPLE OF TWO, BUT IF IT IS EVEN, THEN THE DISTANCES MUST BE ODD, BECAUSE ALL PRIMES EXCEPT 2 ITSELF ARE NECESSARILY ODD. IT IS FORTUNATE THAT WE ARE ALL ACCUSTOMED TO A NUMBER SYSTEM BASED ON TEN, SINCE IT MEANS THAT IT IS EASY WHEN THINKING ABOUT PRIMES TO ELIMINATE ANY NUMBER (EXCEPT 2 AND 5) WHICH DOES NOT HAVE AS ITS LAST DIGIT 1 OR 3 OR 7 OR 9.
NO THREE SUCCESSIVE ODD NUMBERS N, N+2, N+4, WHERE N IS GREATER THAN 3, CAN ALL BE PRIME, SINCE ONE OF THEM WILL ALWAYS BE DIVISIBLE BY 3. THIS MEANS THAT SEQUENCES OF SUCCESSIVE ODD NUMBERS ENDING WITH THE DIGITS 3,7,9,1 OR 9,1,3,7 OR 7,9,1,3 NECESSARILY CONTAIN AT LEAST ONE COMPOSITE NUMBER. SO WE ARE LEFT WITH 1,3,7,9 IN THAT ORDER AS THE LAST DIGITS OF A PRIME QUADRUPLET. FROM THIS WE CAN SEE THAT WE NEED A PIVCOM ENDING IN (AND THEREFORE DIVISIBLE BY) 5, WITH DISTANCES OF -4, -2, +2, +4. CONSIDER NOW DIVISIBILITY BY 3. IF THE PIVCOM IS CONGRUENT TO 1 OR 2(MOD3) THEN EITHER THE +2 OR THE +4 NUMBER IS DIVISIBLE BY 3 AND NOT A PRIME. THEREFORE THE PIVCOM MUST BE CONGRUENT TO Ø (MOD3), IN OTHER WORDS, DIVISIBLE BY 3 AS WELL AS BY 5. WE HAVE ALREADY SEEN THAT WHERE THE DISTANCES ARE EVEN THE PIVCOM MUST BE ODD, SO WE CAN ALSO SPECIFY THAT IT MUST BE AN ODD MULTIPLE OF 15. FROM THIS IT FOLLOWS THAT THE NEAREST THE PIVCOMS OF TWO PRIME QUADRUPLETS CAN BE TO EACH OTHER IS 3Ø. THAT IS WHY A GROUP OF TWO PRIME NUMBER QUADRUPLETS IS SPECIAL AND BRINGS US TO THE MAIN SUBJECT OF THIS ARTICLE.

PNQ3ØS

THE ABBREVIATION PNQ3Ø WAS COINED TO STAND FOR "PRIME NUMBER QUADRUPLETS WITHIN 3Ø CONSECUTIVE INTEGERS", SINCE PQ3Ø HAS MEANING IN OTHER CONTEXTS. FOR CONVENIENCE OF LISTING, THE FIRST PRIME OF THE FIRST QUADRUPLET IS USED, AND ALWAYS ENDS WITH 1.
FOR A START, WE KNOW THAT THE PIVCOM OF A PNQ3Ø MUST BE DIVISIBLE BY 3Ø, SINCE IT IS THE EVEN MULTIPLE OF 15 WHICH LIES BETWEEN TWO CONSECUTIVE ODD MULTIPLES OF 15, NAMELY THE PIVCOMS OF THE TWO PRIME QUADRUPLETS. SECONDLY, SINCE THE PIVCOM OF A PNQ3Ø DIFFERS FROM THE PIVCOMS OF ITS TWO QUADRUPLETS BY-15 AND +15 RESPECTIVELY, AND EACH OF THESE DIFFER BY -4, -2, +2 AND +4 FROM THE PRIMES OF WHICH THEY ARE COMPOSED, IT FOLLOWS THAT THE DISTANCES OF THE PRIMES OF A PNQ3Ø FROM ITS PIVCOM ARE -19, -17, -13, -11, +11, +13, +17, AND +19. THESE DISTANCES IMPLY THAT THE PIVCOM OF A PNQ3Ø IS NOT DIVISIBLE BY 11, 13, 17

OR 19, SINCE DIVISIBILITY BY ANY ONE OF THESE FACTORS WOULD MEAN THAT TWO OF THE NUMBERS GIVEN BY THE DISTANCES, ONE EACH SIDE OF THE PIVCOM, WOULD ALSO BE COMPOSITE. WE HAVE NOW ACCOUNTED FOR THE EFFECT ON THE PIVCOM OF ALL THE PRIMES UP TO 19, EXCEPT ONE, NAMELY 7. THE PIVCOM IS DIVISIBLE BY 2, 3 AND 5, AND NOT DIVISIBLE BY THE OTHERS. LET US NOW LOOK AT HOW THE PIVCOM OF A PNQ3Ø STANDS IN RELATION TO DIVISIBILITY BY 7.
IT WAS MENTIONED ABOVE THAT A NUMBER WHICH HAS A RESIDUE IN RESPECT OF A MODULUS WHICH IS DIVISIBLE BY A FACTOR OF THAT MODULUS IS ALSO DIVISIBLE BY THAT FACTOR. THERE ARE SEVEN POSSIBLE RESIDUES IN RESPECT OF MODULUS 7. THESE ARE 1, 2, 3, 4, 5, 6 AND Ø.
IF PIVCOM(MOD7) = 1, THE POTENTIAL PRIME WITH DISTANCE +13 FROM THE PIVCOM WOULD BE CONGRUENT TO 14(MOD 7) AND IS THEREFORE ALSO DIVISIBLE BY SEVEN AND SO NOT PRIME.

IF PIVCOM(MOD7) = 2, THE POTENTIAL PRIME WITH DISTANCE +19 WOULD BE CONGRUENT TO 21(MOD7).
IF PIVCOM(MOD7) = 3, THE POTENTIAL PRIME WITH DISTANCE +11 WOULD BE CONGRUENT TO 14(MOD7)
IF PIVCOM(MOD7) = 4, THE POTENTIAL PRIME WITH DISTANCE -11 WOULD BE CONGRUENT TO -7(MOD7).
IF PIVCOM(MOD7) = 5, THE POTENTIAL PRIME WITH DISTANCE -19 WOULD BE CONGRUENT TO -14(MOD7).
IF PIVCOM(MOD7) = 6, THE POTENTIAL PRIME WITH DISTANCE -13 WOULD BE CONGRUENT TO -7(MOD7).
THEREFORE ONLY IF PIVCOM (MOD7) = Ø ARE ALL THE POTENTIAL PRIMES ABLE TO EXIST, SO ONLY IN THAT CASE CAN A PNQ3Ø EXIST. SINCE THIS IS EQUIVALENT TO SAYING THE PIVCOM MUST BE DIVISIBLE BY 7, THE REQUIREMENT FOR A PIVCOM TO BE A MULTIPLE OF 3Ø CAN BE INCREASED TO A MULTIPLE OF 21Ø. TO SUMMARISE, A NECESSARY REQUIREMENT FOR A PNQ3Ø IS THAT ITS PIVCOM MUST BE A MULTIPLE OF 21Ø BUT NOT OF 11, 13, 17 OR 19; AND THAT THE DISTANCES OF THE PRIMES FROM THE PIVCOM MUST BE -19, -17, -13, -11, +11, +13, +17, AND +19. AS WITH THE OTHER PIVCOMS, IT IS NOT SUFFICIENT FOR THIS SPECIFICATION TO BE MET TO PRODUCE A PNQ3Ø, BUT UNLESS IT IS MET, NO PNQ3Ø CAN EXIST.

5TP39

FURTHER CONSIDERATION OF THE ABOVE LED TO THE REALISATION THAT THERE WAS A POSSIBILITY THAT IN RARE CASES, SINCE THE PIVCOM OF A PNQ3Ø IS NECESSARILY DIVISIBLE BY 6, IT MIGHT ALSO ITSELF BE THE PIVCOM OF A TWIN PRIME. THIS WOULD MAKE FIVE TWIN PRIMES WITHIN 39 CONSECUTIVE INTEGERS, OR 5TP39 FOR SHORT. THEY ARE ALSO KNOWN AS QTPS - QUINTUPLET TWIN PRIMES.
THIS SPECULATION WAS COMMUNICATED TO D LA PIERRE BALLARD, THE PIONEER INVESTIGATOR OF PNQ3ØS - AND WITHIN A VERY FEW DAYS HE HAD DISCOVERED THE SMALLEST EXAMPLE OF SUCH A GROUPING. VERY GENEROUSLY, HE HAS CHOSEN TO NAME SUCH CLUSTERS AFTER ME, BUT IN MY ESTIMATION THEY SHOULD BE CALLED BALLARD PRIMES AFTER THIS FIRST DISCOVERY AND HIS SUBSEQUENT SKILL AND DILIGENCE IN DISCOVERING EXAMPLES.

THE XMAS PRINT-OUT HAS BECOME A BIT OF A TRADITION HERE AT TRS8BIT BUT THIS YEAR'S PRINT HAS REAL XMAS MESSAGE! AS ALWAYS I HOPE IT MAKES YOU SMILE.

HAVE A WONDERFUL XMAS

```
10 CLEAR500
20 M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMAS"
30 GOSUB 210
40 READ S
50 IF S=0 GOSUB 150
60 READ L
70 T$=MID$(M$,S,L)
80     X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256

90 REM THIS AND THE NEXT LINE ARE NOT NEEDED ON 4K or 16K SYSTEMS
100     IF X>32767 THEN X=X-65536

110         FOR Y=1TOL
120         POKE X+S+Y,ASC(MID$(T$,Y,1))
130         NEXT Y
140 GOTO 40
150 LPRINT P$
160 GOSUB210
170 READ S
180 IFS=0THENLPRINTP$:READS
190 IF S=99 THEN END
200 RETURN
210 P$=STRING$(75,32)
220 RETURN

230 DATA 1,75,0,1,75,0,1,75,0,1,2,13,3,25,6,37,15,57,8,71,5,0
240 DATA 1,6,9,7,18,6,26,4,32,4,38,12,52,5,59,5,66,4,72,4,0
250 DATA 1,6,9,7,18,7,27,2,31,18,51,7,60,3,65,6,73,2,0
260 DATA 1,6,9,7,18,6,26,4,33,17,52,5,59,4,65,6,73,3,0
270 DATA 1,6,9,7,25,7,35,5,46,5,58,5,65,6,73,3,0
280 DATA 1,6,9,7,18,3,23,11,36,14,52,5,59,4,65,6,73,3,0
290 DATA 1,6,9,7,18,4,24,12,38,11,51,7,60,3,65,6,73,3,0
300 DATA 1,6,9,7,18,5,25,4,31,4,37,13,52,5,59,5,66,4,72,4,0
310 DATA 1,6,9,7,18,6,26,4,36,16,57,8,71,5,0
320 DATA 1,75,0,1,75,0,1,75,0
330 DATA 99
```

```
MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASM
MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASM
MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASM
ME           RYX          MASMER       MERRYXMASMERRYX      RRYXMASM       ASM
MERRYX   SMERRYX   SMERRY   ASME   YXMA   ERRYXMASMERR   MASME   YXMAS   RRYX   SM
MERRYX   SMERRYX   SMERRYX   SM   RYXMASMERRYXMASMER   XMASMER   XMA   ERRYXM   M
MERRYX   SMERRYX   SMERRY   ASME    XMASMERRYXMASMERR   MASME   YXMA   ERRYXM   M
MERRYX   SMERRYX           MASMERR    ASMER        MERRY         RYXMA   ERRYXM   M
MERRYX   SMERRYX   SME   YXMASMERRYX   SMERRYXMASMERR   MASME   YXMA   ERRYXM   M
MERRYX   SMERRYX   SMER   XMASMERRYXMA   ERRYXMASMER   XMASMER   XMA   ERRYXM   M
MERRYX   SMERRYX   SMERR   MASM   RYXM   MERRYXMASMERR   MASME   YXMAS   RRYX   SM
MERRYX   SMERRYX   SMERRY   ASME        SMERRYXMASMERRYX      RRYXMASM       ASM
MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASM
MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASM
MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASM
```

IN MAV'S WORKSHOP
BY IAN MAVRIC

BUILDING THE SUPER MICRO PART 1

WAY BACK IN 1984 AN OBSCURE TRS-80 PUBLICATION CALLED COMPUTER USER RAN A 4 PART SERIES "BUILDING THE SUPER MICRO" ABOUT HOW TO UPGRADE A MODEL III INTO A "SUPER MICRO" WITH PERFORMANCE CHARACTERISTICS TO OUT-CLASS THE THEN NEWLY-RELEASED MODEL 4[1]. THESE SERIES OF ARTICLES BY ME LOOKS AT MY VERSION OF A SUPER MICRO WHICH I PUT TOGETHER MYSELF IN MY WORKSHOP AND THE PROBLEMS YOU MAY ENCOUNTER WITH THESE MACHINES WHICH ARE OVER 25 YEARS OLD NOW.

I STARTED WITH AN ABANDONED MODEL 4 THAT WAS SITTING AT THE BACK OF MY GARAGE WHICH HAD AN INTERESTING HISTORY; IN 1983 A SOON-TO-BE FAMOUS AUSTRALIAN NOVELIST CAME TO ME SAYING SHE WANTED TO DITCH HER TYPEWRITER AND BUY A WORD PROCESSOR. I EXAMINED HER REQUIREMENTS AND MINISCULE BUDGET AND SETTLED ON A 16K CASSETTE MODEL 4 SCRIPSIT-TAPE VERSION AND A DAISY WHEEL PRINTER II. MY RATIONALE AT THE TIME WAS ONCE SHE SOLD A FEW BOOKS SHE COULD UPGRADE THE COMPUTER TO A PROPER DISK SYSTEM AND THAT'S EXACTLY WHAT HAPPENED. BY 1988 IT WAS A FULL 64K DUAL DISK MODEL 4 RUNNING SUPER SCRIPSIT AND IT REMAINED THAT WAY UNTIL THE MID-90S WHEN IT WAS REPLACED WITH A MAC. I SOMEHOW RESCUED THE SYSTEM AND ITS LANGUISHED UNUSED IN MY GARAGE EVER SINCE. IT'S A PARTICULARLY EARLY ONE CAT. NO. 26-1067 S/N 0000083.
DECIDING ON THE SPECS WAS EASY REFERRING TO THE OLD CU SERIES THEY TOOK A 16K CASSETTE MODEL III MAXED OUT ITS RAM INSTALLED 4 INTERNAL FLOPPY DRIVES AN RS232 A CP/M BOARD AND AN 80 COLUMN VIDEO BOARD. SINCE THE LAST TWO ITEMS ARE ALREADY PART OF THE MODEL 4S STANDARD CAPABILITIES I THOUGHT THIS WOULD MAKE A NICE BASIS FOR MY M4 SUPER MICRO.

PROJECT STARTS

THE MODEL 4 WAS IN VERY SAD SHAPE WHEN I DUG IT OUT EARLIER THIS YEAR. IT WOULDN'T EVEN POWER UP AND I FOUND THE POWER SWITCH HAD CORRODED INTERNALLY AND NEEDED TO BE DISASSEMBLED CLEANED AND RE-ASSEMBLED (SEE PIC). THEN THE POWER SUPPLY STARTED SMOKING FROM THE MKT CAPS IMMEDIATELY THE PICTURE WAS SHAKY ON THE CRT THE DISK DRIVES DIDN'T WANT TO SPIN AND IT DIDN'T RESPOND TO KEYBOARD INPUT. IN SHORT IT'S NOT UNLIKE MOST MODEL IIIS AND 4S YOU FIND ON EBAY WHICH THE SELLER STATES "WORKED FINE WHEN PUT AWAY".

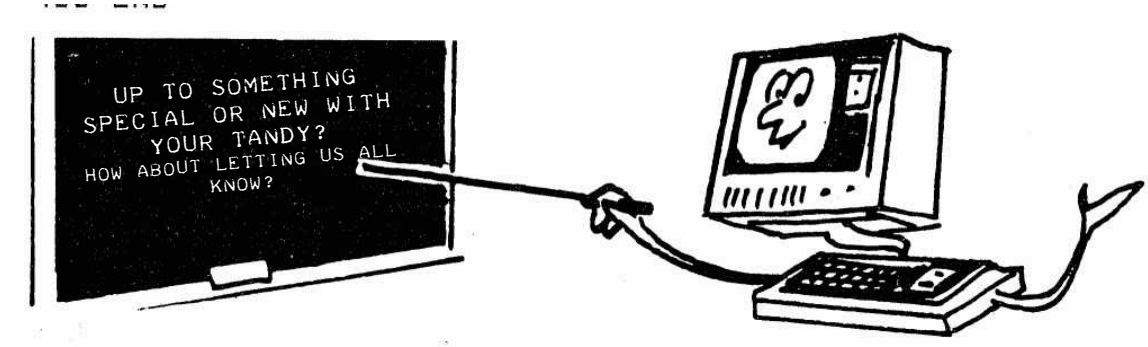
I CONTINUED WITH A COMPLETE DISASSEMBLY AND REPLACED THE MKT CAPS ON THE POWER SUPPLY (C1 C2 AND C13 ON THE 65W ASTEC 120/240V SUPPLY[2]) AS WELL AS REMOVING THE HT VIDEO BOARD AND KEYBOARD. THE HT VIDEO BOARD NEEDED ITS (SOLDER COVERED) CONTACTS CLEANED BUT AT THE SAME TIME SINCE IT WAS OUT I RE-SOLDERED ALL JOINTS AS THESE HAVE BEEN KNOWN

TO HAVE VERY HARD TO DETECT DRY JOINTS FORM ON THEM. THE KEYBOARD ITSELF WAS VERY SAD; ALL THE KEYCAPS HAD BEEN WORN SHINY DUE TO THOUSANDS OF HOURS OF TYPING AND OF THE 70 KEYS ABOUT 50 OF THEM SHOWED UP NO RESPONSE ON A MULTI METER REQUIRING EACH OF THOSE TO BE UNSOLDERED CLEANED THEN RE-SOLDERED BACK IN..AN EXTREMELY REPETITIVE AND TIME CONSUMING TASK (SEE PIC). KNUT'S WEB SITE SHOWS HOW TO FIX THE KEYBOARDS[3]. WITH THE DISK CONTROLLER DISCONNECTED I WAS ABLE TO FINALLY GET THE COMPUTER TO START TO "CASS?" PROMPT WITH A STABLE PICTURE AND NO SMOKING OR OTHER ANOMALIES FROM THE POWER SUPPLY. MOST OF THE KEYS WORKED AND BY THE END OF THE WEEKEND I HAD A PRETTY GOOD WORKING CASSETTE BASED MODEL 4 AGAIN. IT EVEN LOADED A TAPE PROPERLY.

MAX OUT THE MEMORY

64K FOR MOST USES FOR THE MODEL 4 IS PERFECTLY ADEQUATE BUT MOST OF THE MORE INTERESTING LATER SOFTWARE TOOK ADVANTAGE OF 128K OR MORE SO I FELT IT AN OPPORTUNE TIME TO UPGRADE THE MEMORY WHILE THE COMPUTER WAS IN PIECES THE RS KIT COMPRISES OF 8 4164 DRAM CHIPS A PROGRAMMED PAL AND A "128K RAM" EMBLEM FOR THE KEYBOARD AND I HAPPENED TO HAVE ONE KIT ON HAND. YOU JUST INSTALL THE 8 RAMS IN THE EMPTY SOCKETS ON THE MOTHERBOARD PRIZE OUT THE SHUNT IN U72 AND REPLACE IT WITH THE PAL AND ON THE KEYBOARD BEZEL TWO TABS HOLD THE 64K EMBLEM JUST BEND THEM REMOVE THE OLD AND INSTALL THE NEW. JOB SORTED.

NEXT TIME: INSTALLING 4 INTERNAL DISK DRIVES: THREE 80 TRACK DSDD AND ONE 40 TRACK DSDD (SEE PIC OF COMPLETED SUPER MICRO)



IN MAV'S PROBLEM SOLVING WORKSHOP

THOSE WHITE CABLES

THOSE OF YOU WITH MODEL IIIS AND EARLY NON-GATE ARRAY MODEL 4S KNOW THAT INSIDE THE COMPUTER CONNECTING THE FDC AND RS232 BOARD TO THE MAINBOARD ARE FLIMSY "SPECTRA STRIP" CONNECTORS WHICH HAVE BEEN KNOWN TO PERISH WITH AGE AND EVEN IF THEY DON'T THEY ARE JUST GENERALLY FLIMSY. THEY DON'T STAND UP TO REPEATED REMOVAL/RE-

INSERTION, WHICH WOULDN'T BE MUCH OF A PROBLEM IF THEY WERE CHEAP AND PLENTIFUL BUT TANDY HASN'T STOCKED THEM SINCE 1992, AND I'VE NEVER FOUND ANOTHER SUPPLIER SINCE, SO YOU HAVE TO TREAT THEM VERY GENTLY, AND EVEN THEN THEY BREAK EASILY.

FOR YEARS I'VE SPECULATED THAT IF WE COULD JUST RUN A REGULAR PC IDE HARD-DRIVE CABLE BETWEEN THEM OUR PROBLEMS WOULD BE SOLVED. IT WOULD BE GREAT IF YOU HAVE A MACHINE THAT YOU REGULARLY PULL APART, NOT HAVING TO WORRY ABOUT THE WHITE CABLES. I'VE NOW DONE ONE AND IT WORKS. HERE'S HOW I DID IT: I REMOVED THE BOARD AND LOOKING AT THE EXISTING CONNECTOR, LEVER IT BACK AND FORTH UNTIL IT SNAPS OFF. YOU ARE LEFT WITH 20 PINS SOLDERED INTO 20 HOLES. WITH A HOT SOLDERING IRON AND A SOLDER SUCKER YOU CAN CLEAR THE HOLE AND THE PIN IN ONE ACTION, IT TAKES ABOUT 5 MIN WITH THE MOTHERBOARD AND ABOUT TWICE AS LONG WITH THE FDC (THIS IS DUE TO THE FDC HOLES SEEMING TO BE A FRACTION SMALLER THAN THE MOTHERBOARD ONES).

BUY A 40-PIN STRIP HEADER (SUCH AS JAYCAR[4] CAT. NO. HM3212, RRP. $A0.95C), CUT IT IN HALF AND INSERT INTO THE BOARDS. IF IT WON'T GO THEN THE HOLES AREN'T CLEAR ENOUGH OF SOLDER. ONCE IN THEN SOLDER THE STRIP HEADER IN. TAKE AN IDE CABLE, THEY USUALLY HAVE TWO CONNECTORS RELATIVELY CLOSE TO EACH OTHER AND ONE DOWN THE OTHER END. CUT OFF THE LONG END AND DISCARD, INSTALL THE SHORT PIECE BETWEEN THE FDC AND MOTHERBOARD. MODIFICATION IS COMPLETE (SEE PICS).

GOOD NEWS FOR MODEL 1 OWNERS

THE SAME MODIFICATION WORKS ON THE MODEL 1 KEYBOARD CABLE WHICH IS ALSO NOTORIOUSLY FLIMSY, THE IDE CABLE MODIFICATION WORKS WELL AND YOU END UP WITH A KEYBOARD YOU CAN EASILY DISCONNECT FROM THE M1 MAIN BOARD FOR MAINTENANCE PURPOSES. SPACE IS A BIT OF A PREMIUM, AS THE IDE CABLE IS MUCH LONGER THAN THE ORIGINAL CABLE BUT IT DOES INDEED FIT IF YOU FOLD THE CABLE APPROPRIATELY.

IANM@TRS-80.COM

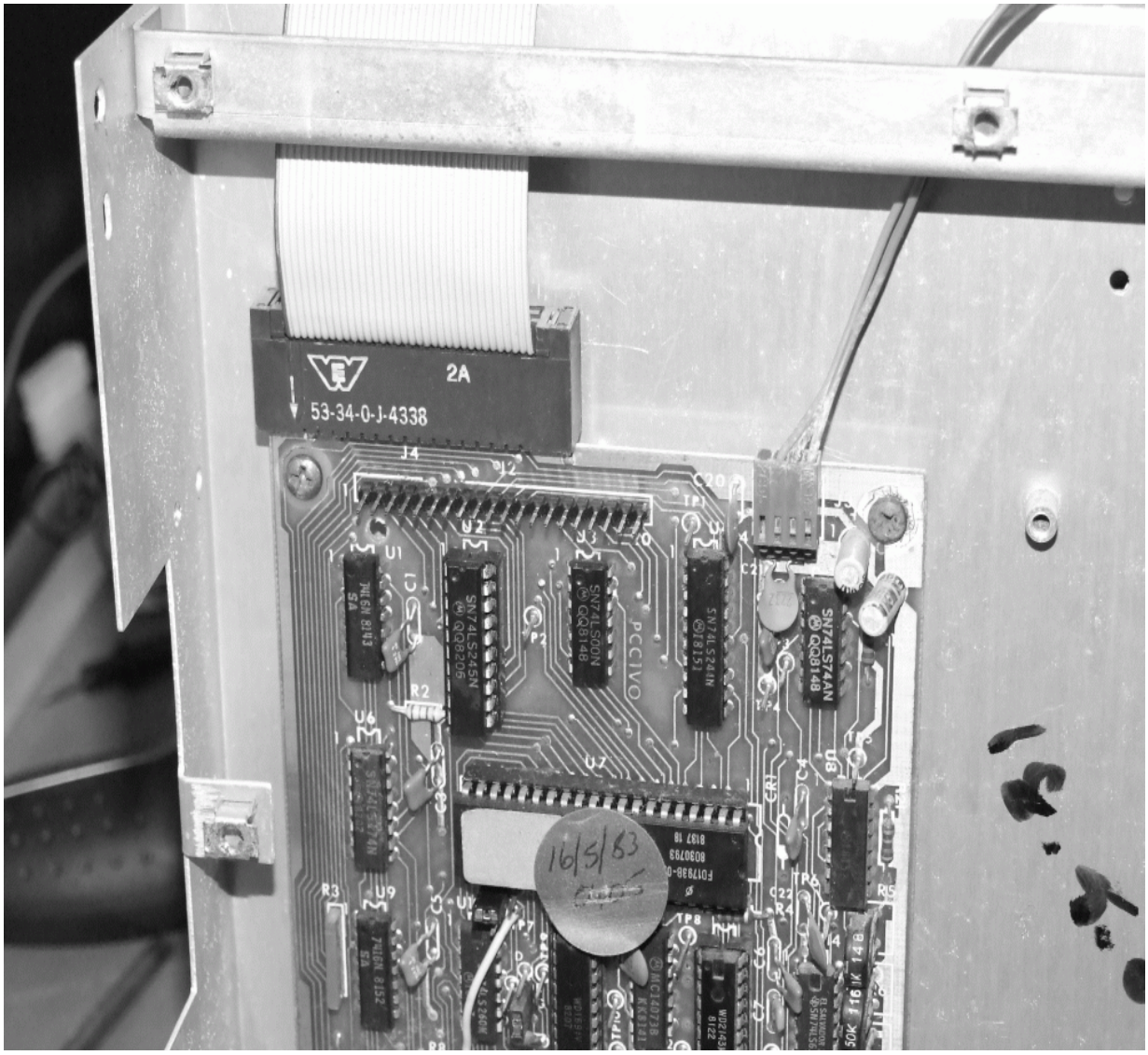
REFERENCES:

1. "BUILDING THE SUPER MICRO", COMPUTER USER, FEBRUARY 1984 PP. 14-16; MARCH 1984 PP. 14-18; APRIL 1984 PP. 36-38; MAY 1984 PP. 26-30.

2. C2 IS 0.1UF JAYCAR CAT. NO. RM7215 AND C1 AND C13 ARE 0.01UF JAYCAR CAT. NO. RM7065

3. HTTP://HOME.ONLINE.NO/~KR-LUND/REPAIR.HT

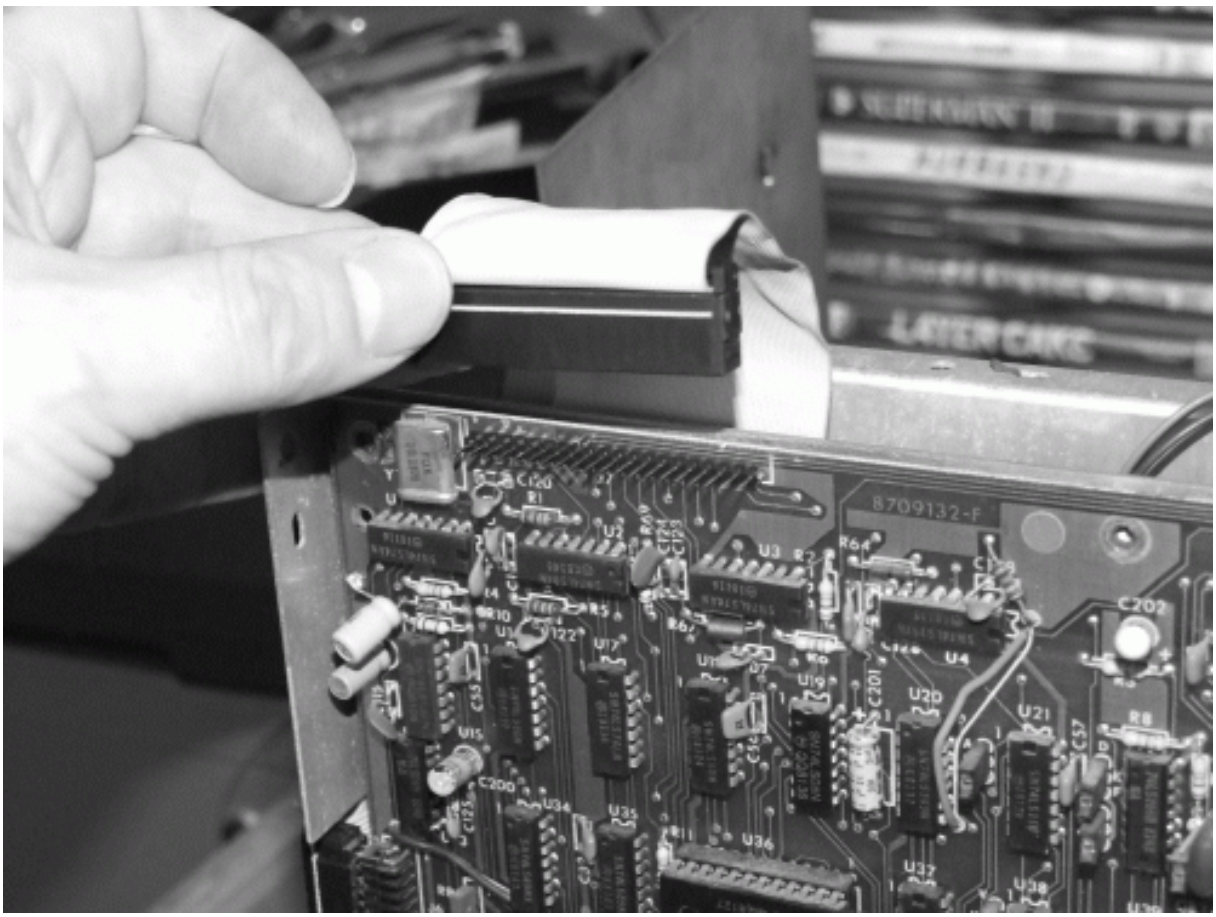
4. HTTP://WWW.JAYCAR.COM.AU/



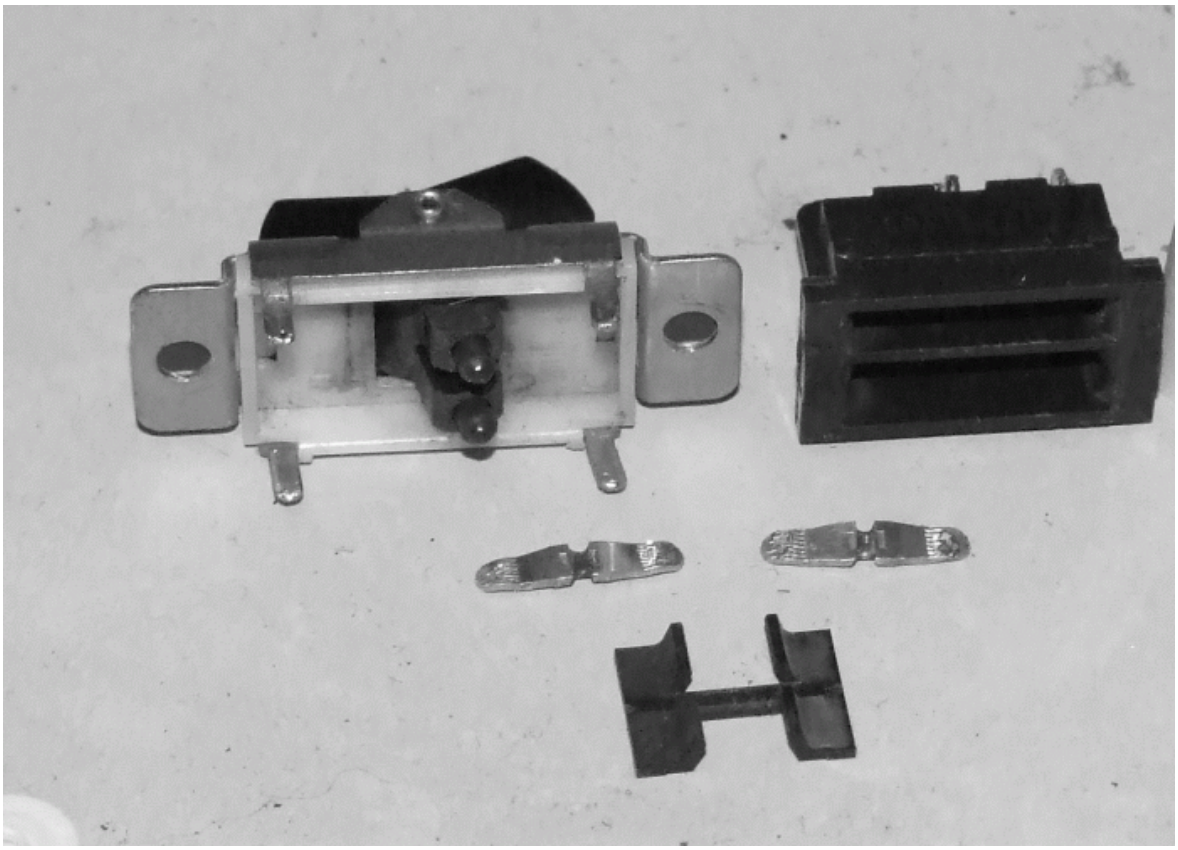
STRIP HEADER INSTALLED ON FDC CARD



ALL THE KEYS WITH TOPS REMOVED WERE NON-FUNCTIONAL



ABOUT TO PLUG IDE CABLE INTO STRIP HEADER ON MAINBOARD



DISASSEMBLED POWER SWITCH

ATTENTION!

RADIO SHACK FANS....

FIND THE COMPUTER EQUIPMENT TANDY NO LONGER SELLS!

-- COMPUTERS --

MODEL 1□ 16K + MON. + TAPE	$A499
MODEL 3□ 48K□ 2DR	$A495
MODEL 4□ 16K□ ØDR	$A195
MODEL 4□ 64K□ 2DR	$A45Ø
MODEL 4P□ 64K□ 2DR	$A495
TANDY 4D□ 64K□ 2DR	$A55Ø

-- ACCESSORIES --

DMP-1ØØ PRINTER	$A6Ø
M3 MOTHERBOARD□ 48K	$A1ØØ
M3/4 RS232 BOARD	$A5Ø
M3/4 FLOPPY DRIVE	$A75
M3/4 FLOPPY DISK CONTROLLER	$A75
M4 MOTHERBOARD□ 64K	$125

-- SOFTWARE --

BOOKS	FROM $A5
MAGAZINES	FROM $A1Ø

HARDWARE MANUALS (ORIG) FROM $A1Ø

AUS TRS-8Ø NEWSLETTERS CD $A1Ø

* ALL EQUIPMENT IS GUARANTEED TO BE IN GOOD WORKING ORDER
* EQUIPMENT IS CLEANED AND TESTED
* DRIVES ARE CLEANED AND TIMED AS NEEDED

TRS-8Ø UNIVERSE□ OF MELBOURNE□ AUSTRALIA.

HTTP://IANMAV.CUSTOMER.NETSPACE.NET.AU/TRS8Ø/

IANM@TRS-8Ø.COM

CRYPTOGRAPHY (Part 1)

By Peter Stone

While I was in the throws of writing my last article, making use of the back issues of TRS8BIT for some technical info I needed, I came across a comment written by Dusty that I'd promised to write an article on cryptography, specifically the ENIGMA machine. I then realised what my next article could be based on.

Before we get into the technical things, a few definitions need clarifying. The word cryptography comes from the Greek, & can be translated as 'the study of secret writing'. There are two types of secret writing, codes & ciphers. Simply put, a code is when words are replaced either by symbols, or other words. For example, the quick brown fox could be represented by *&?+ or by the word mountain. A cipher is when the individual letters in the words, are replaced by other letters according to some predetermined key. The problem with a code is that it requires the distribution of a book showing the correlation between the symbols & the words. Because of this codes are no longer used, & these articles will deal solely with ciphers.

The earliest forms of ciphers were transposition types, which simply rearranged the order of the letters. For example, hello world could become, orlellohwd. An early form of cipher known to everyone, is the Caesar cipher, also known as a monoalphabetic substitution cipher. Use of this cipher involves shifting the letters by a predetermined number so a random message seems to appear. A modern example of this is ROT13, used as a simple way to hide spoilers & other info on the net from casual reading. In ROT13 letters are simply moved by 13, so A becomes N & N becomes A etc.

Solving a Caesar cipher is easy, because all you would have to do is simply increment the letters by one, on a short piece of ciphertext, until recognisable text appeared & the cipher is broken. The weakness of a Caesar cipher was identified, & eventually overcome by the use of a keyword. A keyword was chosen, for example, PETER STONE, then duplicate letters & spaces are removed e.g. PETRSON then this would be followed by the remaining letters in a particular order, agreed beforehand by the users, such as; PETRSONABCDJKLMFGHIQUVWXYZ. This is then linked with an alphabet & the ciphertext produced.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
P	E	T	R	S	O	N	A	B	C	D	J	K	L	M	F	G	H	I	Q	U	V	W	X	Y	Z

In the above example, the plaintext is shown in lower case, & the ciphertext shown in uppercase. This is a convention that will be followed throughout this article. Using the above grid, the plaintext message 'the cat sat on the mat', would be enciphered by first removing all the spaces, so the plaintext becomes 'thecatsatonthemat', then be converted by reading the plaintext letter & taking the equivalent ciphertext letter below it. So the ciphertext becomes 'QASTPQIPQMLQASKPQ'. It must be admitted that using such a trivial arrangement of the cipher key provides enough clues to allow an attempt an decipherment. The above technique was used for a large period of time.



The next advancement in the field of cryptography, came from the Islamic part of the world. In the year 750, the golden age of Islamic civilisation began, which required administration, which in turn required secure communication achieved through the use of encryption. It is recorded that tax records were protected through cryptography. Further evidence comes from many administrative manuals, such as the tenth-century *Adab al-Kuttab* (*'The* Secretary's Manual'). If this had been the only contribution made, there wouldn't be much more to be said. However, this golden age produced the science of destroying ciphers, otherwise known as cryptanalysis, the science of unscrambling a message without knowing the key.

Although it's not known who first discovered frequency analysis, Islamic texts certainly made mention of the technique, & how it could be applied. Basically, the letter frequency of a language can easily be found by the simple analysis of a piece of text. A classic example of deciphering a cipher in this manner, is found in Sir Arthur Conan-Doyle's short Sherlock Holmes story, 'The Sign of the Dancing Men'.

The letter frequency commonly used in the English language is shown below.

The letter frequency in normal usage is;
e t a o i n s r h l d c u m f p g w y b v k x j q z
& the letter frequency in general fiction is;
e t a o h n i s r d l u w m c g f y p v k b j x z q

In truth, there will be some slight variation in the order of the letters, around the c u stage, but the above will serve for most decipherings. It was inevitable that the knowledge of how monoalphabetic substitution ciphers could be broken by frequency analysis would make its way to the western world. Mary Queen of Scots trusted a monoalphabetic substitution cipher to hide the plot to free her & kill Queen Elizabeth, the letters were intercepted & the cipher cracked leading to her trial & execution in 1586.

Vigenère cipher

What is now known as the Vigenère cipher was originally developed by Leon Battista Alberti. Sometime in the 1460s, he became involved in conversation with his friend Leonardo Dato on the finer points of cryptography. the result of this conversation led to the publication of a treatise on a new form of cryptography. Alberti proposed using two or more cipher alphabets, switching between them to encrypt the message & confuse potential cryptanalysts.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	Z	B	V	K	I	X	A	Y	M	E	P	L	S	D	H	J	O	R	G	N	Q	C	U	T	W
G	O	X	B	F	W	T	H	Q	I	L	A	P	Z	J	D	E	S	V	Y	C	R	K	U	H	N

For example, two possible cipher alphabets are shown above, (in uppercase), with the plaintext above them. Let us take our plaintext message 'the cat sat on the mat', which we saw was trivial to solve, & encrypt it using the new cipher, by taking the ciphertext from each of the alphabets alternately. Again, we remove all the spaces, 'thecatsatonthemat' & the encrypted message becomes GHKBGGVFYDZGHKPFY. Although my choice of plain text does give some clues which would help an experienced cryptanalyst to decrypt the message, GHK being the most obvious, the fact that 'cat' encrypts to BGG does increase the strength of the cipher. Although he'd hit on the most significant breakthrough in encryption for over a thousand years, he failed to follow it through to a fully formed system of encryption. Others built on Alberti's work, until Blaise de Vigenère examined the work of all the others, & built it into its final form.

Plaintext

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

The strength of the Vigenère cipher lies in its use of not one, but 26 distinct cipher alphabets to encrypt the message, with each cipher alphabet shifted by one letter with respect to the previous alphabet. As shown in the table above. If only one row was used to encrypt a message, then this would be a simple Caesar cipher & relatively easy to solve. The strength is that different rows are used to encrypt different letters of the message, providing no clues to the help the cryptanalyst. To decipher the message, the recipient needs to know which rows have been used to encrypt the message, this is done by using a keyword.

The first step in encipherment is to construct the above table. The next step is to choose a keyword, & the final step is to encrypt the message. as an example let's take our original message, 'the cat sat on the mat' & encrypt it with the keyword ROBIN. We start encryption by writing down the plaintext message with the keyword repeated underneath it as many times as is required, as shown below.

T	H	E	C	A	T	S	A	T	O	N	T	H	E	M	A	T
R	O	B	I	N	R	O	B	I	N	R	O	B	I	N	R	O
K	V	F	K	N	K	G	B	B	B	E	H	I	M	Z	R	H

The next step is to find the relevant letter in the plaintext row, then follow down that column down until it intersects the row corresponding with the letter of the keyword. For example, if we go down the column for the plaintext letter t until we intersect the row for R, we get the letter K. Going through the plaintext, the final enciphered word is KVFKNKGBBBEHIMZRH, a much stronger cipher, as the first 'the' enciphers to KVF & the second enciphers to HIM. The sequence 'ato' enciphers to BBB, & the sequences of 'at' encipher to NK, BB & RH respectively. This prevents any attempt at frequency analysis. Although Vigenère published his treatise *Traicté de chiffes* in 1586, it was ignored & not adopted until 200 years later. It is an intriguing thought, but if Mary, Queen of Scots had read his work & implemented his cipher, then history may have taken a much different course. For many years, the Vigenère cipher was thought to be unbreakable, which led to it being called, le chiffre indéchiffrable (French for 'the unbreakable cipher').

In 1863 Friedrick Wilhelm Kasiski published a method of breaking Vigenère's cipher, & for years was thought to be the first to break it. However in the 20th century, researchers going through the notes of Charles Babbage, found that he had cracked Vigenère's cipher in 1854. Why Babbage didn't publish his work may never be answered. However, Simon Singh in his book offers a tantalising theory, that Babbage was asked not to publish, because at that time Britain was fighting the Crimean war, & Vigenère's cipher was in regular use, so by suppressing the knowledge that the cipher was broken, Britain gained an advantage.

How Babbage came to break the cipher is an interesting story. In 1854 a Bristol dentist named John Hall Brock Thwates, an enthusiastic amateur cryptologist, wrote to the *Journal of the Society of Arts,* describing a new cipher he'd come up with. Babbage wrote back saying that it wasn't new & was found in the textbooks. This annoyed Thwates, who challenged Babbage to crack it. Without going into a six page detailed explanation, which can be found in Simon Singh's book, Babbage was sent a cipher consisting of 367 letters. He examined the the text, & realized that if he could calculate the length of the keyword, then frequency analysis could be applied to crack the cipher. Babbage examined the ciphertext & found repetitions within the text. I've copied the table of repetitions out below

REPEATED SEQUENCE	REPEAT SPACING	POSSIBLE LENGTH OF KEY (OR FACTORS)																	
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
EFIQ	95				X														X	
PSDLP	5				X															
WCXYM	20	X		X	X					X										X
ETRL	120	X	X	X	X	X		X		X		X			X					X

It can be seen from the table that the only key length that fits is 5. Therefore, if we split the ciphertext into 5 separate columns, they become 5 Ceaser ciphers, & the letter frequencies can be calculated, which is what Babbage did. He next analysed a piece of text of the same length, & again did the frequency analysis. If the peaks & troughs of the control text, is compared to the peaks & troughs of the individual Ceaser ciphers, then the shift can be calculated & the keyword worked out. It is worth reading Simon Singh's book to see how the it was solved in detail.

If you use the Vigenère grid above to encrypt & decrypt messages, the first thing that you notice is how intensive the work is to do. This is a task that is eminently suited to using a computer to carry out.

Below is a short program that will encipher & decipher a Vigenère cipher.

```
 10 'Vigenere Cipher
 20 CLEAR 1000
 30 INPUT "Enter Keyword : ";W$
 40 LW=LEN(W$)
 50 DIM W(LW)
 60 FOR I=1 TO LW:W(I)= ASC (MID$(W$,I,1)) - 65:NEXT
100 INPUT "<E>ncrypt or <D>ecrypt : ";E$
110 E=+1:IF E$="D" THEN E=-1
120 INPUT "Enter Plaintext or Ciphertext, letters only ";C$
140 P$=""
150 FOR I =1 TO LEN(C$): X=ASC (MID$ (C$,I,1))-65
160 Y=(I-1)-LW*INT((I-1)/LW)
```

```
170 X=X+E*W(Y+1)
180 IF X>25 THEN X=X-26
190 IF X<0 THEN X=X+26
200 P$=P$+CHR$(X+65)
210 NEXT I
220 PRINT P$
230 END
```

It does this by assigning the numbers 0 to 25 to the letters A to Z. Modulo arithmetic, (mod 26) is carried out, & the result obtained. In fact, modulo arithmetic is frequently used in cryptography. To express a number in Mod 26, say, we add or subtract 26 to or from it until the result is in the range 1 to 25. This is equivalent to dividing by 26, discarding the quotient & keeping the remainder.

Splitting the program into blocks & analysing it we get;

```
  20 CLEAR 1000
```

Line 20 allocates 1000 bytes of memory for use by string variables in the program. This is necessary as by default the TRS80 only allocates 50 bytes.

```
  30 INPUT "Enter Keyword : ";W$
  40 LW=LEN(W$)
  50 DIM W(LW)
  60 FOR I=1 TO LW:W(I)= ASC (MID$(W$,I,1)) - 65:NEXT
```

Line 30 asks you to input the keyword, which should be in uppercase.
Line 40 calculates the length of the keyword.
Line 50 dimensions a numerical array W to the length of the keyword.
Line 60 reads each character in the keyword, converts it to its numerical value & then subtracts 65, which converts it to a number between 0 & 25.

```
100 INPUT "<E>ncrypt or <D>ecrypt : ";E$
110 E=+1:IF E$="D" THEN E=-1
120 INPUT "Enter Plaintext or Ciphertext, letters only ";C$
140 P$=""
```

Line 100 simply asks you to select either E for encrypt, or D for decrypt.
Line 110 by default is set to +1, however, if we select D, then E is set to -1.
Line 120 is where we enter our plaintext or ciphertext depending on whether we are encrypting or decrypting, & assigns it to variable C$.

Line 140 sets variable P$ to null.

```
150 FOR I =1 TO LEN(C$): X=ASC (MID$ (C$,I,1))-65
160 Y=(I-1)-LW*INT((I-1)/LW)
170 X=X+E*W(Y+1)
180 IF X>25 THEN X=X-26
190 IF X<0 THEN X=X+26
200 P$=P$+CHR$(X+65)
210 NEXT I
220 PRINT P$
```

Line 150 carries out the same function as line 60, converting them to the range 0<X>25, but for the letters in the plain/cipher-text.
Line 160 carries out the X mod LW function. For a key length of 5, it generates values of 0, 1, 2, 3, 4
Line 170 takes the value from the line above, adds 1 to it, & then adds the numerical content of the current location in the plain/cipher text string, to the numerical value at the location pointed to in the keyword.
Line 180 checks if the result of the previous calculation is greater than 25. If so, then 26 is subtracted to bring it within the range 0<X>25.
Line 190 does the same check, but to see if the result is less than 25. If so 26 is added to bring it within the range 0<X>25.
Line 200 takes the result of the previous two lines, adds 65 to it to convert it to an ASCII value, then stores the resultant character to P$.
Line 210 is the end of the FOR - NEXT loop & increments the pointer I by 1.
Line 220 prints out either the ciphertext or the plaintext contained in P$ after all the text has been converted.

```
230 END
```

Line 230 just provides a way of ending the program upon completion.

The program could be modified in several ways. The first would be to include statements to read or write the plain or encrypted text to files, for easy storage & accuracy. Another would be to allow the input of text & the keyword in upper or lowercase.

There are some modifications which could be made to the program statements.

The mod function in Line 160, could be replaced with the following line at the start of the program.

....DEF FN M(A,B)= A-B*INT(A/B)

Then use Z=FN M(X,Y) whenever Z=X mod Y is needed. I decided not to do this, as DEF FN is only present in disk basic.

In truth, this program on a TRS80 isn't really as useful as it could be, as coping the files on & off the virtual floppy disks used by the emulator, adds an unnecessary step or two. It could perhaps, be better programmed in GWBASIC, on a MSDOS based computer, running on a USB thumb drive, (which I've done), or under Visual BASIC et all on a Windows machine.

You may wonder why, if the Vigenère cipher has been broken, I've written at length about it. This is because the Vigenère cipher led to a device that I'll be writing about in the next instalment, the Enigma machine.

BIBLIOGORAPHY

The Code Book by Simon Singh.
A good general text on codes & ciphers. My copy is a hardback version, published in 2000. but still available in paperback.

Codes, ciphers, secrets & cryptic communication by Fred B. Wrixon, Published by Black Dog & Leventhal, 2005. ISBN 1-57912-2
A comprehensive & interesting book on all forms of cryptic communication, including some obscure ones.

The Radio Hackers Code Book by George Sassoon. Published by Duckworth, 1986. ISBN 0 7156 2068 1
Although dated, this book contains programs & circuit diagrams for using a computer to encode & decode messages. It is well worth hunting for a copy though.

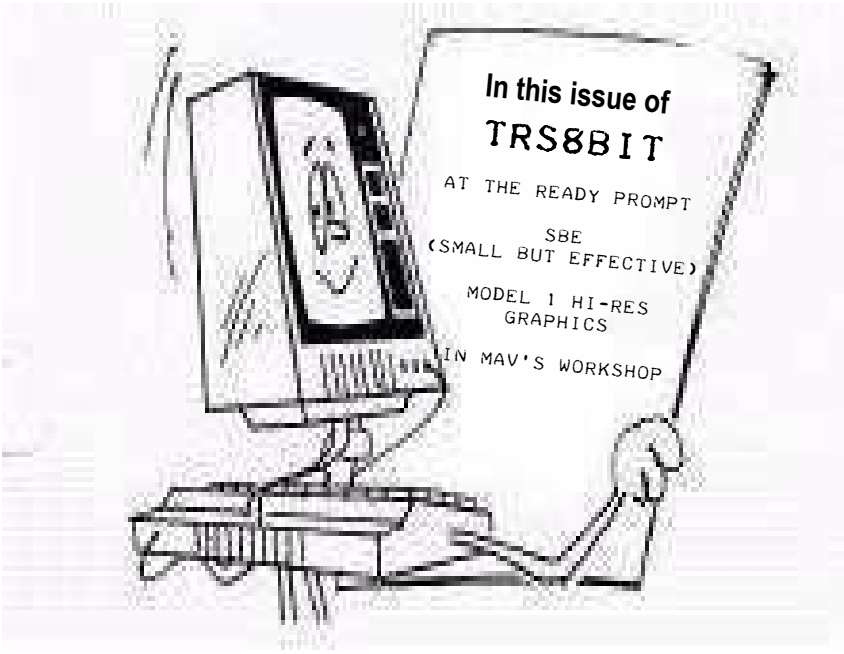
If you have any queries about this article, or want to suggest topics for future articles, then feel free to email me at peter_stone45@hotmail.com

TRS8BIT



HI THERE EVERYONE AND WELCOME TO THE START OF OUR 6TH YEAR WITH THE MARCH 2012 EDITION.
IT'S NOT JUST SPRING OUT THERE, THERE'S A SPRING IN MY STEP AS I CAN REPORT THAT FOR THE FIRST TIME, WE'VE HAD OVER 100 DOWNLOADS OF THE LAST ISSUE. (MIND YOU IT WAS THE XMAS EDITION EH!).
THANKS, EVERYONE FOR YOUR CONTINUED SUPPORT AND INTEREST.

RIGHT! LETS GET DOWN TO BUSINESS.
JUST TO KEEP EVERYONE UP TO DATE, HERE'S NEWS OF SOME RECENT SOFTWARE UPDATES.
MIGUEL DUTRA HAS RELEASED V1.3 OF HIS VIRTUAL DISK KIT.
VERNON B HESTER HAS RELEASED VARIOUS MULTIDOS UPGRADES AND ENHANCEMENTS.
DAVE HUFFMAN HAS RELEASED V3.6X OF HIS RAMDISK PROGRAM.
AND FINALLY, MATTHEW REED HAS PRODUCED AND JUST RELEASED A Z80 ASSEMBLER FOR A WINDOWS PC, TOGETHER WITH UPDATES FOR TRSREAD AND TRSWRITE.
ALL OF THE ABOVE SOFTWARE IS AVAILABLE FOR DOWNLOADING, EITHER FROM IRA'S WEBSITE OR FROM THE LINKS ON OUR SITE. SO PLEASE, IF YOU GET A CHANCE, SUPPORT AND ENCOURAGE THESE GUYS; GET YOURSELF AN UPGRADE.
I'VE ACTUALLY ORDERED A COPY OF MATTHEW'S PC Z80 ASSEMBLER, EVEN THOUGH IT REQUIRES, AS A MINIMUM, WINDOWS XP. (I HOPED THIS MIGHT BE THE DECIDING FACTOR AND GIVE ME THAT FINAL, EXTRA PUSH IN GETTING ME MOTIVATED TO UPGRADE FROM WINDOWS ME!).



THERE'S NOT BEEN THAT MUCH ACTIVITY ON EBAY (UK) OVER THE LAST FEW MONTHS. THE ONLY THINGS THAT CAUGHT MY EYE WERE A M102 THAT FETCHED JUST UNDER £50 AND A VIDEO GENIE WHICH WITH POSTAGE, TOOK THE GREATER PART OF £100.
EBAY.COM IS AS BUSY AS EVER THOUGH, TANDY'S IN THE USA SEEM TO BE BOOMING!
THINGS ALSO SEEM TO BE ON THE UP IN AUSTRALIA TOO, IF YOU DON'T BELIEVE ME, JUST CHECK-OUT MAV'S WEBSITE AND SEE SOME OF THE FANTASTIC STUFF HE'S OFFERING FOR SALE.

I'VE HAD NO CONTACTS FROM ANYONE IN THE UK REGARDING REPAIRS ETC ON TRS-80'S. I'VE SET UP AN

(CONTINUED ON PAGE 15)

AT THE
READY>
PROMPT



HERE'S A NEAT LITTLE TRICK I WAS REMINDED OF QUITE RECENTLY. THE CMD"O" SORT IN NEWDOS8Ø CAN BE USED IN IMMEDIATE MODE. BY PRESSING THE BREAK KEY WHILE IN BASIC YOU CAN SORT AN ARRAY OF DATA AND PRINT ITS CONTENTS BEFORE RETURNING BACK TO YOUR PROGRAM. A SLIGHT PROBLEM CAN ARISE IF YOU ENCOUNTER A "SYNTAX ERROR" BUT IT IS POSSIBLE TO GET BACK TO THE PROGRAM WITH VARIABLES INTACT, BY ISSUING A "GOTO NN". (IN MY CASE, THIS WOULD BE THE LINE NUMBER OF THE PROGRAM'S MAIN MENU SCREEN). NOW I'M SURE THAT THIS WOULD BE LOOK UPON AS VERY BAD PRACTICE, BUT IN AN EMERGENCY, THIS COULD BE A LIFE SAVER!

NOW, HERE'S THE TIME WHEN I TEACH MY GRAND MOTHER TO SUCK EGGS SO TO SPEAK. WELL MAYBE NOT AS IT'S ALWAYS HANDY TO BE REMINDED. ROUNDING, THE USUAL 'PIA', BUT SHOULD YOU WISH TO GIVE 'B' THE VALUE OF 'A' ROUNDED TO 'D' DECIMAL PLACES, HAVE A GO AT THE FOLLOWING

```
B=INT(A*10 ] D+.5)/10 ] D
```

NEAT EH? THE SQUARE BRACKET] IS 'TO THE POWER OF' (THERE'S NO UP-ARROW WITH THE TELETYPE FONT! SORRY!)

--== oØo ==--

SPORTING FORECASTS

Professor Frank George's well-known Football Pools Forecasting program is now available on the

SINCLAIR ZX81 16K

as well as:

APPLE II 32K
PET 32K
SHARP MZ80K 36K

Versions coming soon for:
Video Genie, BBC-micro, TI 99/4

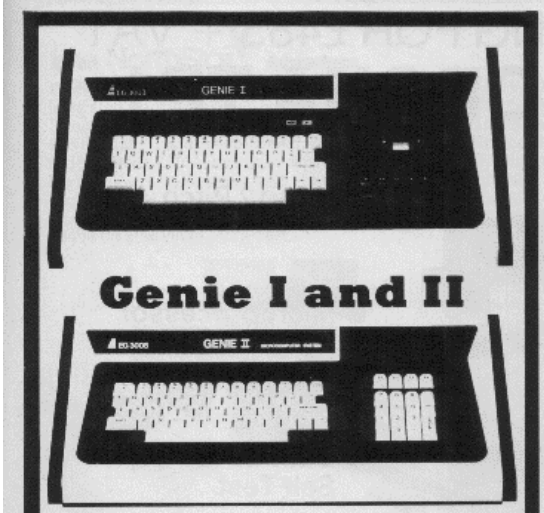
A Horse-Race Forecast Program in preparation.

Write to: Professor F.H. George Bureau of Information Science Commerce House, High Street, Chalfont St. Giles, Bucks.

CONSULTANT ELECTRONICS

VIDEO GENIE COMPUTERS
Books — C10 Cassettes — Disks
Your own 'Basic' Genie / TRS80 Programs listed.
Vic 20, ZX81, Genie, TRS80 Software.

59 Mason Road, Erdington, Birmingham. Tel 021-382 7247



GENIE 1	16K	£279.00 + VAT
GENIE 11	16K	£299.00 + VAT
COLOUR GENIE	16K	£173.00 + VAT
	32K	£214.00 + VAT

3 MONTH WARRANTY
For further information ring (084 421) 5369

12 Jefferson Way
Thame Industrial Estate
Thame, Oxfordshire
OX9 3UP

pebbleglow limited

24 hr service on 5369

SBE - A SIMPLE BUT EFFECTIVE MACHINE CODE GENERATOR

ARIELA & GEOFFREY TAYLOR

SBE COMES FROM THE SAME SOURCE AS ACCEL, EDIT AND ENB WHICH ARE GENERALLY REGARDED AS AMONG THE BETTER SERIOUS PROGRAMMES FOR THE TRS 80. IT IS A SYSTEM WHICH SIMPLIFIES AND SPEEDS UP MACHINE CODE PROGRAMMING BY ALLOWING YOU TO WRITE, SAVE, TEST AND DEBUG IN ALMOST NORMAL BASIC AND PRODUCE RELOCATABLE /CMD/OBJ/REL OR /CIM FILES - EDIT AND END WERE WRITTEN WITH SBE.

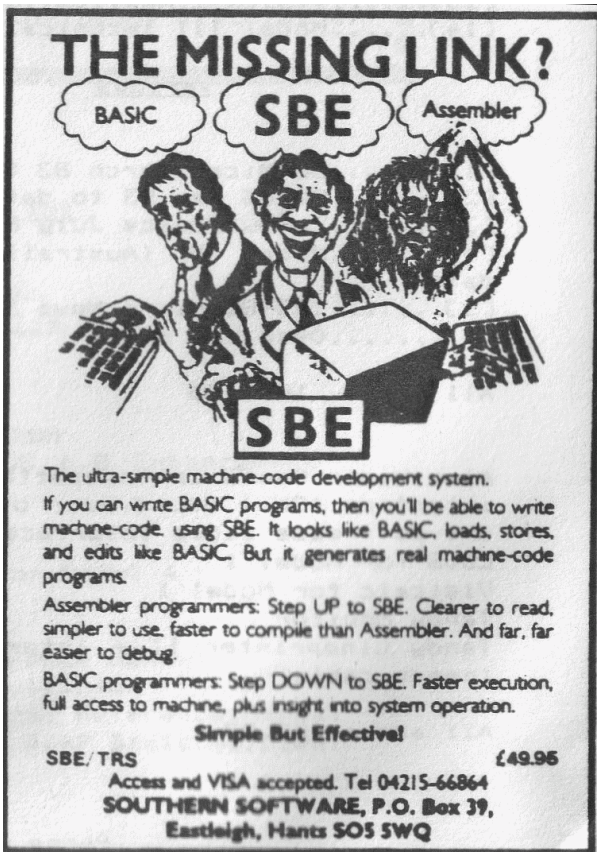
SBE GENERATES MACHINE CODE INSTRUCTIONS DIRECT FROM YOUR PROGRAMME CODE SO TRS-80/SBE PROGRAMMES WILL RUN ON ANY Z80 SYSTEM, NOT JUST TRS-80. AN IBM PC VERSION OF SBE IS ALSO AVAILABLE GENERATING CODE FOR 8066/8088, WHICH MEANS THAT TRS-80/SBE PROGRAMMES CAN BE TRANSFERRED TO THE IBM PC, ALTHOUGH SYSTEM SPECIFIC CALLS OBVIOUSLY MUST BE CHANGED.

SBE SHOULD RUN WITH ALL THE USUAL TRS-80 DOS'S AND WAS HAPPY IN BOTH 64 AND 80 CHARACTER NEWDOS 80(2) ON OUR VIDEO GENIE. JUST TYPE SBE <ENTER> AND IT LOADS, PROTECTS ITSELF AND LEAVES YOU IN BASIC WITH NORMAL EDITING, LOAD AND SAVE. WHEN READY TO TRY OUT WHAT YOU'VE WRITTEN THE COMMAND /CLS COMPILES THE PROGRAMME. SAVE IT FIRST THOUGH! ALTHOUGH SYNTAX ERRORS AND MISSING DEFINITIONS ARE TRAPPED AND CAN BE CORRECTED IMMEDIATELY, OTHER ERRORS CAN HAVE THE SAME EXASPERATING RESULTS AS IN ASSEMBLER - AND NEWDOS80'S BASIC * DOES NOT WORK IN SBE.

AFTER COMPILING, /RUN RUNS YOUR PROGRAMME. TRON/TROFF COMMANDS ARE USED TO SINGLE STEP ALL OR PART OF IT. STEPPING IS BY PROGRAMME LINE AND THE LINE TO BE EXECUTED, REGISTERS, PROGRAMME COUNTER AND STACK POINTER ARE DISPLAYED AT EACH PAUSE POINT. WHILE PAUSED YOU CAN EXAMINE NAMED VARIABLES AND MEMORY LOCATIONS, CALL A SUBROUTINE OR ENTER A NEW LINE NUMBER TO RESTART SINGLE STEPPING FROM. ONCE THE PROGRAMME RUNS PROPERLY RETURN TO DOS AND TYPE CUT NAME/EXT XNNNN (NNNN IS PRINTED AT THE END OF EACH COMPILATION) TO CREATE THE /CMD ETC FILE. THIS IS NOW A STAND-ALONE RELOCATABLE MACHINE CODE PROGRAMME WHICH DOES NOT NEED SBE TO RUN. VERY LARGE PROGRAMMES NEED TO BE WRITTEN IN MODULES OF LESS THAN 10K AND LINKED TOGETHER, A SIMPLE PROCESS WITH THE LINKING UTILITY DEALING WITH ALL CROSS-REFERENCES.

SBE PROGRAMMES LOOK LIKE BASIC, WITH ASSEMBLER OVERTONES AND SOME SYNTAX WHICH IS SPECIFIC TO SBE - (SEE EXAMPLES). NAMED VARIABLES CAN BE USED AS IN BASIC, THEY CAN BE LONGER THAN 2 CHARACTERS BUT MUST BE INDIVIDUALLY INITIALISED. VALUES CAN BE ADDRESSES, INCLUDING ROM CALLS E.G. DEF DISPLAY:&H033A ENSURES THAT THE ROM ROUTINE AT 033A IS CALLED BY THE WORD DISPLAY WHENEVER IT OCCURS IN THE PROGRAMME. IN ADDITION THE REGISTERS (AND REGISTER

PAIRS) A B C D E H AND L CAN BE TREATED AS VARIABLES ALTHOUGH WITH THE LIMITATIONS IMPOSED BY THE Z80 INSTRUCTION SET. STACK MANAGEMENT IS YOUR RESPONSIBILITY, AS IN ASSEMBLER BUT AS REGISTERS AND STACK POINTER ARE DISPLAYED WHEN IN TRACE MODE THIS IS NOT TOO DIFFICULT EVEN FOR THOSE WITH NO ASSEMBLER EXPERIENCE. IF/THEN/ELSE, FOR/NEXT LOOPS AND GOSUB AND GOTOS ARE SIMILAR TO BASIC BUT USE LABELS INSTEAD OF LINE NUMBERS. SBE'S CMD STATEMENT ALLOWS TRANSFER OF INFORMATION FROM THE CALL TO THE SUBROUTINE E.G. IN LINES 500-530 OF EXAMPLE 1 THE ASCII CODES IN BRACKETS ARE PASSED TO THE SUBROUTINE IN LINE 620 WHICH SENDS THEM IN TURN TO THE PRINTER.



SBE DOES NOT ATTEMPT TO GENERATE ALL Z80 INSTRUCTIONS DIRECTLY. ASSIGNMENTS (A=10,DE=(HL)), MODIFICATION (HL=1,CP+64), AND COMPARISON (IF HL>DE, IF A="0") ARE DIRECTLY SUPPORTED. OTHER FUNCTIONS LIKE ROTATES AND BLOCK MOVES ARE USED BY SIMPLY PUTTING IN THE HEX CODES, EITHER DIRECTLY INTO A PROGRAMME LINE OR AS A MACRO (MACRO IS USED IN SAME WAY AS IN DBASE AND NOT AS IN ASSEMBLER- I.E. AS A SHORTHAND WHICH IS EXPANDED AT COMPILATION - SEE EXAMPLE 2). THIS SYSTEM MAKES IT EASY TO TRANSLATE ORDINARY ASSEMBLER LISTINGS INTO SBE. EXAMPLE 2 IS A SCREEN FLASH ROUTINE EXTRACTED FROM A LONG GAME LISTING IN HARDIN BROTHERS' LAST ARTICLE ON ASSEMBLER IN 80-MICRO. EXAMPLE 1 IS DERIVED FROM MICHAEL TOMLINSON'S PRINTER SETTING PROGRAMME IN JUNE NATGUG PLUS A SIMILAR ONE IN AN OLD 80-US.

DOCUMENTATION FOR THE TRS-80 VERSION OF SBE CONSISTS OF A 62 PAGE MANUAL AND A Z80 INSTRUCTION SUMMARY BOOKLET. ABOUT HALF THE MANUAL DEALS WITH THE LANGUAGE AND HOW TO USE IT; THE REST INCLUDES DESCRIPTIONS OF Z80 CHIP OPERATIONS AND STACK MANIPULATIONS, ANNOTATED PROGRAMMES ILLUSTRATING BASIC USR ROUTINES, FILE PROCESSING AND CREATING NEW BASIC 'COMMANDS', A LONG LIST OF ROM CALLS AND USEFUL ADDRESSES AND A LIST OF THE MOST OFTEN USED MACROS. THE MANUAL DOESN'T QUITE LIVE UP TO THE SIMPLE BIT OF SBES NAME AND MAKES FEW CONCESSIONS TO ASSEMBLER IGNORAMI BUT THERE IS A LONG

TUTORIAL FILE ON THE DISK WHICH SHOULD BRIDGE THE GAP AS WELL AS PROVIDING A SMALL LIBRARY OF USEFUL FUNCTIONS LIKE MESSAGE AND GRAPHICS DISPLAY, MATHS, KEYBOARD INPUT AND FILE I/O.

SBE FOR THE TRS / VIDEO GENIES COSTS ALMOST £50, NOT A SUM TO BE SPENT LIGHTLY, SO IS SBE REALLY SIMPLE BUT EFFECTIVE AND IS IT WORTH BUYING?. OUR OPINION IS DEFINITELY 'YES'. ONE OF US HAD ONLY PROGRAMMED IN BASIC AND DBASE II BEFORE TRYING SBE BUT WROTE A USEFUL SMALL MACHINE CODE UTILITY THE FIRST EVENING. THE OTHER WAS ALREADY FAMILIAR WITH ASSEMBLER AND FOUND SBE TO BE QUICKER AND EASIER THAN HIS USUAL TOOLS OF ZEN AND TASMON. IT REALLY DOES MAKE WRITING MACHINE CODE ALMOST AS EASY AS BASIC AND WE HAVEN'T FOUND ANY DISADVANTAGES SO FAR. WE'VE NOT TRIED PROGRAMME MIGRATION YET BUT WOULD BE SURPRISED IF THIS DIDN'T WORK AS CLAIMED SO MAYBE WE'LL BE ABLE TO USE OUR FAVOURITE PROGRAMMES ON OUR NEXT COMPUTER TOO.

--=== oØo ===--

NATGUG

Users of Tandy & Genie microcomputers supported

★ Meetings on Basic, M.Code, W.P. Model 16, Model 100, DOS
★ Vast range of expertise
★ Free software library
★ Local groups
★ Regularly manned 'phone
★ Back issues of newsletters and specialist journals

Write: NATGUG, 24 Oxford Street, Stony Stratford, Milton Keynes
Tel: Brian Pain on 0908 564271

TRS-80
VIDEO GENIE

NEWDOS-80
APL-80
PASCAL-80
FORTH (MMS)

Details of these and over 200 other programs are contained in our new loose leaf catalogue price £1.00 (refundable) from:-

MICROCOMPUTER APPLICATIONS
41 QUEEN'S ROAD
BLANDFORD FORUM
DORSET DT11 7LA
TEL: (0258) 55100

TANDY MODEL 100 USERS CLUB

Newsletter planned and range of software

SAE for details

REMSOFT
18 George Street
Brighton BN2 1RH
Tel: 0273 602354

TRS - 80 ★ BBC ★ GENIE

B/L britannia software library

Worried about a software purchase? Join B/L and "Try before you buy" by hiring **BOTH** games **AND** business software from only £1.25 for 14 days!
Interested? Send a **S.A.E. TODAY** for our **FREE** information pack!

B/L britannia software library
9 SOUTH BEACH ROAD
ARDROSSAN KA22 8AX

MODEL 1 HIRES GRAPHICS

JOHN KILPATRICK

MAIN CHARACTER GENERATOR.

EACH HI-RES GRAPHICS CHARACTER IS MADE UP OF A MATRIX 6 PIXELS ACROSS AND 12 LINES DOWN. THIS CHARACTER IS THE SAME 'SIZE' AS THE ORIGINAL TANDY CHARACTER NO. 191 BUT INSTEAD OF BEING SUB-DIVIDED INTO 2X3 IT IS NOW 6X12 EACH OF THE 72 DOTS OF A CHARACTER HAS L BIT OF RAM ALLOCATED TO IT. TO ACHIEVE THIS EACH LINE OF 6 DOTS IS ALLOCATED 1 BYTE OF RAM WITH 2 BITS NOT USED.
BIT 0 IS THE RIGHTMOST DOT AND BIT 6 THE LEFTMOST, WITH BITS 7 AND 8 NOT USED. AS THE LINE COUNT FOR EACH CHARACTER IS Ø-11 LINES 12 TO 13 ARE NEVER ACCESSED, GATING OF THE ADDRESS LINES COULD HAVE BEEN EMPLOYED HERE TO USE THE MINIMUM AMOUNT OF RAM, BUT AS YET MANUFACTURERS OF 1536X8 (128X12) RAMS ARE HARD TO FIND, SO 2048X8 (128X16) HAVE BEEN USED AND 4 BYTES WASTED ON EACH CHARACTER.

THE CHARACTER GENERATOR SECTION IS MADE UP 4KX4 LOW POWER RAMS (2114S), I/CS 7 TO 1Ø, I/C 11 GATE AND HEX TRISTATE GATE I/C 6. THE RAMS ARE CONFIGURED INTO 2 1KX8 BLOCKS, WITH NA 1Ø AND NA 1Ø (VIA I/C 11 INVERTER) ACTING AS THE BLOCK SELECTOR. I/C 6 ACTS AS A BUFFER BETWEEN THE CPU DATA BUS AND THE CHARACTER GENERATOR OUTPUT.

IN OPERATION THERE ARE SIMPLY TWO MODES, FIRST, READ, WHEN THE CHARACTER GENERATOR IS SUPPLYING VIDEO CHARACTER LINE INFORMATION TO THE VIDEO SHIFT REGISTER, AND SECOND, WRITE, WHEREBY THE CHARACTER GENERATOR IS BEING WRITTEN TO BY THE CPU. (SIMILAR TO THE WRITING OF INFORMATION TO THE 1K, OF VIDEO RAM).
IN THE READ MODE THE CONTROL LINE TO THE ADDRESS MULTIPLEXERS (I/C 3, 4 AND 5) IS LOW, THIS MEANS THAT THE RAM R/W IS HIGH, SETTING THE RAMS TO THE READ MODE AND I/C 6 BUFFER OUTPUTS TO THE TRISTATE MODE, IT ALSO MEANS THAT THE CHARACTER GENERATOR IS GETTING IT'S ADDRESS INPUTS FROM THE VIDEO TIMING CIRCUITS OF THE KEYBOARD (LØ, LBI ETC). THUS THE RAM OUTPUTS ARE PROVIDING SIGNALS FOR THE HI RES VIDEO SHIFT REGISTER (I/C 12).
IN THE WRITE MODE THE CONTROL LINE TO THE ADDRESS MULTIPLEXER IS HIGH. THE RAM R/W LINE IS NOW A FUNCTION OF THE Ø TO 2K DECODING AND THE CPU WR LINE. TO ACTUALLY WRITE A BYTE TO THE GENERATOR THE CPU MUST DO A WRITE OPERATION TO ANY LOCATION BETWEEN Ø AND 2Ø47 WITH THE ADDRESS MULTIPLEXER CONTROL LINE IN THIS HIGH STATE. ADDRESS Ø - 11 CORRESPOND TO LINES 1 - 12 OF CHARACTER 128, 16 - 27 TO CHARACTER 129, 32 - 43 TO CHARACTER 13Ø ETC, THE LINE DOT PATTERN AS DESCRIBED EARLIER, BUT TO INDICATE IT, Ø WILL GIVE A BLANK LINE, 63 A COMPLETE LINE, 1 A RIGHTMOST DOT AND 32 A L OF LEFTMOST ONE.
REMEMBER THOUGH, AS YOUR CHARACTERS ARE NOW RAM BASED, ON POWER-UP THEY WILL BE FILLED WITH 'JUNK'. TO CLEAR ALL THE CHARACTERS

RUN THE FOLLOWING PROGRAM:-

```
10 OUT 140,1:FOR
X=1T02047:POKE X,0:NEXT:OUT
130,1
```

OR IN ASSEMBLER FOR SPEED:-

```
LD HL, 0
LD (HL),0
PUSH HL
POP DE
INC HL
LD BC,2047
OUT (140),A
LDIR
OUT (130),A
```

ADDRESS DECODING

CPU ADDRESS LINES A3 - A7 INCLUSIVE ARE USED WITH THE OUT SIGNAL TO DECODE THE FOLLOWING I/P PORTS 130, 140, 150 AND 155 VIA I/C 1, THESE SIGNALS ARE USED TO TOGGLE TWO 'D' TYPE FLIP-FLOPS (I/C 2), (BOTH THESE FLIP-FLOPS ARE RESET ON POWER-UP AND ON OPERATION OF THE RESET BUTTON). WHEN AN I/O OPERATION IS PERFORMED TO PORT 130 THIS PRODUCES A -VE GOING PULSE WHICH IS APPLIED TO THE CLK OF I/C 2A FLIP-FLOP, THIS LATCHES THE LOW HELD ON THE 'D' INPUT ONTO THE 'Q' OUTPUT. THIS OUTPUT IS USED TO CONTROL THE RAM CHARACTER GENERATOR ADDRESS MULTIPLEXERS (I/C 3, 4 AND 5). WHEN IT IS LOW THE GENERATOR RECEIVES IT'S ADDRESSES FROM THE VIDEO DIVIDER CHAIN L1, L2, LA, L?, LBO ETC.

WHEN AN I/O OPERATION IS PERFORMED TO PORT 140 A -VE GOING PULSE IS APPLIED TO THE SET OF I/C 2A, THIS LATCHES A HIGH ON THE 'Q' OUTPUT & THUS THE ADDRESS MULTIPLEXER CONTROL LINE. WITH THE LINE IN THIS STATE THE RAM CHARACTER GENERATOR RECEIVES IT'S ADDRESSES FROM THE CPU ADDRESS BUS.

WHEN AN I/Ø OPERATION IS PERFORMED TO PORT 150 A -VE GOING PULSE IS APPLIED TO THE CLK OF FLIP-FLOP I/C 2B, THIS LATCHES THE LOW HELD ON THE 'D' INPUT ONTO THE 'Q' OUTPUT AND THE Q OUTPUT IS SET TO A HIGH. THE LOW ON THE Q OUTPUT IS FED TO THE CLEAR INPUT OF THE HL-RES VIDEO SHIFT REGISTER I/C 12, THIS FORCES THE SHIFT REGISTER TO CONTINUALLY SHIFT OUT A 'LOW' SIGNAL, THIS IS INVERTED BY I/C 11 AND SENT TO THE KEYBOARD VIA PIN 9 ON THE 20 WAY RIBBON CABLE AND THE I/F PCB TO THE VIDEO DRIVER I/C Z41 PIN 7, THUS THE HI-RES GRAPHICS ARE INHIBITED.



MEMORY MAPPED ADDRESS DECODING

CPU ADDRESS LINES A11 - A15 INCLUSIVE ARE DECODED BY 1/C 14 TO PRODUCE A LOW SIGNAL CALLED ROM A WHENEVER THE CPU ADDRESS BUS IS WITHIN THE Ø - 2K RANGE. THIS SIGNAL IS THEN GATED (1/C 11) WITH THE CPU WR SIGNAL TO PROVIDE THE R/WR REQUIRED BY THE RAM CHARACTER GENERATOR WHILST IT IS BEING PROGRAMMED.

HI-RES VIDEO SHIFT REGISTER.

THE HI-RES VIDEO SHIFT REGISTER SUPPLIES, WHEN ENABLED, VIA INVERTER 1/C 11 SERIAL GRAPHIC VIDEO DATA TO BE MIXED WITH ASCII DATA AT Z41 IN THE KEYBOARD. THE TIMING SIGNALS SHIFT & GRAPHIC SHIFT/LOAD ARE BROUGHT IN FROM THE KEYBOARD VIA THE 2Ø WAY RIBBON CONNECTOR. THE SHIFT REGISTER IS ENABLED & DISABLED BY HAVING IT'S CLEAR LINE TAKEN LOW TO DISABLE IT, THIS IS SELECTED BY DOING AN I/O OPERATION TO PORT 15Ø, FOR MORE INFORMATION SEE ADDRESS DECODING I/O SECTION.

NEW ASCII CHARACTER GENERATOR

THE NEW ASCII CHARACTER GENERATOR IS IN THE FORM OF A PRE PROGRAMMED 2KX8 EPROM. IT PROVIDES TRUE DESCENDING TAILS BY INSTEAD OF BLANKING THE INTER-CHARACTER LINES (L8 - L11) AT Z26-12. THE SIGNAL IS REMOVED AND THE GENERATOR IS PROGRAMMED TO GIVE 4 BLANK LINES TO CHARACTERS WITHOUT TAILS.
(SEE TRACK CUT IN FIG 2 OF FITTING INSTRUCTIONS).
THE GENERATOR ALSO PROVIDES 5 SPECIAL CHARACTERS 123 - 127 TO ENABLE A GRID TO BE DRAWN FOR THE CHARACTER SET-UP PROGRAM SUPPLIED WITH THE MODIFICATION.
THESE NEW CHARACTERS NEED TO BE THE FULL SIX DOTS WIDE, HENCE THE PINK WIRE TO THE ISOLATED LEG Z1Ø-14.

POWER SUPPLIES.

THE UNIT USES THE +5V OF THE KEYBOARD ASSEMBLY FOR IT'S POWER.

GENERAL.

ALL REFERENCES TO PL1 REFER TO THE 4Ø WAY BUS EXTENSION CONNECTOR. IT MUST BE NOTED THAT THE NUMBERING AT PRESENT USES THE CONVENTION OF THE MANUFACTURERS, WHICH IS DIFFERENT TO THE TANDY NUMBERING.

--== oØo ==--

SPORTING FORECASTS

Professor Frank George's well-known Football Pools Forecasting program is now available on the

SINCLAIR ZX81 16K

as well as:

APPLE II 32K
PET 32K
SHARP MZ80K 36K

Versions coming soon for:
Video Genie, BBC-micro, TI 99/4

A Horse-Race Forecast Program in preparation.

Write to: Professor F.H. George
Bureau of Information Science
Commerce House, High Street,
Chalfont St. Giles, Bucks.

IN MAV'S WORKSHOP
BY IAN MAVRIC

BUILDING THE SUPER MICRO PART 2

IN THE LAST ISSUE I TALKED ABOUT HOW I'D SALVAGED A RATHER UNHAPPY MODEL 4 AND WAS TURNING IT INTO THE "SUPER MICRO" WITH MAXED OUT MEMORY, FOUR DISK DRIVES, ETC. IN A MODERN-DAY VERSION OF THE C-U SERIES OF THE SAME NAME[1]. THIS IS HOW I WENT ABOUT THE REPAIRS TO GIVE SOME SORT OF INSIGHT AS TO WHAT GOES INTO A GROUND-UP REBUILD AND UPGRADE TO A 25 YEAR OLD COMPUTER. I CONTINUE THIS MONTH BY INSTALLING FOUR INTERNAL FLOPPY DISK DRIVES.

2. QUAD DOUBLE SIDED DISK DRIVES FOR OVER 2.5MB DISK SPACE

THE STANDARD MODEL 4 DISK DRIVES FOR 1983 WERE TYPICAL OF COMPUTERS OF THE TIME, BEING 5.25" FULL HEIGHT, SINGLE SIDED DOUBLE DENSITY STORING ABOUT 184K EACH, AND THE CABINET HAD SPACE FOR 2 OF THEM. WITHIN A YEAR OR TWO FULL HEIGHT FLOPPY DRIVES HAD MADE WAY FOR HALF HEIGHT FLOPPY DRIVES AND BY 1985 DOUBLE SIDED DISK DRIVES WERE THE STANDARD. IN SOME HIGH-END COMPUTERS LIKE THE TANDY TRS-80 MODEL 2000, THEY EVEN HAD TWIN HALF HEIGHT DOUBLE SIDED 80-TRACK DISK DRIVES[2], AND THAT IS WHERE THINGS BECOME INTERESTING FOR US BECAUSE THE MODEL 4 (AND THE MODEL III, 4P, AND EVEN THE MODEL 1 (TO A LIMIT OF 3 DRIVES)) ALREADY HAD THE HARDWARE IN PLACE TO WORK WITH DOUBLE SIDED AND 80 TRACK DISK DRIVES. TURNS OUT ALL YOU NEED IS TO MAKE A SPECIAL CABLE[3], MAKE POWER SPLITTERS FOR 4 DRIVES, AND MOUNT THEM ALL IN THE MACHINE. HERE'S HOW I DID IT:

2A. SOURCE THE DRIVES

I DECIDED TO INSTALL THREE DISK DRIVES FROM A TANDY 2000 WHICH ARE MITSUBISHI M4853 WHICH ARE, AS STATED, 80 TRACK DOUBLE SIDED DOUBLE DENSITY DISK DRIVES WITH A CAPACITY OF 720K EACH. THEY ARE GREAT BECAUSE THEY ARE LOW-POWER CONSUMPTION DIRECT DRIVE MOTORS WITH TYPICAL JAPANESE OVER-ENGINEERING. FOR THE FOURTH DISK DRIVE I WENT TO EBAY AND BOUGHT A MITSUBISHI M4851 WHICH IS BASICALLY THE SAME DRIVE, EXCEPT 40 TRACK 360K CAPACITY BUT LOOKS THE SAME SO THE FINISHED PROJECT HAS A NEAT CONSISTENT APPEARANCE ABOUT IT, NOT A MISHMASH OF DISK DRIVES. (THAT'S NOT TO SAY A MISHMASH OF DRIVES DOESN'T WORK - IT DOES - BUT HAVING ALL THE FOUR DRIVES LOOKING THE SAME IS PRETTIER.) WITH MY FOUR DRIVES THE SYSTEM NOW HAS 2520K STORAGE [(720*3)+360] VS. THE ORIGINAL 368K (184*2) AND AROUND 1/2 THE CAPACITY OF ONE OF TANDY'S 5MB EXTERNAL HARD DRIVES[4].

2B. CONTROLLERS AND CABLE

I WENT WITH A PROGRESSIVE COMPUTER PRODUCTS FDC-III DISK CONTROLLER I HAD LYING AROUND HERE AND IT ALLOWS YOU TO CONNECT A 4 DRIVE CABLE FOR A NEAT INSTALLATION (SEE PIC). UNLIKE TANDY'S DISK DRIVE CABLE, NO TEETH ARE PULLED, SO THE CABLE HAS AT ONE END A 34-WAY CONNECTOR, THEN AT THE OTHER END THE 4 DISK DRIVE CONNECTORS SPACED APPROPRIATELY APART TO CONNECT THE FOUR DRIVES. I

MADE THE CABLE UP WITH PARTS FROM TANDY USING RIBBON CABLE (CAT. NO. 276-9773) AND 34-POSITION CARD EDGE CONNECTORS (CAT. NO. 276-1564).

ALTERNATELY IF YOU USE A TANDY DISK CONTROLLER THE CABLE SET UP IS A BIT DIFFERENT, REQUIRING TWO CABLES, YOU NEED TO MAKE A CABLE WITH 2 CONNECTORS FOR DRIVE Ø AND 1, WHICH CONNECTS TO THE TOP OF THE DISK CONTROLLER, AND THEN MAKE A 2ND LONGER CABLE WHICH CONNECTS FROM THE BOTTOM OF THE DISK CONTROLLER TO THE TWO TOP DRIVES (2 AND 3). YOU PLUG THIS 2ND CABLE INTO THE BOTTOM OF THE DISK DRIVE PORT AND ROUTE IT UP INTO THE CASE TO CONNECT TO THE OTHER TWO DRIVES.

TO POWER ALL FOUR DRIVES FROM THE EXISTING TWO DISK DRIVE POWER JACKS, YOU'LL NEED A PAIR OF "MOLEX" DISK DRIVE POWER CABLE SPLITTERS (JAYCAR[5] PART NUMBER PLØ75Ø) AND WE NOW HAVE POWER AND DATA CONNECTIONS TO ALL 4 DRIVES.

AT THIS POINT YOU NEED TO DETERMINE WHICH DRIVE WILL BE Ø,1,2 AND 3, AND THEN SET THE DRIVE SELECT JUMPERS ACCORDINGLY. DRIVE Ø WILL USUALLY BE AT THE END OF THE LONG 4-DRIVE CABLE, SO YOU NEED TO LEAVE THE TERMINATOR ENABLED ON THAT DRIVE, AND REMOVE THE TERMINATOR FROM THE OTHER THREE DRIVES.

2C. MOUNT THE 4 DISK DRIVES

NOW IS WHERE THE FUN STARTS, I'LL DESCRIBE MY PREFERRED WAY TO INSTALL THEM. IF WE TAKE INTO ACCOUNT THAT DRIVE Ø AND 2 WILL INSTALL IN THE RIGHT PLACE USING THE ALREADY-EXISTING HOLES, IT THEN BECOMES SIMPLY A PROBLEM OF DRILLING SUITABLE HOLES TWO MOUNT THE OTHER 2 DRIVES. I REMOVE THE DRIVE TOWERS AND RULE LINES FROM TOP TO BOTTOM THROUGH THE EXISTING HOLES. THIS GIVES ME THE VERTICAL POSITIONING. I THEN DRILL LARGE 6MM HOLES "MORE OR LESS" WHERE THE NEW DISK DRIVES NEED TO MOUNT. I THEN USE SCREWS WITH LARGE WASHERS, WHICH GIVE ME A CERTAIN AMOUNT OF BACK-FORTH-UP-DOWN MOVEMENT OF THE DRIVES TO GET THEM CORRECTLY INTO PLACE. YOU'LL NOTICE ON THE TOP DRIVE IT CAN ONLY BE MOUNTED BY TWO SCREWS ON THE LEFT-HAND TOWER, BECAUSE THE RIGHT-HAND TOWER IS SIGNIFICANTLY SHORTER, SO ARRANGEMENTS NEED TO BE MADE ON THAT SIDE SO THE MOTORS ON DRIVE 3 DON'T INTERFERE WITH THE ELECTRONICS ON DRIVE 2. I MODIFIED THE RIGHT-SIDE RFI SHIELD TO GIVE DRIVE 3 JUST ENOUGH CLEARANCE FROM DRIVE 2 AND IT WORKS - NO SHORTS OR OTHER DRAMAS... EXCEPT ONE...

2D. KEEP IT ALL COOL

WITH 4 DISK DRIVES IN THE CABINET IT DOES GET PREDICTABLY PRETTY HOT IN THERE - AS YOU KNOW WHEN A TRS-8Ø ACCESSES ONE DISK DRIVE, ALL OF THEM SPIN UP, AND SO A GREAT DEAL OF HEAT IS GENERATED INSIDE THE CABINET, AND THE DISKS FEEL QUITE WARM WHEN REMOVED FROM THE DRIVE. I SOLVED THIS BY INSTALLING A SMALL 12V 8ØMM PC CASE FAN (JAYCAR CAT. NO. YX2512) IN THE BOTTOM OF THE M4, POWERED OFF ONE OF THE DISK DRIVE POWER CABLES. IT DOESN'T MAKE MUCH NOISE LIKE

AN A/C FAN, AND MOVES ENOUGH AIR AROUND TO KEEP TEMPERATURES UNDER CONTROL.

NEXT TIME: PART 3: MAKING 80-TRACK BOOTING VERSIONS OF POPULAR OPERATING SYSTEMS

---=== o0o ===---

IN MAV'S PROBLEM SOLVING WORKSHOP

YELLOWED KEYS

ON YOUR MODEL I, III AND ESPECIALLY MODEL 4, WE HAVE SOME QUITE NICE LOOKING WHITE KEYS. "ENTER" ON THE EARLY MACHINES AND "ENTER" "CAPS" "CTRL" "BREAK" "CLEAR" "SHIFT" "F1" "F2" "F3", ARROWS ETC. ON THE MODEL 4. WHEN TANDY DESIGNED THESE MACHINES THEY WERE NEVER PREPARED FOR THE ENVIRONMENTS THEY WERE EVENTUALLY SENT TO, NOR WHAT POOR CONDITIONS THEY WOULD BE STORED IN FOR YEARS OR DECADES, OR THAT INDEED ENTHUSIASTS WOULD STILL BE DOING THINGS WITH THEM 30+ YEARS AFTER THEY WERE MADE.

IF YOU'VE EVER LOOKED AT OLD RADIO SHACK COMPUTER CATALOGUES YOU'D NOTICE WITH SOME DISMAY HOW BEAUTIFULLY WHITE THE KEYS ORIGINALLY WERE AND HOW THEY LOOK THESE DAYS... THEY SEEM TO BE AFFECTED THREE THINGS WHICH WILL TURN THEM YELLOW OVER TIME. 1. DIRECT SUNLIGHT, 2. INDIRECT SUNLIGHT, AND 3. NICOTINE.
DIRECT SUNLIGHT IS WHEN A COMPUTER IS STORED OR USED OUTDOORS OR UNDER A WINDOW, WITH THE SUNS RAYS DIRECTLY HITTING THE KEYS. THEY WILL YELLOW, AS WILL THE PLASTICS OF THE COMPUTER. INDIRECT SUNLIGHT IS WHERE A COMPUTER IS SET UP IN A NORMAL ROOM, LIKE MY COMPUTERS ARE, THE SUN NEVER SHINES DIRECTLY ON THEM, BUT THE ROOM IS WELL LIT BY LARGISH WINDOWS, AND THE KEYS GO A BIT OFF COLOUR. NICOTINE YELLOWED KEYS ARE FROM COMPUTERS WHO HAVE LIVED IN HOUSEHOLDS WITH SMOKERS, AND AFTER SOME TIME PRACTICALLY EVERYTHING IN THE HOUSE TAKES ON A YELLOWISH HUE. I RECENTLY ACQUIRED A MODEL 4 WHICH HAD LIVED ITS WHOLE LIFE IN A PASSIVE SMOKE ENVIRONMENT (FOR THE 10 YEARS IT WAS IN USE, THEN THE 15 YEARS IT WASN'T IN USE SITTING UNDER A DESK) AND THEY KEYS WERE A PARTICULARLY AWFUL YELLOW. AN OLD TRICK WITH YELLOWED PLASTICS ON CARS (LIKE THE RADIATOR EXPANSION TANK AND WINDSCREEN WASHER BOTTLE) WAS TO PUT THEM IN HOUSEHOLD BLEACH[6] FOR A FEW DAYS WHICH TAKES AWAY THE YELLOWY/ORANGEY APPEARANCE ON THOSE PARTS WHICH LIVE IN THE ENGINE BAY. I ALWAYS SPECULATED THAT IT WOULD WORK WITH WHITE KEYS AND YOU CAN SEE THE DIFFERENCE IN THE BEFORE/AFTER PICTURE PRESENTED HERE IN COLOUR TO SHOW HOW WELL BLEACH WORKS ON NICOTINE.
NOTES:
THE KEYS IN THE PICTURE WERE BLEACHED FOR 7 DAYS. MORE THAN THAT AND THE WRITING ON THE KEY STARTS TO TURN A DULL GREY.
KEYS AFFECTED BY INDIRECT SUNLIGHT ALSO WHITEN UP NICELY, CHECK THEM DAILY AND THEY START TO LOOK GOOD AFTER ABOUT 5 DAYS.
KEYS AFFECTED BY DIRECT SUNLIGHT NEVER REALLY GET BACK THEIR ORIGINAL WHITENESS, THEY CAN BE

IMPROVED WITH ABOUT 7 DAYS BLEACHING BUT NEVER LOOK AS GOOD AS ORIGINAL.
DON'T BOTHER WITH BLACK KEYS BECAUSE THE BLEACH AFFECTS THE TEXTURED FINGER SURFACE, TURNING IT A DULL DARK GREY INSTEAD OF BLACK.
BLEACH IS H-A-R-S-H AND THE KEYS NEED A GOOD LONG RINSE IN CLEAN WATER ONCE DONE TO STOP THE CAUSTIC PROCESS.
OTHER PEOPLE HAVE SUCCESSFULLY USED A RETRØBRITE PROCESS (GOOGLE IT) WHICH IS LESS HARSH BUT MUCH MORE PREPARATION AND MIXING OF CHEMICALS IS REQUIRED. MY METHOD INVOLVES REGULAR $1.20/LITRE HOUSEHOLD BLEACH.

IN MAV'S RESTORATION WORKSHOP

THE YASS SERVICE STATION MODEL 1

I WAS CONTACTED VIA MY WEB SITE FROM SOMEONE WISHING TO DISPOSE OF A MODEL I COMPUTER SYSTEM IN YASS, NSW. YASS IS A SMALL TOWN IN THE MIDDLE OF NOWHERE TYPICAL OF MANY AUSTRALIAN SMALL TOWNS, IT HAS A PUB, A MOTEL, A GENERAL STORE, SOME HOUSES AND AS SERVICE STATION. I COMMENTED NOT MANY MODEL 1S WOULD HAVE BEEN SOLD IN YASS AND HE SAID THIS WAS THE OLD SERVICE STATION COMPUTER.
IT WAS IN USE FROM 1980 UNTIL THE LATE-90S AND ITS JOB WAS TO CALCULATE PETROL INVOICING AND ON MORE THAN ONE OCCASION HELPED SAVE THE PLACE A GOOD AMOUNT OF MONEY BY IDENTIFYING WHEN EMPLOYEES WERE STEALING PETROL.

THE SOFTWARE WASN'T INCLUDED, THE DISKS LONG SINCE PERISHED BUT THE COMPUTER WAS PRETTY NORMAL OF MANY SMALL BUSINESS MODEL 1 SYSTEMS: A 16K LEVEL II SYSTEM WITH NO LOWERCASE, A 32K EXPANSION INTERFACE AND STANDARD TANDY/SANYO B&W MONITOR.
THE DISK DRIVES WERE IN A HOME MADE WOODEN BOX WITH RATHER SCARY LOOKING POWER SUPPLY, AND THE SYSTEM WAS STANDARD SINGLE DENSITY WITH NO RS232. WHEN I SET IT UP ON MY WORKBENCH IT DIDN'T WANT TO RUN, JUST HAD A SCREEN FULL OF GARBAGE.

UPON OPENING UP THE COMPUTER I FOUND A MUMFORD MICRO SYSTEMS[7] SPEEDUP MODIFICATION WHICH TURNED OUT TO BE CAUSING THE PROBLEM, SO AFTER CAREFULLY REMOVING THAT THE MACHINE STARTED UP TO THE USUAL MEM SIZE? AND WITH A DISK IN THE DRIVE, OFF IT BOOTED AND WORKED QUITE WELL. THE SCREEN HAD SOME SCREEN-BURN AND THE PAINT WAS WORN OFF THE FRONT OF THE COMPUTER, BUT IT DID RUN WELL ENOUGH.
I CAN ONLY SUPPOSE THE SPEEDUP KIT WAS ADDED TO ASSIST WITH THE END-OF-MONTH BATCH PROCESSING OF ALL THE PETROL INVOICES.
IT'S DIFFICULT TO EXPLAIN TO NON-AUSTRALIAN READERS THE IMPORTANCE OF THE YASS SERVICE STATION, ITS STRATEGICALLY PLACED ON THE HUME HWY AT THE RIGHT PLACE TO FILL UP PETROL OR DIESEL DURING THE LONG AND POPULAR MELBOURNE-SYDNEY ROAD TRIP.
THOUSANDS OF CARS AND TRUCKS NEED TO FILL UP THERE EVERY WEEK AND ITS A MAJOR EMPLOYER FOR THE CITY OF YASS.

IANM@TRS-80.COM

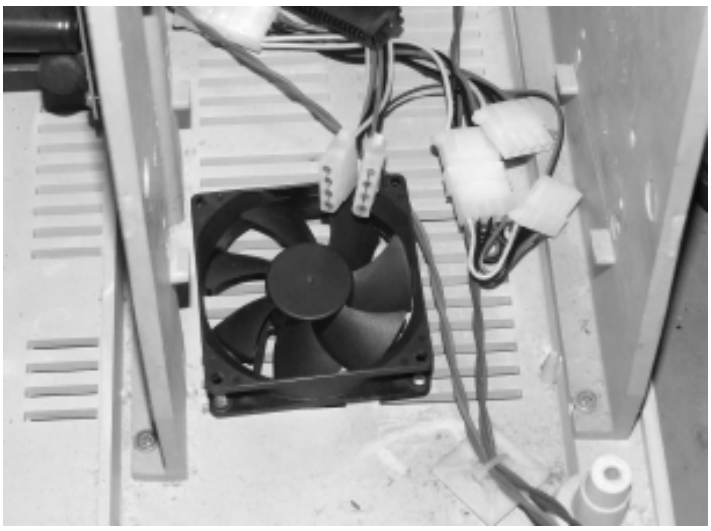
REFERENCES:

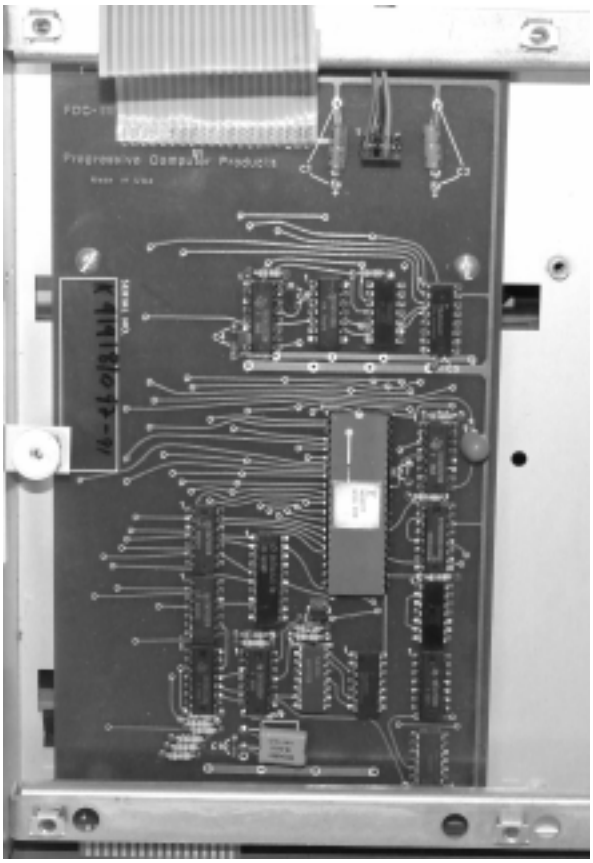
1. "BUILDING THE SUPER MICRO", COMPUTER USER, FEBRUARY 1984 PP. 14- 16; MARCH 1984 PP. 14-18; APRIL 1984 PP. 36-38; MAY 1984 PP. 26-30.
2. RADIO SHACK COMPUTER CATALOG RSC-11, PP. 4-7
3. "HYDRA DISK", 80MICROCOMPUTING #27. MARCH 1982 PP.206-208.
4. RADIO SHACK COMPUTER CATALOG RSC-8, PP. 52
5. HTTP://WWW.JAYCAR.COM.AU/
6. HTTP://WWW.DOMESTOS.CO.UK/PRODUCTS/MULTIUSE-BLEACH.PHP
7. HTTP://WWW.BMUMFORD.COM/












NEWDOS/80




BEFORE
BREAK
F1
F2
F3
ENTER
CLEAR
ENTER
CTRL
SHIFT


BREAK
F1
F2
F3
ENTER
CLEAR
AFTER
CTRL
SHIFT
ENTER

(CONTINUED FROM PAGE 1)
ADDITIONAL WEB PAGE, JUST IN CASE AND I LIVE IN HOPES THAT SOMEONE OUT THERE MIGHT BE WILLING AND ABLE TO HELP!

I'VE STILL NO FURTHER NEWS TO REPORT ABOUT EITHER FRANK CERVENKA'S ANDROID TRS-80 EMULATOR, DROIDTRS OR GETTING AN ACULAB FT RUNNING ON MATTHEW'S EMULATOR. IF I DO HEAR ANYTHING, I'LL POST IT ON THE WEBSITE ASAP.
ALSO, I'VE HAD NO RESPONSE REGARDING A 'WORKSHOP' DAY IN THE UK. I NOTICE THAT ON THE VINTAGE COMPUTER FORUM, THERE'S A 'BIG' GET TOGETHER IN NEW YORK IN MAY 2012. A BIT TOO FAR (AND EXPENSIVE) TO TRAVEL THOUGH I BET IT WOULD BE FUN!

AND NOW, DOWN TO THIS ISSUE. AT THE READY PROMPT RETURNS WITH A COUPLE OF IDEAS FROM A CODING PERSPECTIVE.

I CAME ACROSS AN ADVERT FOR A SOUTHERN SOFTWARE PROGRAM CALLED 'SBE' (SMALL BUT EFFECTIVE). I PUT OUT A REQUEST TO SEE IF ANYONE HAD A COPY FOR SALE ON THE VINTAGE COMPUTER FORUM, BUT ALTHOUGH IT HAD OVER 80 HITS, NO ONE CAME UP WITH ANYTHING. THEN I MANAGED TO FIND A REVIEW BY ARIELA AND GEOFFERY TAYLOR WHICH WAS ORIGINALLY PUBLISHED IN NATGUG NEWS. I HOPE YOU FIND IT OF INTEREST, IT DOES SEEM TO BE A FASCINATING PIECE OF SOFTWARE.

WHILE I'M ON THE SUBJECT OF ADVERTS, I'VE MANAGED TO FIND SOME LESS WELL KNOWN ONES, (THERE'S EVEN ONE FOR NATGUG!) FROM AN OLD COPY OF PERSONAL COMPUTER WORLD, WHICH SHOULD PUT A SMILE ON YOUR FACES! .
THE ONE ON THE BACK PAGE ESPECIALLY SO, AS PETER STONE HIS HOPING TO COMPLETE HIS SERIES OF ARTICLES ON CODING IN THE NEXT ISSUE.
AND WHILE I'M ON ABOUT THE NEXT ISSUE, I SHALL BE FEATURING 'ONE-LINERS'. WE'VE NOT HEARD ANYTHING ABOUT THEM FOR AGES SO I THOUGHT, FOR THE FIRST TIME, I'LL HAVE A FEATURE ISSUE. SO, **CAN YOU HELP AND SEND IN A ONE-LINER**? NOW'S THE TIME TO PUT YOUR THINKING CAPS ON AND HAVE A GO OVER THE NEXT COUPLE OF MONTHS.

FOR THOSE OF YOU WHO ARE REGULAR VISITORS TO THE WEBSITE, YOU WILL NO DOUBT HAVE NOTICED THAT JOHN BENSON, IN AUSTRALIA, HAS BEEN ASKING FOR DETAILED INSTALLATION INSTRUCTION FOR THE SPECIFIC TANDY HIRES UPGRADE, WHICH WAS ONLY AVAILABLE IN THE UK. NO LUCK SO FAR, BUT THIS DETAILED REVIEW, BY JOHN KILPATRICK IS VERY COMPREHENSIVE.

WE FINISH OFF THIS ISSUE WITH PART 2 OF IAN MAVRIC'S ACCOUNT OF BUILDING THE SUPER MICRO, TOGETHER WITH A COUPLE OF OTHER RECENT PROBLEMS HE'S ENCOUNTERED. HE REALLY HAS COME UP WITH SOME CRACKING IDEAS.

WELL. I THINK THAT'S ABOUT IT FOR THIS ISSUE. PLEASE HAVE A GO AT THE ONE-LINERS, I'M LOOKING FORWARD TO HEARING FROM YOU.



TAKE CARE & BYE FOR NOW

DUSTY

Software News

INNOVATIVE TRS 80·GENIE SOFTWARE

from the professionals

£250 REWARD

Below you will find described a new program entitled Enigma. It is a true simulation of the German wartime cypher machine of that name. It will encipher messages which may be communicated to third parties by any means who, assuming they have the key, will be able to use their Enigma program to decipher.

We will pay the sum of £250 to anyone [who has purchased the program] who can demonstrate an infallible method of deciphering the coded message supplied in the program's instructions. We consider Enigma to be the best program of its kind on the U.K. or U.S. market; contestants may therefore use any orthodox means to crack the code, including microcomputer programs other than Enigma.

The original message and keys will be lodged with our Solicitors for safe keeping in a sealed envelope. In the [hopefully] unlikely event that the code is cracked by more than one person, the reward will be paid to the first customer who demonstrates to us that he has succeeded.

MOLIMERX LTD.

During the 1939/45 war the German Army and Intelligence used a deciphering machine called Enigma. It was a fascinating machine and the stories that have surrounded it are equally interesting. There have been some four or five books written about the machine, and with regard to the way in which the British counter intelligence managed to crack the code.

That they did so was the culmination of some fortuitous circumstances, a lot of luck, but mainly it was due to the fact that the people who did it were extremely clever mathematicians. The fact that it took so much brain power, plus a rudimentary type of computer and a specimen of the machine in order to crack the code is an indication of how complex that code is.

The Enigma microcomputer program that we are selling is a simulation of the original machine, together with one or two improvements which were suggested by Gordon Welchman, who wrote the book "The Hut Six Story" last year and was also the leader of the team that cracked the code.

Although the machine and, therefore, the program is so complicated, its use is amazingly simple. One simply inputs a key and a message and the code is supplied. To decipher, the message is input again with the key and if the key is correct then the decoded message is displayed. With the cassette version it is necessary to input from the keyboard but with disks both inputs and outputs may be to disk files if required. A printer is of course supported.

The code may be transmitted in any way which the written word can be transmitted. Companies who wish to fully protect their communications will no doubt have the program generate the code and then tap it into a telex. Tape users will have to send either the output from their printer or write down the code direct from the screen.

Enigma is a fascinating program designed, not only for those people who are interested in encryption professionally or as a hobby, but also for companies or private persons who wish to communicate with others in an entirely secure manner. As is shown by the above Reward Notice, we have great faith in the powers of this piece of software.

ENIGMA (Tape) ... **£17.25**
ENIGMA (Disk) ... **£23.00**
Inclusive of V.A.T. P&P 75p

TEL: [0424] 220391/223636

MOLIMERX™ LTD
A J HARDING (MOLIMERX)

TELEX 86736 SOTEX G

1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 plus £1 postage.

TRS8BIT



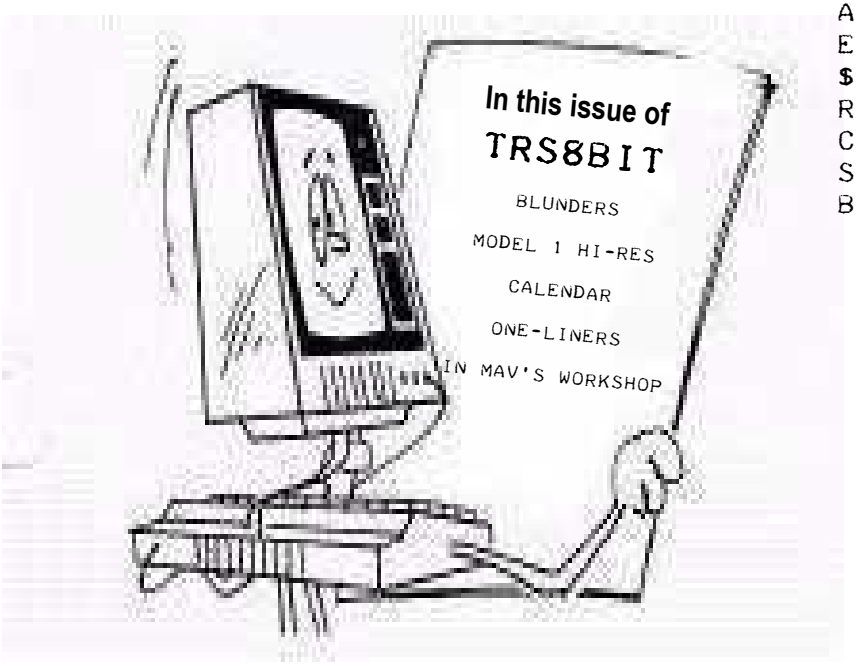
HI EVERYONE AND WELCOME TO THIS VERY 'SPECIAL' JUNE 2012 EDITION.

NO PRIZES FOR GUESSING THAT THIS ISSUE FEATURES ONE-LINERS. THE RED BANNER IS A DEAD GIVE AWAY EH! MANY THANKS TO ALL THOSE WHO CONTRIBUTED, MAKING THIS, ONE OF THE BIGGEST ISSUE TO DATE, WELL WORTH THE TITLE OF 'SPECIAL'. I HOPE YOU FIND ALL THE CONTRIBUTIONS AS ENJOYABLE AND AMAZING AS I DO.

I'VE STILL NO INFORMATION ABOUT ANYONE IN THE UK WHO CAN OFFER A REPAIR OR UPGRADE SERVICE FOR TANDY'S, BUT I'LL KEEP THE PAGE ON THE WEBSITE OPEN, JUST IN CASE I HEAR FROM ANYONE!

Special 'one-liner' programs collectors edition

THE UK'S EBAY HAS BEEN QUITE BUSY WITH BITS FOR SALE OVER THE LAST THREE MONTHS WITH M1'S, E.I.'S AND M4 APPEARING ON A REGULAR BASIS. NO SPECTACULAR PRICES TO REPORT. HOWEVER, MAV. EMAILED ME WITH DETAILS ON AN ITEM, SELLING ON EBAY.COM, AT JUST OVER $2000. THAT'S AN AWFUL 'LO-RA-MONEY' JUST FOR A COMPUTER GAME EH! I'M SURPRISED IT'S THAT RARE, BEING A GAME.





27)

BLUNDERS!
AND APOLOGIES



I'VE REPLACED, JUST FOR THIS ISSUE, AT THE READY PROMPT, WITH THESE FEW WORDS!

FIRST- THE ARTICLE IN THE LAST EDITION OF TRS8BIT REGARDING A HI-RES GRAPHICS MOD. A COUPLE OF YOU CORRECTLY POINTED OUT THAT IT DIDN'T CLEARLY STATE WHICH OR WHO'S MOD IT WAS.

THE ARTICLE, BY JOHN KILPATRIC, ORIGINALLY APPEARED IN NATGUG NEWS AND I ASSUMED THAT, AT THAT TIME, 1981 (ISH), THERE WAS PERHAPS JUST THAT MOD AVAILABLE SO FURTHER DETAILS REGARDING MANUFACTURE ETC. WERE FELT TO BE UNNECESSARY. I WAS HOPING THAT, AS THERE SEEMS TO BE QUITE A BIT OF INTEREST IN M1 HI-RES OUT THERE, SOMEONE MIGHT RECOGNISE, BY READING THE ARTICLE, WHICH BIT OF KIT IT WAS AND EVEN, PERHAPS, IDENTIFY THE MANUFACTURER.

SORRY-I'LL TRY AND MAKE SUCH THINGS CLEARER IN THE FUTURE.

I KNOW THAT JOHN BENSON, IN AUSTRALIA IS STILL LOOKING FOR DETAILED INSTRUCTION FOR THE TANDY HI-RES MOD, REF: 26Ø-98ØØ, (WHICH WAS ONLY AVAILABLE IN THE UK).

AS A CONSEQUENCE OF FURTHER SEARCHING, I CAME ACROSS ANOTHER ARTICLE IN NATGUG NEWS, THIS TIME BY E.C. KILPATRICK, FEATURING ANOTHER HI-RES KIT WHICH I'VE INCLUDED IN THIS ISSUE, HOPING THAT, AGAIN, IT MIGHT JUST HELP SOMEONE OUT THERE WITH A PROJECT. IF YOU DO IDENTIFY THE MOD KIT, PLEASE LET ME KNOW AND I'LL PUT DETAILS ON THE WEB SITE.

SECONDLY- I AM SORRY IF THE NEWSLETTER SEEMS TO BE SOMEWHAT BIASED TOWARDS THE MODEL 1. ALTHOUGH UNINTENTIONAL, THE REASON FOR THIS IS QUITE STRAIGHT FORWARD. OVER THE YEARS, I'VE ONLY OWNED A NON-DISK M1 SETUP. EVEN NOW, MY 'REAL' SYSTEM IS A 16K M1 WITH AN E.I., CASSETTE RECORDER AND A (NOT FULLY WORKING) ACULAB FLOPPY TAPE. THE ONLY ACCESS TO DISKS I HAVE, IS VIA MATTHEW REED'S FANTASTIC EMULATOR.
SO PLEASE, DON'T THINK THAT I'M ONLY LOOKING FOR ARTICLES ABOUT A M1. MODEL 2,3,4 AND 4P ARTICLES WILL BE MOST WELCOME IF YOU WANT TO SEND THEM IN!

MIND YOU, THE MODEL 1 IS STILL, BY FAR, MY FAVOURITE COMPUTER!

CONSULTANT ELECTRONICS

VIDEO GENIE COMPUTERS
Books – C10 Cassettes – Disks
Your own 'Basic' Genie /
TRS80 Programs listed.
Vic 20, ZX81, Genie, TRS80
Software.

59 Mason Road, Erdington,
Birmingham. Tel 021-382 7247

MODEL 1 HI-RES

EC KILPATRICK

THE MODIFICATION COMES IN TWO PARTS

1. A SEPARATE, FULLY TESTED & ASSEMBLED, GRAPHICS BOX THAT PLUGS INTO THE LEFT HAND SIDE OF THE INTERFACE OR THE KEYBOARD EDGE CONNECTOR.
2. AN INTERFACE PRINTED CIRCUIT BOARD, WITH ASSOCIATED COMPONENTS, THAT HAS TO BE MOUNTED INSIDE THE KEYBOARD.
BEFORE ATTEMPTING THE MODIFICATION, PLEASE STUDY THE DIAGRAMS & INSTRUCTIONS CAREFULLY.

FITTING INSTRUCTIONS.

1. CAREFULLY OPEN THE KEYBOARD & POSITION IT IN A MANNER THAT WILL ALLOW EASY ACCESS FOR A SYSTEMATIC APPROACH TO THE MODIFICATIONS REQUIRED. IF YOUR KEYBOARD IS NOT THE NEW STYLE THEN Z29 WILL NOT BE IN AN IC HOLDER, IT WILL NEED REMOVING AND REPLACING WITH AN 18 PIN IC SOCKET. Z46 WILL ALSO NEED REMOVING AND REPLACING WITH A 16 PIN IC SOCKET.
A CAUTIONARY NOTE.
GREAT CARE MUST BE EXERCISED WHEN SOLDERING SOCKETS TO PREVENT SOLDER CAUSING SHORTS BETWEEN ADJACENT PINS.
2. TWO TRACK CUTS ARE NECESSARY AS INDICATED IN FIGS. 1 & 2. . THE FOLLOWING IC PINS NEED TO BE ISOLATED FROM THEIR CURRENT POSITIONS. THIS CAN BE ACHIEVED BY EITHER DE-SOLDERING THE LEG & BENDING IT AT RIGHT ANGLES TO THE BOARD, OR BY USING A SHARP PAIR OF POINTED NOSE CUTTERS, CUT THE IC LEG JUST ABOVE THE BOARD & BEND THE REMAINS OF THE LEG AT RIGHT ANGLES TO THE BOARD.

IC	PIN
Z10	14
Z11	9
Z41	7
Z72	14

4. INSERT PIGGY-BACK 21L02 RAM INTO Z46 SOCKET & ATTACH THE WIRES AS SHOWN IN FIG 3
WARNING.
IT IS ADVISABLE TO CHECK FOR SHORTS BETWEEN IC PINS BEFORE INSTALLING HOLDERS.
5. CONNECTIONS.
LINK PINS 12 & 13 OF IC Z26.
USING AN INSULATED WIRE CONNECT IC Z72 PIN 14 (LEG ONLY) TO IC Z74 PIN 9.
COVER THE CONTACTS OF THE POWER SWITCH WITH INSULATING TAPE AS IN FIG 5.
PLACE THE NEW PRINTED CIRCUIT BOARD BEHIND THE POWER, VIDEO & CASSETTE SOCKETS AS IN FIG 5.
CONNECT THE COLOURED WIRES TO THE POSITIONS INDICATED IN FIG 4.
SLEEVES SHOULD BE USED OVER CONNECTIONS TO THE BENT OUT LEGS.
6. CONNECT 16 PIN JUMPER LEAD TO SOCKET IN Z29 AS SHOWN IN FIG 5, LEAVING PINS 9 & 10 OPEN CIRCUIT.
AT THIS POINT CHECK ALL CONNECTIONS (OLD ROM VERSIONS CHECK FOR PUNCTURED WIRES, CONNECTING THE SMALL ROM BOARD STUCK TO THE BACK OF THE MAIN PRINTED CIRCUIT BOARD TO THE MAIN BOARD).
7. CONNECT POWER & VIDEO & TEST THAT MACHINE FUNCTIONS CORRECTLY (ENSURING THERE ARE NOT ANY METAL OBJECTS UNDER THE EXPOSED BOARD).
THE 20 WAY CABLE MUST BE ROUTED OUT OF THE REAR OF THE TWO HALVES OF THE KEYBOARD CASE ASSEMBLY. (SEE FIG 4).
CONNECTION TO THE EXTERNAL GRAPHICS BOX IS BY WAY OF THE 20 WAY PLUG WITH IT'S LOCATING PEG FACING UPWARDS.
ENSURE THAT THE PLUG IS FIRMLY PUSHED IN & THAT THE SOCKET'S EJECTOR LUGS ARE CLOSED SO THE CONNECTION IS POSITIVE.
8. CAREFULLY FIT THE MACHINE BACK INTO IT'S CASE, POWER UP & RUN THE DEMONSTRATION PROGRAM.

CALENDAR

KEVIN SMITH

I CAME ACROSS THIS RATHER SUPER LITTLE PROGRAM, ORIGINALLY WRITTEN FOR THE NASCOM 2 BY KEVIN SMITH.

IT CALCULATES AND PRINTS THE CALENDAR FOR ANY MONTH IN ANY YEAR BETWEEN THE DATES (WAIT FOR IT) 25000 BC TO 20000 AD. IT EVEN TAKES INTO ACCOUNT THE CALENDAR REFORMS OF 1752. TRY THE CALENDAR FOR SEPTEMBER 1752 WHEN THE JULIEN CALENDAR WAS ABANDONED IN FAVOUR OF THE GREGORIAN ONE.

TO ENTER A DATE BC, PREFIX THE YEAR WITH A MINUS SIGN. THE ARITHMETIC IS QUITE CLEVER AND SEEMS TO MANAGE THE LEAP YEAR IN 1900, 2000 AND 2012 CORRECTLY.

```
1000 REM CALENDARS BY KEVIN SMITH
1010 CLS:CLEAR1000:PRINT"CALENDARS":PRINT
1020 DIMM$(12):FORI=1TO12:READM$(I):NEXT
1030 DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
1040 INPUT"ENTER MONTH AND YEAR";M,Y
1050 IF M<0 OR M>12 OR Y<-25000 OR Y>20000 THEN 1040
1060 I=Y:A$="AD":IFY<0THENA$="BC":I=-I:Y=Y+1
1070 CLS:PRINTTAB(11);"MONTH - ";M$(M);I;A$
1080 GOSUB1210:I=J:PRINT
1090 PRINTTAB(11);"S  M  T  W  T  F  S"
1100 M=M+1:IFM>12THENM=1:Y=Y+1
1110 GOSUB1210:N=J-I:J=I-INT(I/7)*7+1
1120 IFJ=7THENJ=0
1130 J=J*3+10:K=1
1140 IF Y<>1752 OR M<>10THEN1160
1150 PRINTTAB(J);" 1  2";:K=14:J=22:N=30
1160 FORI=KTON:PRINTTAB(J);
1170 IF I>9THENPRINTCHR$(8);
1180 PRINTI;
1190 J=J+3:IFJ>30 THEN PRINT:J=10
1200 NEXTI:PRINT:PRINT:GOTO1040
1210 K=Y+4712:J=INT(K/4)+365*K
1220 N=30.6*M-32.3
1230 IFM>2 THEN1250
1240 N=N+2.3:IFK-INT(K/4)*4=0THENJ=J-1
1250 J=J+INT(N+1)
1260 IF J<=2361221 THEN RETURN
1270 K=Y-300
1280 IF M<3 THEN K=K-1
1290 N=INT(K/100)
1300 J=J-INT(.75*N)-1:RETURN
1398 REM ORIGINALLY WRITTEN FOR THE NASCOM 2 CIRA 1983
1399 REM AMENDED FOR TRS8BIT 2012
```

--== o0o ==--

ONE-LINERS
FOR TRS-80 BASIC (PART 1)

BRIAN RAITER

THIS PROGRAM IS A COLLECTION OF ONE-LINERS WRITTEN BY VARIOUS AUTHORS. THAT IS, EACH SUB-PROGRAM IN THIS PROGRAM ORIGINALLY STARTED OUT AS A ONE-LINE PROGRAM. THEY HAVE BEEN COLLECTED TOGETHER INTO A SINGLE PROGRAM SO THAT YOU CAN EXPERIENCE THEM ALL FOR YOURSELF WITHOUT HAVING TO LOAD EACH ONE UP SEPARATELY.

THERE ARE A TOTAL OF 14 DIFFERENT ONE-LINERS. MOST ONE-LINERS RUN IN INFINITE (OR NEAR-INFINITE) LOOPS. A FEW OF THE PROGRAMS ORIGINALLY EXITED TO THE READY PROMPT; THOSE ARE HANDLED HERE BY LETTING THEM FALL INTO INFINITE LOOPS INSERTED BETWEEN ONE-LINERS. AT ANY TIME, YOU CAN HIT SHIFT-CLEAR TO LEAVE A ONE-LINER AND RETURN TO THE INITIAL MENU. TO EXIT THE PROGRAM, PRESS ENTER AT THE INITIAL MENU - I.E. WITHOUT PROVIDING ANY INPUT.

NOTE: IT IS RECOMMENDED THAT YOU DO NOT EXIT THE PROGRAM SIMPLY BY PRESSING BREAK. IF THIS DOES HAPPEN, YOU SHOULD PRESS SHIFT-CLEAR AND THEN ENTER TO EXIT CLEANLY BEFORE RETURNING TO THE DOS OR LOADING A NEW BASIC PROGRAM. OTHERWISE, SOONER OR LATER THE MACHINE WILL PROBABLY HANG.

ALL THE ONE-LINERS, EXCEPT FOR THE LAST TWO, WERE ORIGINALLY PUBLISHED IN SOFTSIDE MAGAZINE. THEY USUALLY APPEARED AS FILLER MATERIAL, OCCUPYING THE SPACE OF A SMALL ADVERTISEMENT. MY HIGH-SCHOOL COMPUTER PROGRAMMING TEACHER HAD A SMALL HOARD OF VARIOUS TRS-80 MAGAZINES, INCLUDING NUMEROUS SOFTSIDES, AND THIS IS HOW I FIRST BECAME ACQUAINTED WITH THE CHALLENGE OF WRITING ONE-LINERS. I IMMEDIATELY FELL IN LOVE WITH THE IDEA. (AFTER ALL, THE ONE-LINER MINDSET WAS LITTLE MORE THAN MY NORMAL PROGRAMMING MINDSET, BACK THEN, TAKEN TO ITS LOGICAL CONCLUSION.) I CAREFULLY PAGED THROUGH ALL OF MY TEACHER'S SOFTSIDES IN ORDER TO FIND ALL THE TRS-80 ONE-LINERS. INEVITABLY I EVENTUALLY DECIDED THAT I NEEDED TO COLLECT THEM ALL TOGETHER INTO A SINGLE PROGRAM.

IN ORDER TO PROVIDE A WAY TO ACCESS EACH ONE-LINER AS A SEPARATE PROGRAM, I NEEDED TO ADD TWO THINGS: A WAY FOR THE USER TO SELECT WHICH ONE-LINER TO RUN, AND A WAY TO CLEANLY EXIT A ONE-LINER WITHOUT USING BREAK, WHICH WOULD EXIT THE WHOLE PROGRAM.

THE FORMER PROBLEM WAS SOLVED WITH A SIMPLE ONE-LINER, LINE 1 OF THE PROGRAM. ALL IT DOES IS RESET SOME OF THE SHARED STATE OF THE BASIC INTERPRETER AND THEN DISPLAYS A PROMPT. IT IS NOTABLE ONLY FOR THE FACT THAT THE PROMPT IS TAILORED TO REMIND USERS OF THE OPENING PROMPT OF THE SCOTT ADAMS ADVENTURE COLLECTIONS.

THE LATTER PROBLEM WAS A BIT TRICKIER, AND I EVENTUALLY SOLVED IT BY WRITING A SHORT ASSEMBLY-LANGUAGE ROUTINE TO HOOK

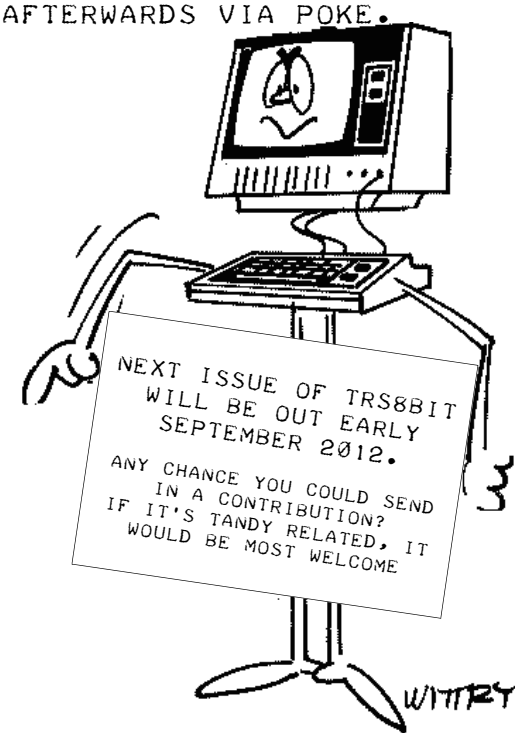
INTO THE KEYBOARD HANDLER. THE HOOK CODE SIMPLY LOOKS FOR THE SHIFT-CLEAR KEY COMBINATION, AND WHEN IT SEES IT IT FORCES THE BASIC INTERPRETER TO JUMP BACK TO LINE 1 OF THE PROGRAM. PROVIDING AND INSTALLING THE MACHINE-LANGUAGE CODE IS DONE IN LINE Ø OF THE PROGRAM. THE MACHINE-LANGUAGE BYTES ARE EMBEDDED IN A REM STATEMENT. I DIDN'T WANT TO USE THE STANDARD USR INTERFACE FOR INVOKING THE CODE, HOWEVER, OUT OF CONCERN THAT IT MIGHT SOMEHOW INTERFERE WITH A FUTURE ONE-LINER, SO INSTEAD I HIJACKED THE LITTLE-USED NAME COMMAND. IF YOU'RE CURIOUS TO UNDERSTAND HOW IT WORKS, I'VE INCLUDED A DETAILED EXPLANATION AT THE END OF THE PAGE.

HERE IS A LIST OF THE AVAILABLE ONE-LINERS:

1) A ONE-LINE ADVENTURE GAME [AUTHOR: PHILLIP CASE]
2) INSTALLING AND RUNNING AN ASSEMBLY PROGRAM
3) A PERSONALIZED CHRISTMAS GREETING
4) USING CONTROL CHARACTERS TO MAKE HIDDEN MESSAGES
5) A ONE-LINE DODGING GAME
6) LUNAR LANDER (ENTER YOUR X-VELOCITY AT START)
7) SYMMETRICAL LINE PATTERNS (PRESS R TO CLEAR SCREEN)
8) ANOTHER FORAY INTO TRS-8Ø ART IN ONE LINE
9) A ONE-LINE DRIVING GAME
A) RANDOM KALEIDOSCOPE PATTERNS
B) SIMPLE SHOOTING GALLERY (USE SPACE TO FIRE)
C) AN EXCURSION IN NON-ORTHOGONAL RANDOM LINES
D) GUESS MY NUMBER [AUTHORS: BRIAN RAITER, REX WHEELER]
E) A HEX-TO-DECIMAL CONVERTER [AUTHOR: BRIAN RAITER]

(MY PERSONAL FAVORITES ARE NUMBERS 1, 6, AND 7.)

THE FOLLOWING IS A LISTING, OR REALLY A REPRESENTATION OF THE LISTING, OF THE COMPLETE PROGRAM. IT IS PROVIDED FOR REFERENCE ONLY; IT'S MUCH LESS EFFORT TO USE THE PROVIDED LINK AT THE TOP OF THE PAGE TO OBTAIN A USABLE COPY OF THE PROGRAM. MANUALLY ENTERING ONE-LINERS IS OFTEN NOT A STRAIGHTFORWARD MATTER: THE INTERPRETER WOULD ONLY ACCEPT 255 BYTES IN A SINGLE LINE OF BASIC, SO ONE HAD TO OCCASIONALLY OMIT THE LAST FEW CHARACTERS WHEN ENTERING A LINE, AND THEN ADD THE REST USING THE EDIT COMMAND AFTER THE INTERPRETER HAD HAD A CHANCE TO PARSE THE INPUT. (EDIT WAS ALSO USED TO EMBED LINE BREAKS AND OTHER SPECIAL CHARACTERS IN STRING LITERALS.) AND OF COURSE, THE BYTES OF THE MACHINE-LANGUAGE ROUTINE IN LINE Ø HAD TO BE CREATED BY ENTERING FILLER CHARACTERS AND THEN MODIFYING THEM AFTERWARDS VIA POKE.



```
0 B=PEEK(16548)+256*PEEK(16549)+124:FORX=0TO2:POKEB+79+X*2,PEEK(
16782+X):NEXT:POKE16782,195:POKE16783,BAND255:POKE16784,B/256:NA
ME:REM * One-liner key hook
SQR*GET@EDITSQR MKS$K@SQRqSQRpSQR"@ SQR SQR"IFAINKEY$+?
```

```
1 CLEAR50:CLS:RANDOM:POKE16412,0:POKE16419,95:LINEINPUT"Press SH
IFT-CLEAR to return to the menu.
Which One-liner? (1-Z) ";L1$:IFL1$="",NAME:ENDELSEONASC(L1$)-48G
OTO2,4,5,6,8,9,11,12,13,1,1,1,1,1,1,1,15,16,17,18,19:GOTO1:REM *
 One-liner menu
```

```
2 CLS:PRINT"Adventure
Visible Items: Trees
Obvious Exits: EAST WEST
"STRING$(30,45):INPUT"Tell me";A$:IFA$<>"E",2ELSEPRINT@0,"I'm in
 a cave."@79,"Nothing";:PRINT@143,"None",;:PRINT@263,"";:INPUTA$
:IFASC(A$)=72ANDMID$(A$,4)="P",PRINT"You win!!"ELSE2
```

```
4 CLEAR22:A$=STRING$(22,32):J=VARPTR(A$):I=PEEK(J+1)+256*PEEK(J+
2):I=I+65536*(I>32767):FORK=ITOI+21:READZ:POKEK,Z:NEXT:DEFUSR=I:
FORX=1TO2:POKEI+10,RND(63)+128:L=USR(0):X=1:NEXT:DATA33,0,60,17,
1,60,1,255,3,54,0,237,176,6,5,33,0,0,43,124,181,201
```

```
5 INPUT"ENTER YOUR NAME";A$:A$=A$+" ":L=LEN(A$):P=30:CLS:PRINT@P
+1,"**":FORI=1TOL:PRINT@P-I+64*I,"* ";:PRINTMID$(A$,1,I);:PRINTM
ID$(A$,1,I)+" *":NEXT:PRINTTAB(P)"MERRY":PRINTTAB(P);"XMAS!":FOR
T=1TO5000:NEXT:PRINTTAB(P-11)"FROM SOFTSIDE PUBLICATIONS":RUN5
```

```
6 CLEAR999:CLS:PRINT"M A G I C   C O D E":PRINT"E N T E R   M S
G":INPUTA$:PRINTCHR$(28):PRINTCHR$(31):X=LEN(A$):FORY=1TOX:B$=B$
+MID$(A$,Y,1):C$=CHR$(RND(58)+31):B$=B$+C$:NEXT:PRINT"E N T E R"
:INPUTD$:PRINTB$:PRINT:PRINT"E N T E R":INPUTD$:PRINTCHR$(23)
```

```
8 P=28:C$="V":FORT=1TO9E9:Q=RND(12):IFT/4=INT(T/4)PRINT@916+Q,"O
":NEXTELSEA$="I             I":PRINT@P,C$@980,A$@1023,:IFPEEK(154
24+P)=79ORP>32ORP<21PRINT@P,"BOOM";:FORW=1TO9:NEXT:RUN8ELSEK=PEE
K(14400):IFK=32P=P-1:NEXTELSEIFK=64P=P+1:NEXTELSENEXT
```

```
9 INPUTH:CLS:G=42:FORT=0TO127:G=G+RND(3+(G=47))-2:FORD=GTO47:SET
(T,D):NEXTD,T:FORF=0TO99-H*5:P=PEEK(14400):H=H+(P=32)/2-(P=64)/2
:V=V+.25+(P=8)/2:RESET(X,Y):X=X+H:Y=Y+V:IFPOINT(X,Y)IFV>10RHRESE
T(X,Y):RUN9ELSESET(X,Y-1):PRINT@349,"Landed"ELSESET(X,Y):NEXT
```

```
11 CLS:DEFINTA-Z:RANDOM:X=32:Y=12:FORK=1TO2STEP0:IFINKEY$="R"THE
N11ELSEL=RND(25):XD=RND(3)-2:YD=RND(3)-2:FORA=1TOL:SET(X,Y):SET(
127-X,Y):SET(127-X,47-Y):SET(X,47-Y):X=X+XD:Y=Y+YD:X=X-128*INT(X
/128):Y=Y-48*INT(Y/48):NEXTA,K:REM "R"=RESTART
```

```
12 CLS:PRINT@402,"C O M P U T E R  A R T  ! !":PRINT:DEFINTA-Z:R
ANDOM:FORT=1TO1000:NEXTT:CLS:FORT=1TO10:A=RND(62)-1:B=RND(62)+61
:C=RND(22)-1:D=RND(22)+21:FORX=ATOB:SET(X,D):SET(X,C):NEXTX:FORY
=CTOD:SET(B,Y):SET(A,Y):NEXTY,T:FORT=1TO3000:NEXTT:GOTO12
```

```
13 CLS:J=27:P=15391:FORM=1TO9E9:T=RND(99):FORL=RND(T)TOTSTEPRND(
9)/9:J=50-ABS(ABS(J+SIN(L))-50):A=PEEK(14400)/32:P=P-((AAND1)-(A
/2AND1))*(PEEK(14464)+1):PRINTTAB(J)"!!!   .   !!!":IFPEEK(P)=33
PRINT"SCORE:"SELSEPOKEP,191:S=S+1:NEXTL,M
```

```
15 CLS:PRINT"Hit space bar to see a new pattern":FORX=1TO700:NEX
TX:FORT=1TO5000:CLS:FORZ=1TO1160:X=RND(64)-1:Y=RND(24)-1:SET(X,Y
):SET(X,47-Y):SET(127-X,47-Y):SET(127-X,Y):IFINKEY$=""THENNEXTZ,
T:GOTO15ELSENEXTT:GOTO15
```

```
16 CLS:CLEAR99:C=RND(62)+33:PRINTSTRING$(63,C):FORX=0TO1E9:IFINK
EY$="",NEXTELSEFORH=0TO1E9:R=RND(63)-1:IFPEEK(15360+R)<>C,NEXTEL
SEFORL=960+RTORSTEP-64:PRINT@L,"!";:PRINT@L," ";:NEXT:FORL=0TO9:
PRINT@R,CHR$(RND(159)+32);:NEXT:PRINT@R," ";:NEXTXELSENEXTX
```

```
17 CLS:DEFINTA-Z:X=RND(127):Y=RND(47):FORA=0TO10+RND(60):I=RND(7
)-4:J=RND(5)-3:FORB=0TORND(30):SET(X,Y):I=I+2*I*((I+X>127)OR(I+X
<0)):X=X+I:J=J+J*2*((J+Y>47)OR(J+Y<0)):Y=Y+J:NEXT:NEXT:FORI=0TO9
99:NEXT:RUN17
```

```
18 CLS:RANDOM:A=1:B=1E3:Q=RND(1E3):FORX=1TO8E9:PRINTA"-"B:INPUT"
 Guess";C:IFC<AORC>B,X=X-1:NEXTELSEIFC<QPRINT"Too low":A=C+1:NEX
TELSEIFC>QPRINT"Too high":B=C-1:NEXTELSEPRINTC"is right!!
 It took you"X"tries.
":X=9E9:NEXT:FORX=0TO0:X=INKEY$="":NEXT:RUN18
```

```
19 PRINT"Hex: --"STRING$(2,24);:FORA=1TO0STEP-1:FORQ=0TO0:A$=INK
EY$:Q=A$="":NEXT:B(A)=ASC(A$):IFB(A)>47ANDB(A)<58ORB(A)>64ANDB(A
)<71THENPRINTCHR$(B(A));:NEXT:FORA=0TO1:H=H+((-B(A)+55)*(B(A)>64
)+VAL(CHR$(B(A))))*16↑A:NEXT:PRINT"
Dec"H:RUN19ELSEA=A+1:NEXT
```

ONCE AGAIN, TO BE CLEAR: I AM NOT THE AUTHOR OF THE MAJORITY OF THESE PROGRAM.
I AM MERELY THE ANTHOLOGIST.
THE ORIGINAL AUTHORS AND/OR COPYRIGHT HOLDERS CONTINUE TO OWN AND RESERVE ALL RIGHTS TO THIS CODE. BEING THE CALLOW YOUTH THAT I WAS AT THE TIME, I NEGLECTED TO WRITE DOWN THE AUTHORS' NAMES (THOUGH I THINK SOFTSIDE ALSO OFTEN NEGLECTED TO PROVIDE THEM).
IN ANY CASE, I WELCOME INFORMATION FROM ANYONE WHO CAN INDICATE WHERE CREDIT IS DUE.

PLEASE CONTACT ME AT **BREADBOX@MUPPETLABS.COM.**

APPENDIX: HOW LINE 0 WORKS

THE BASIC PART OF LINE 0 IS STRAIGHTFORWARD. THE FIRST STATEMENT:

```
B=PEEK(16548)+256*PEEK(16549)+124
```

INITIALIZES B WITH THE ADDRESS OF THE BASIC PROGRAM, PLUS 124 BYTES, WHICH IS THE OFFSET OF THE MACHINE-LANGUAGE ROUTINE WITHIN THE REM. NEXT COMES A SHORT LOOP:

```
FORX=0TO2:POKEB+79+X*2,PEEK(16782+X):NEXT
```

WHICH COPIES THE EXISTING NAME VECTOR TO A SAFE PLACE INSIDE THE MACHINE-LANGUAGE ROUTINES (IN THE BYTES CONTAINING QUESTION MARKS BELOW). ONCE THESE BYTES HAVE BEEN SAVED, THE PROGRAM CAN THEN REPLACE THEM:

```
POKE16782,195:POKE16783,BAND255:POKE16784,B/256
```

WITH A VECTOR THAT JUMPS TO THE MACHINE-LANGUAGE ROUTINE. WITH THIS IN PLACE, THE BASIC PROGRAM CAN THEN USE THE STATEMENT: 'NAME' TO JUMP TO THE MACHINE-LANGUAGE ROUTINE.

THE MACHINE-LANGUAGE CODE, BESIDES BEING NECESSARY TO HOOK THE KEYBOARD, ALSO ALLOWS MUCH MORE FUNCTIONALITY TO BE PACKED INTO A SMALL NUMBER OF BYTES. TURNING TO THE ORIGINAL ASSEMBLY, THE MACHINE LANGUAGE ROUTINE AT OFFSET 124 (007CH) FROM THE START OF THE BASIC PROGRAM IS AS FOLLOWS:

```
                    ; THE HOOK INSTALLER ROUTINE
007C  DD2AA440  SETUP:  LD   IX,(40A4H)       ; IX=LINE 0 POS
0080  119D01            LD   DE,0100H+HOOK    ; DE=OFFSET OF
0083  15                DEC  D                ;   HOOK RTN
0084  DD19              ADD  X,DE             ; ADVANCE IX
0086  ED4B1640          LD   BC,(4016H)       ; BC=KEYBD RTN
008A  DD7101            LD   (IX+1),C         ; STORE IN CALL
008D  DD7002            LD   (IX+2),B         ; INSTRUCTION
0090  DD221640          LD   (4016H),IX       ; REPLACE W/IX
0094  1E19              LD   E,REMOVE-HOOK    ; CHANGE NAME
0096  DD19              ADD  IX,DE            ; VECTOR TO
0098  DD228F41          LD   (418FH),IX       ; REMOVE RTN
009C  C9                RET                   ; SETUP IS DONE
```

4016H CONTAINS THE ADDRESS OF THE CURRENT KEYBOARD HANDLING ROUTINE. THE SETUP ROUTINE RETRIEVES THAT ADDRESS AND STORES IT INSIDE THE HOOK ROUTINE, WHICH IMMEDIATELY FOLLOWS THIS ONE. IT THEN REPLACES THAT VALUE WITH THE ADDRESS OF THE START OF THE HOOK ROUTINE, THUS INSTALLING THE KEYBOARD HOOK. IT ALSO MODIFIES THE VALUE OF THE NAME VECTOR, SO THAT INSTEAD OF POINTING TO THIS ROUTINE, IT WILL POINT TO REMOVE (DESCRIBED BELOW). AFTER THIS, A SUBSEQUENT NAME STATEMENT WILL CALL REMOVE INSTEAD OF SETUP.

THE KEYBOARD HOOK ROUTINE IS AS FOLLOWS:

```
                    ; THE KEYBOARD HOOK
009D  CD3F3F    HOOK:   CALL 3F3FH            ; KEYBD HANDLER
00A0  FE1F              CP   31               ; CLEAR KEY?
00A2  C0                RET  NZ               ; NO, LET PASS
00A3  3A8038            LD   A,(3880H)        ; A SHIFT KEY
00A6  B7                OR   A                ;   PRESSED?
00A7  3E1F              LD   A,31             ; (RESTORE ACC)
00A9  C8                RET  Z                ; NO, LET PASS
00AA  E1                POP  HL               ; YES, GRAB IT
00AB  2AA440            LD   HL,(40A4H)       ; HL=LINE 0 POS
00AE  01D801            LD   BC,0100H+NEXT    ; ADVANCE HL TO
00B1  05                DEC  B                ;   NEXT LINE
00B2  09                ADD  HL,BC            ;   OF PROGRAM
00B3  C31E1D            JP   1D1EH            ; BACK TO BASIC
```

THE ADDRESS OF THE ORIGINAL KEYBOARD ROUTINE IS STORED AS PART OF THE CALL INSTRUCTION AT TOP, REPLACING THE ORIGINAL VALUE OF 3F3FH (OR "??" IN ASCII). THUS THE HOOK ROUTINE FIRST INVOKES THE NORMAL KEYBOARD HANDLER. WHEN IT RETURNS, THE ACCUMULATOR HOLDS THE DECODED CHARACTER. ASCII 31 REPRESENTS THE CLEAR KEY, SO IF THE ACCUMULATOR HOLDS SOME OTHER VALUE, THE ROUTINE RETURNS DIRECTLY. OTHERWISE THE BYTE AT 3880H IS EXAMINED, AND IF IT IS NONZERO THEN A SHIFT KEY IS CURRENTLY DOWN. IF SO, THE HL REGISTER IS ALTERED TO POINT TO THE SECOND LINE OF THE CURRENT BASIC PROGRAM, AND THE HOOK ROUTINE JUMPS DIRECTLY BACK TO THE BASIC INTERPRETER INSTEAD OF RETURNING.

FINALLY, THE REMOVE ROUTINE, INVOKED WHEN A SECOND NAME STATEMENT IS ENCOUNTERED, IS AS FOLLOWS:

```
                    ; THE HOOK UNINSTALLER ROUTINE
00B6  DD2AA440  REMOVE: LD   IX,(40A4H)      ; IX=LINE 0 POS
00BA  119D01            LD   DE,0100H+HOOK   ; DE=OFFSET OF
00BD  15                DEC  D               ; HOOK RTN
00BE  DD19              ADD  IX,DE           ; ADVANCE IX
00C0  DD5E01            LD   E,(IX+1)        ; GET ORIGINAL
00C3  DD5602            LD   D,(IX+2)        ; KEYBD RTN
00C6  ED531640          LD   (4016H),DE      ; RESTORE IT
00CA  3E3F              LD   A,'?'           ; RESTORE THE
00CC  1E3F              LD   E,'?'           ; ORIGINAL
00CE  163F              LD   D,'?'           ; NAME VECTOR
00D0  328E41            LD   (418EH),A       ; PUT HERE BY
00D3  ED538F41          LD   (418FH),DE      ; BASIC PROG
00D7  C9                RET                  ; WE'RE DONE
                  NEXT:
```

THIS ROUTINE SIMPLY RESTORES THE VALUES THAT 4016H AND 418EH ORIGINALLY HAD. THE ORIGINAL VALUE FOR 4016H IS STORED AS PART OF THE CALL INSTRUCTION AT THE TOP OF THE HOOK ROUTINE, WHILE THE ORIGINAL VECTOR VALUE FOR 418EH WAS STORED INSIDE THIS ROUTINE DIRECTLY, REPLACING THE ASCII QUESTION MARK BYTES, AS PART OF THE SETUP ROUTINE. THIS ROUTINE IS INVOKED IN LINE 1 OF THE BASIC PROGRAM, IN THE CASE WHEN THE USER INPUTS AN EMPTY STRING, JUST BEFORE THE FINAL END STATEMENT.

BRIAN RAITER
MUPPETLABS

Nothing is safe from

THE PIRATE

The ULTIMATE program copier!



Though it is not Kansas policy to encourage the copying of programs, the sheer unreliability of imported tapes as prompted many software users to ask us to do something about it.

The problem with the imported tapes is that by the time they eventually reach the user they are in an advanced state of corruption, and with mostly just a single recording, it is essential that a back-up be made.

But here's the crunch--these programs are heavily protected! Which means that no known copy program or Monitor can copy them.

Pirate will.

It does not matter how heavily the program is protected or in fact **how** it is protected. It doesn't matter if it is in machine language with a pre-loader or one of the latest with two separate programs, one loading the other. It can even be a Basic program with a machine language loader attached--**Pirate will copy them all.**

For Pirate differs from every other known copy utility or Monitor, in that it completely ignores the actual program on the tape, but reproduces whatever is there as pulses.

Which means that whatever was on the original tape will be stored in memory then faithfully reproduced onto another tape--protection and all!

The only criteria is that there must be space for both programs, though Pirate takes up very little user RAM.

For Video Genie and Tandy Model I. Available only from Kansas. £14.50 Vat and post paid.

Kansas

As pulishers (we do not retail other people's programs) we have absolute quality control and thus give an unconditional guarantee on all our software. And of course all programs are always in stock for our famed return first class post service whether ordered by cheque or credit card. Ask for a copy of our free catalogue, you won't find any 'South Coast' prices there! Access and Barclaycard welcome.

Kansas City Systems, Unit 3, Sutton Springs Wood, Chesterfield, S44 5XF. Tel. 0246 850357

ONE-LINERS
FOR TRS-80 BASIC (PART 2)

BRIAN RAITER

DUSTY

OKAY, SO I WROTE A VERY SIMPLE ONE-LINER THIS WEEK. UNFORTUNATELY, MY TRS-80 EMULATOR DOESN'T DO TIMING ADJUSTMENT PROPERLY, SO EVERYTHING RUNS AT 21ST-CENTURY SPEEDS. AS A RESULT I'VE NO IDEA HOW IT LOOKS ON AN ACTUAL MACHINE. NONETHELESS, HERE IT IS:

```
0 TWIRL
1 DIMP(112):F=3:FORZ=0TO1STEP
0:F=F+.03:FORT=0TO6:M(T)=TAN(
T*.448799+F):NEXT:N=0:FORY=0
TO15:FORT=0TO6:X=3*(Y-8)/M(T)
+32:P(N)=Y*64+X:N=N-(X>=0ANDX
<63):NEXT:NEXT:CLS:FORK=0TON-
1:PRINT@P(K),".";:NEXT:NEXT
```

(I'VE ALSO PLACED IT WITH THE FEATURED NEW ONES - DUSTY)

IF IT MOVES TOO SLOWLY, YOU COULD PROBABLY IMPROVE ITS APPEARANCE BY BUMPING UP THE INCREMENT VALUE AT F=F+.03. IF IT'S TOO FLICKERY, HOWEVER, THAT PROBABLY CAN'T BE FIXED WITHOUT A MAJOR REWRITE. (I HAD AN ALTERNATE VERSION THAT DID THE REDRAW WITHOUT USING CLS, BUT MY BENCHMARK SUGGESTED THAT IT WAS A LOT SLOWER, AND IT WAS BUGGY TO BOOT.) ANYWAY, FEEL FREE TO USE IT OR NOT.

MORE INTERESTING, PERHAPS, IS THAT WHILE WORKING ON THIS, I HAD TO GO POKING THROUGH SOME OF MY OLD SOFTWARE, AND I CHANCED UPON A LISTING OF SEVERAL MORE ONE-LINERS. IT'S WEIRD, BECAUSE I DON'T REMEMBER SEEING THESE BEFORE. OF COURSE, IT'S BEEN OVER 25 YEARS SINCE I HAD A WORKING TRS-80, SO THE FACT THAT I DON'T REMEMBER THEM MAY NOT BE PARTICULARLY SIGNIFICANT. BUT I HAVE ABSOLUTELY NO IDEA WHERE THEY CAME FROM.
THE LABELS AT THE BOTTOM ARE MINE, BUT THE PROGRAMS THEMSELVES...? THEY MIGHT BE MORE SOFTSIDE ONE-LINERS, BUT THEN WHY DIDN'T I ADD THEM TO MY PROGRAM? I MEAN, I SPENT HOURS LOOKING THROUGH THOSE MAGAZINES AS A KID, TRYING TO FIND MORE ONE-LINERS FOR MY COLLECTION. ON THE OTHER HAND, I DON'T REMEMBER EVER FINDING ANOTHER OTHER SOURCE OF ONE-LINERS. HECK, FOR ALL I KNOW I WROTE THEM ALL MYSELF. (I DOUBT THAT, THOUGH; I REMEMBER WHAT MY TRS-80 PROGRAMMING STYLE LOOKED LIKE, AND NONE OF THESE REALLY MATCH THAT.)

HERE'S THE LISTING, MORE OR LESS AS I FOUND IT ON MY HARD DRIVE:

```
1 A$=INKEY$:IFA$=""THEN1 ELSEZ=ASC(A$):Z=32*((Z>96)-(Z>64 ANDZ<9
1))+Z:PRINTCHR$(Z);:LPRINTCHR$(Z);:GOTO1

2 RANDOM:DEFINTA-Z:CLS:FORI=1TO10:R=RND(20)+2:FORA=-RTOR:X=R*R-A
*A:Y=INT(SQR(X)-.5):SET(2*A+60,23+Y):SET(2*A+60,23-Y):NEXT:NEXT:
FORI=1TO999:NEXT:RUN2

3 RANDOM:IFG=0CLS:G=1:GOTO3ELSEF=47:X=RND(64):Y=RND(F):K=127:FOR
I=TOY:IFINKEY$=" "THENRUN3ELSEX=-(X
                ** THIS ONE SEEMS TO BE INCOMPLETE **
```

```
4 CLEAR999:RANDOM:DEFSTRA:DEFINTB-Z:X=7:FORJ=ØTO9999:A="":FORI=1
TO4:A=A+CHR$(RND(63)+128):NEXT:A=CHR$(174)+A+A+A+A+A+A+A+A+A+A+A
+A+CHR$(157):Y=X+(RND(3)-2):IFY<ØORY>13PRINT@96Ø+X,A:NEXT:RUN4EL
SEX=Y:PRINT@96Ø+X,A:NEXT:RUN4

5 CLEAR999:DEFSTRA-Z:A=STRING$(12,191):B=STRING$(12,24):C=CHR$(2
6):D=STRING$(3,191):E=" ":F=A+C+B+D+E+D+C+B+D+E+D+C+B+A:CLS:PRIN
T@(RND(52)-1)+(RND(12)-1)*64,F;:GOTO5

6 CLS:CLEAR99:DEFINTA-Z:DIMZ(32):RANDOM:W=15360:V=16320:X(1)=179
:X(2)=128:X(3)=191:X(4)=140:FORJ=1TO8:X=RND(4):FORI=ØTO31:Z(I)=X
(RND(X)):NEXT:FORI=ØTO31:POKEW+I,Z(I):POKEV+I,Z(I):POKEW+63-I,Z(
I):POKEV+63-I,Z(I):NEXT:W=W+64:V=V-64:NEXT:FORI=1TO2999:NEXT:RUN6

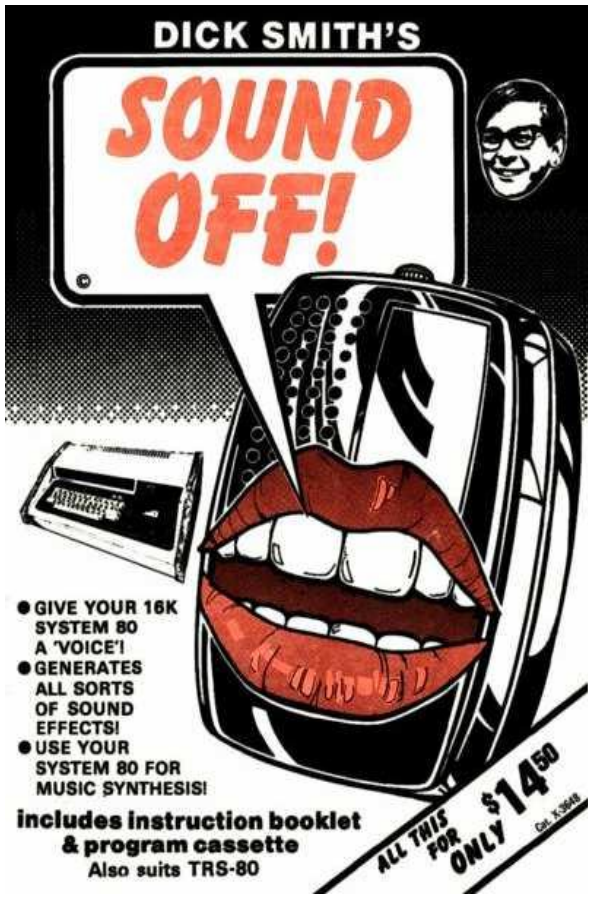
7 CLEAR184:CLS:S=RND(7):P=9:L=15:DEFSTRA-K:G=CHR$(191):D=CHR$(24
)+CHR$(26):E=G+D:F=E+E+E:FORZ=1TO7:A=STRING$(L*3,131):B=STRING$(
L*3,176):L=L-2:C="":FORX=1TOL:C=C+E:NEXT:C=C+G:PRINT@P,C;B;:PRIN
T@P,A;C;:PRINT@P,G;:P=P+63+S:NEXT:FORY=1TO999:NEXT:GOTO7

8 RANDOM:DEFINTA-Z:FORI=15360TO16383:POKEI,128:NEXT:FORI=1TO1Ø:R
=RND(2Ø)+2:FORA=-RTOR:X=R*R-A*A:Y=INT(SQR(X)-.5):SET(2*A+6Ø,23+Y
):SET(2*A+6Ø,23-Y):NEXT:NEXT:FORI=1TO999:NEXT:FORI=1TO2:FORJ=153
6ØTO16383:X=PEEK(J):POKEJ,(NOTX)+64AND255:NEXT:NEXT:RUN8
```

```
1Ø REM LIST OF PROGRAMS
11 REM 1: LINE PRINTER
12 REM 2: RANDOM CIRCLES
13 REM 3: INVERTING
          KALEIDOSCOPE
14 REM 4: WAVING TAPESTRY
15 REM 5: JUMPING SQUARE
16 REM 6: ANOTHER
          KALEIDOSCOPE
17 REM 7: HALLWAY EFFECT
18 REM 8: RANDOM FLASHING
          CIRCLES
```

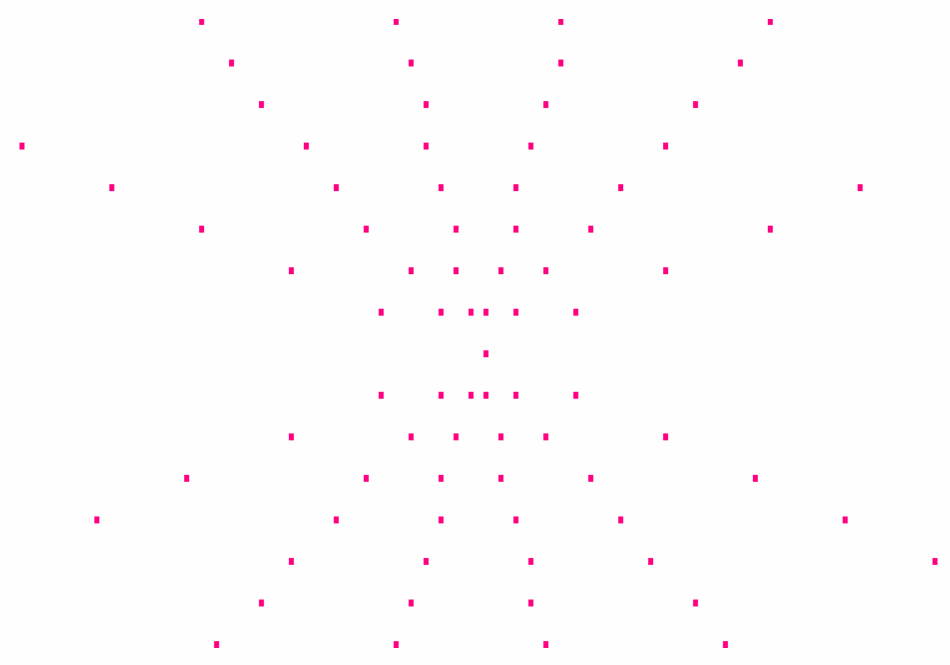
I'M ONCE AGAIN FRUSTRATED THAT I CAN'T GIVE YOU PROPER PROVENANCE FOR THESE PROGRAMS ... BUT AGAIN, FEEL FREE TO MAKE USE OF THEM AS YOU SEE FIT.

BRIAN






```
1 DIMP(112):F=3:FORZ=0TO1STEP0:F=F+.03:FORT=0TO6:M(T)=TAN(T*.448
799+F):NEXT:N=0:FORY=0TO15:FORT=0TO6:X=3*(Y-8)/M(T)+32:P(N)=Y*64
+X:N=N-(X)=0ANDX(63):NEXT:NEXT:CLS:FORK=0TON-1:PRINT@P(K),".";:N
EXT:NEXT
```



HI DUSTY, I USE THIS ONE-LINE PROGRAM ALL THE TIME ON MODEL 1'S. IT SAVES YOU FROM OPENING UP THE COMPUTER TO DETERMINE IF IT HAS LOWERCASE OR NOT, AND WHICH TYPE OF LOWERCASE MOD IS FITTED.

WHAT IT DOES IS PRINTS THE COMPLETE CHARACTER SET ON THE SCREEN WITHOUT NEEDING TO LOAD A LOWER CASE DRIVER FROM TAPE:



WHEN YOU RUN THE PROGRAM YOU CAN DETERMINE WHICH CHARACTER GENERATOR AND LOWERCASE MOD (IF FITTED) IS INSTALLED:

1. ALL CAPITALS = NO LOWERCASE MODIFICATION INSTALLED

2. UPPER/LOWER CASE WITH ONE-PIXEL DECENDERS: RADIO SHACK LOWER CASE WITH 6673 CHARACTER GENERATOR

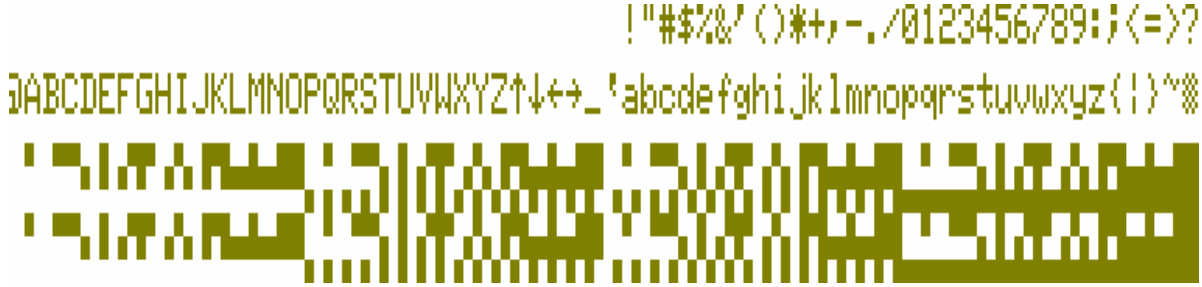
3. UPPER/LOWER CASE WITH NO DESCENDERS AND FLYING 'A': DB KITSZ LOWERCASE MOD UTILISING THE ORIGINAL CHARACTER GENERATOR 3001

4. UPPER/LOWER CASE WITH NO DESCENDERS WITH NORMAL 'A': DB KITSZ LOWERCASE MOD UTILISING THE 6670 CHARACTER GENERATOR

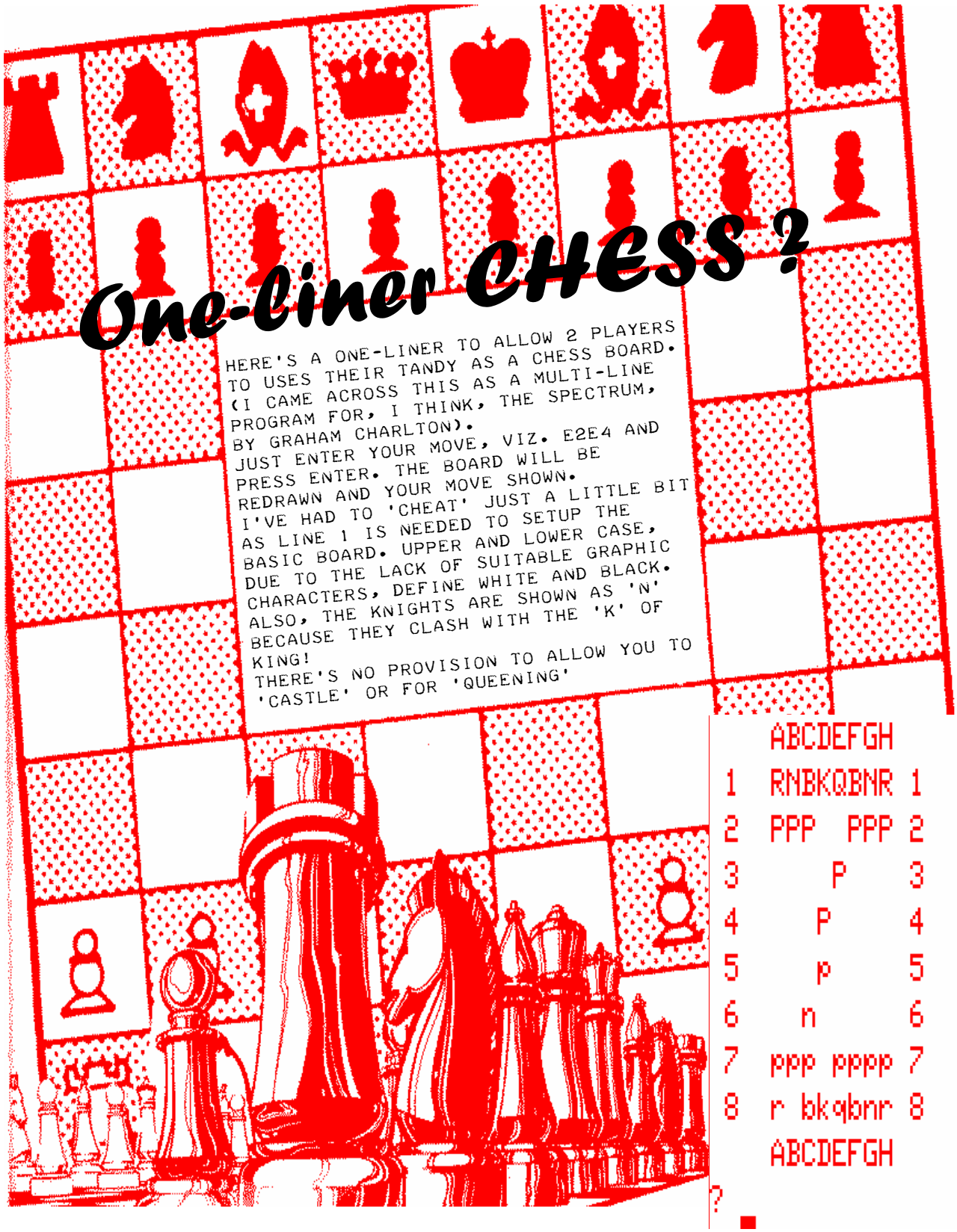
5. UPPER/LOWER CASE WITH 3-PIXEL DESCENDERS: DON MCKENZIE LOWERCASE MOD UTILISING AN EPROM WITH NEW CHARACTER SET IN IT

MAV.

```
10 CLS:FORT=32TO255:POKET+15360,T:NEXT:PRINT@500," "
```



One-liner CHESS ?




```
1 CLEAR500:A$="    ABCDEFGH":B$="RNBKQBNRPPPPPPPP"+STRING$(32,"
")+"pppppppprnbkqbnr"
2 FORY=0TO1STEP0:PRINTA$:FORZ=1TO8:PRINTZ;" ";MID$(B$,(Z*8-7),8)
;Z:NEXT:PRINTA$:INPUTM$:A=ASC(M$)-64:B=VAL(MID$(M$,2,1)):C=ASC(M
ID$(M$,3,1))-64:D=VAL(MID$(M$,4,1)):E=(B-1)*8+A:F=(D-1)*8+C:MID$
(B$,F,1)=MID$(B$,E,1):MID$(B$,E,1)=" ":NEXTY
```

Snail Run -



WHO SAID ONE-LINERS AREN'T MUCH USE FOR ANYTHING? HERE'S A ONE-LINER WHICH IS GREAT FOR 'WINNING' SMARTIES FROM YOUR GRAND KIDS, IF NOTHING ELSE. JUST GET 'EM BETTING ON WHICH 'SNAIL' WILL WIN!

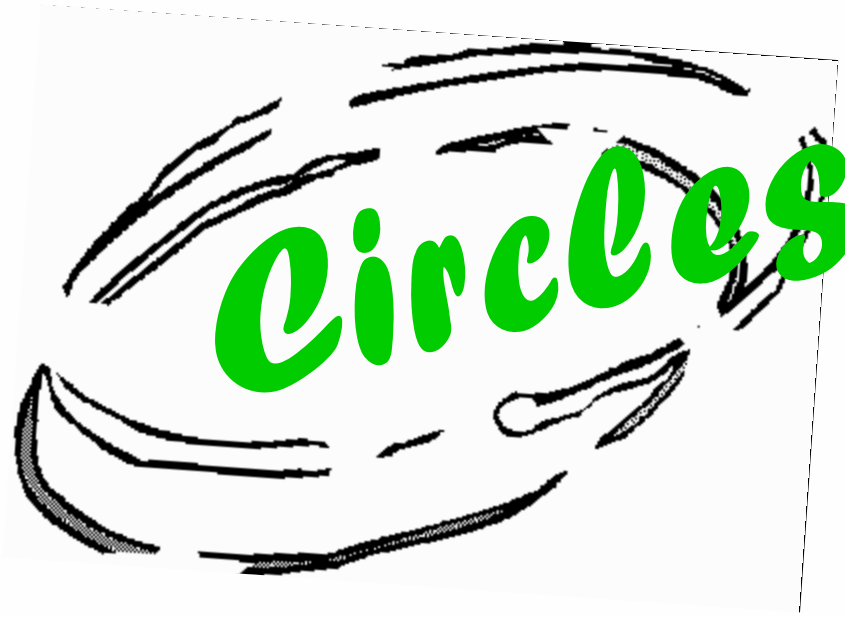
JUST REMEMBER WHEN TYPING IT IN, THAT THE 'NEW LINES' WITHIN THE PRINT STATEMENTS ARE DOWN ARROWS AND NOT THE 'ENTER' KEY. JUST TYPE RUN EACH TIME YOU WANT TO PLAY.

ON THE SCREEN PRINT BELOW, WOULD YOU PUT YOUR SHIRT ON SNAIL NUMBER 2?

```
5 RANDOM:CLS:PRINT"

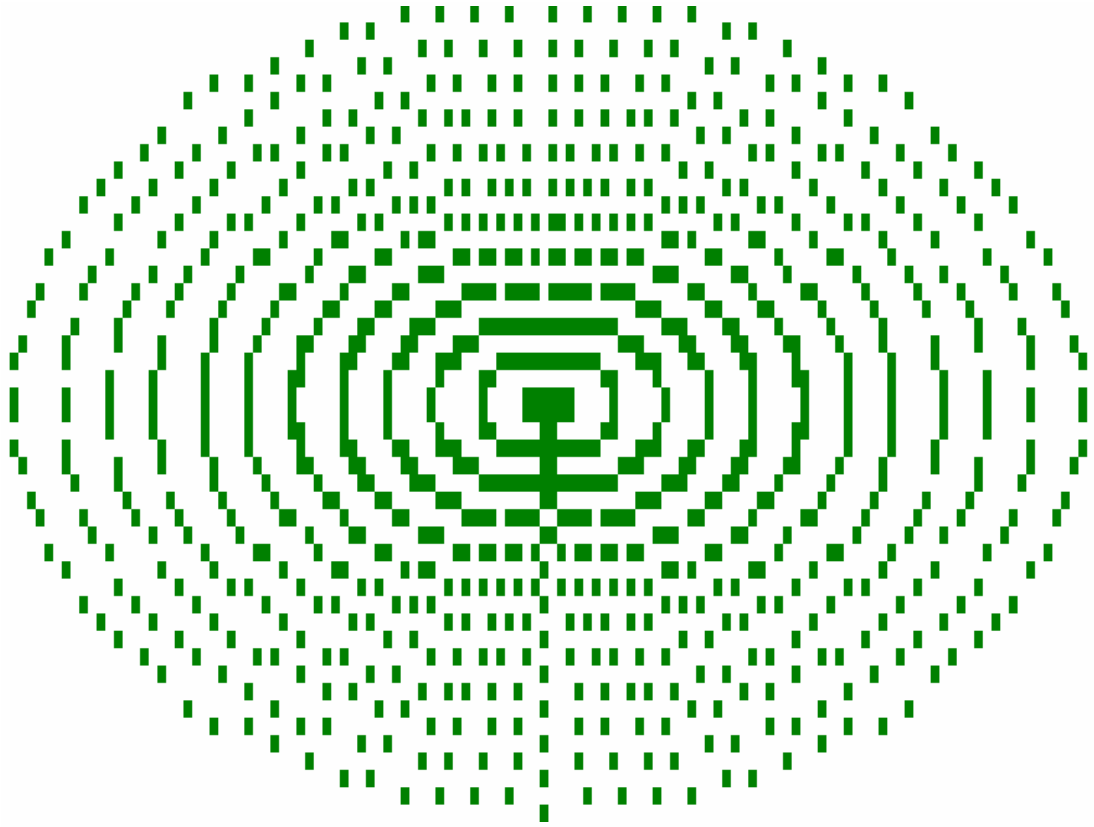
1 @
2 @
3 @
4 @
5 @
6 @
7 @
8 @";:FORX=1TO8:P(X)=X*64+15425:POKEX*64+15487,191:NEXT:FORZ=0TO
1STEP0:A=RND(8):P(A)=P(A)+1:POKEP(A)-1,32:POKEP(A),64:IFPEEK(P(A
)+1)=191THENPRINT"
WINNER IS NO.";AELSENEXT
```

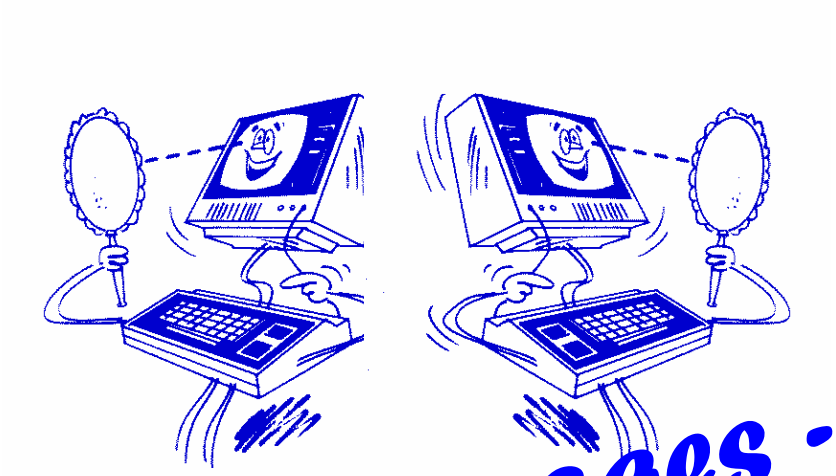




AGAIN, THIS IS STRAIGHT FORWARD. TYPE IT IN AND JUST ENTER 'RUN'. A SAMPLE SCREEN PRINT IS SHOWN BELOW.

```
10 CLS:Y=RND(23):X=RND(99):Z=RND(23):P=3.14159:FORA=0TO1STEP0:FO
RB=0TO2*PSTEPP/X:I=63+Y*(2.6667*SIN(B)):IFI<0THENRUNELSEN=23+Y*C
OS(B):IFN<0THENRUNELSESET(I,N):NEXT:Y=Y-1-Z:NEXT:NEXT
```

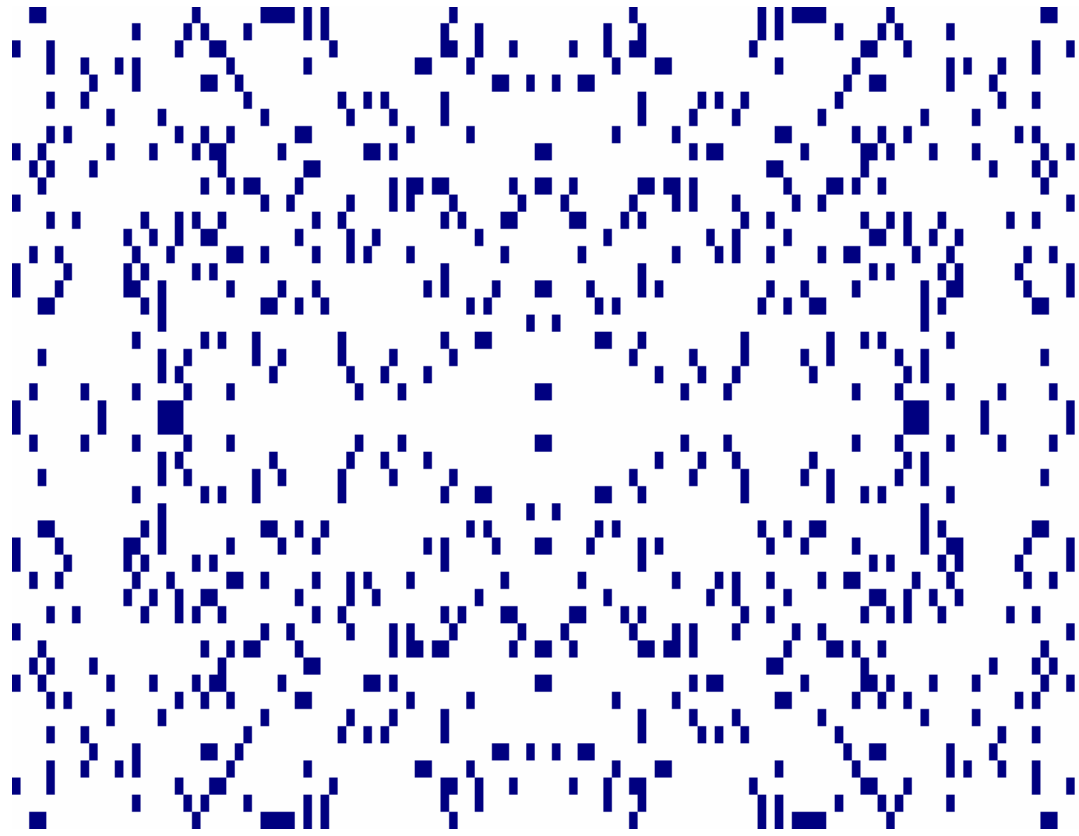


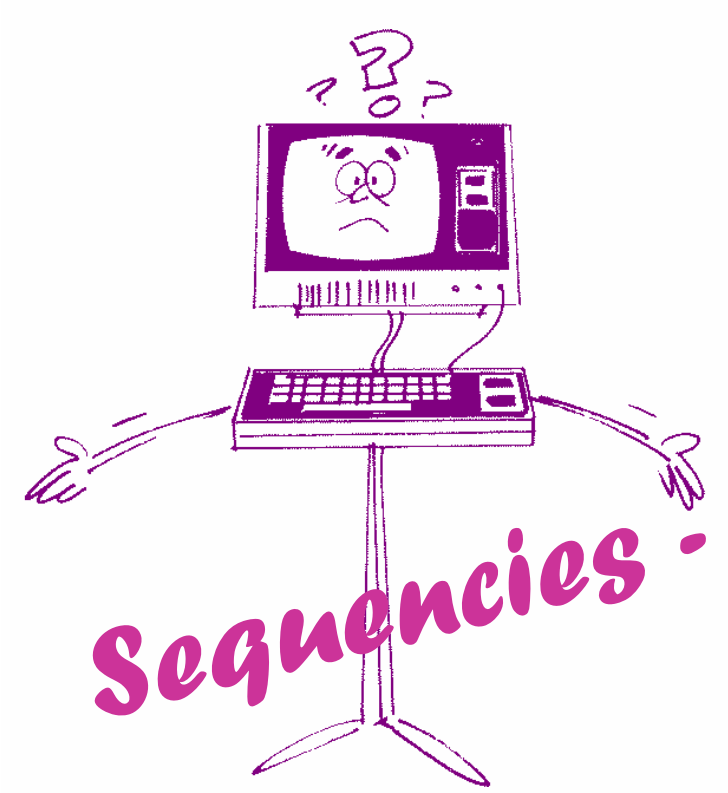


Mirror images -

HERE'S ONE THAT'S JUST AS THE TITLE SUGGESTS. I'LL BET IT COULD BE MOST IMPRESSIVE WITH HIRES GRAPHICS.
NOW THERE'S A THOUGHT ...

```
0 CLS:FORL=0TO1STEP0:A=RND(3):X=RND(62)-1:X1=62+(62-X)-1:Y=RND(2
4)-1:Y1=24+(24-Y)-1:IFA=1THENSET(X,Y):SET(X1,Y1):SET(X,Y1):SET(X
1,Y):NEXTLELSERESET(X,Y):RESET(X1,Y1):RESET(X,Y1):RESET(X1,Y):NE
XTL
```





HI DUSTY, HERE'S A LITTLE ONE-LINER FOR THE NEXT EDITION OF TRS8BIT. IT TRIES TO GUESS THE SEQUENCE OF A GROUP OF NUMBERS WHICH YOU ENTER. THE FIBINACCHI SERIES CATCHES IT OUT BUT FOR A ONE-LINER, I THINK IT MANAGES TO HIT THE SPOT.
KEEP UP THE GOOD WORK WITH TRS8BIT!

KIND REGARDS
E.T. FONEHUME

```
1 CLS:PRINT:PRINT:PRINT"ENTER 5 NUMBERS IN SEQUENCE & I'LL TRY A
ND GUESS THE NEXT NUMBER":FORZ=1TO5:INPUTX(Z):NEXT:X(6)=(X(1)*-2
)+(X(2)*5)+(X(3)*-2)+(X(4)*-4)+(X(5)*4):PRINT"NEXT NUMBER IS ";X
(6)
```

```
ENTER 5 NUMBERS IN SEQUENCE & I'LL TRY AND GUESS THE NEXT NUMBER

? 4
? 9
? 16
? 25
? 36
NEXT NUMBER IS  49
```



A piece of pi ?

(I PREFER CUSTARD ON MINE!)

HERE'S A ONE-LINER THAT DIVIDES ONE NUMBER BY ANOTHER. FOR INSTANCE, TO TRY PI, JUST TYPE 22,7 AND PRESS ENTER. IT SOON REPEATS ON ITSELF, (LIKE ALL GOOD PIES!), SO TRY 21.9999,6.9999. AND OFF IT GOES, NO NEED TO WORRY ABOUT SCREEN BURN EH!
99991,99989 IS ANOTHER GOOD ONE TOO!

```
1 CLS:INPUTX,Y:Z=INT(X/Y):PRINTZ;".";:FORW=0TO1STEP0:X=10*(X-Z*Y
):Z=INT(X/Y):PRINTZ;CHR$(24);:NEXT
```

```
? 21.9999,6.9999
 3 . 1 4 2 8 8 7 5 2 4 6 3 0 8 1 7 1 4 4 3 6 7 1 0 3 3 6 2 3 3
 5 7 3 9 8 3 6 2 5 3 5 4 6 9 3 4 3 0 8 4 5 9 7 2 9 9 1 1 3 6 0
 5 4 8 9 3 7 3 2 5 9 2 8 5 6 8 3 4 9 5 0 5 3 6 9 5 9 0 0 3 5 2
 8 1 0 0 3 3 3 9 6 7 8 8 4 0 5 1 2 9 9 1 4 6 4 3 4 1 5 6 3 6 0
 1 3 9 2 6 0 2 3 9 0 9 7 7 4 4 2 9 5 1 5 4 4 3 8 0 4 9 5 8 6 8
 6 9 0 3 8 0 6 7 9 5 2 1 7 1 8 5 2 6 6 6 9 5 2 9 2 4 4 4 8 7 4
 4 5 1 7 4 9 9 9 5 6 4 6 9 4 4 2 8 1 3 1 2 3 3 4 5 6 4 0 7 5 3
 4 8 4 3 4 5 1 4 4 0 1 8 3 2 6 0 6 8 0 7 9 1 4 8 4 7 2 3 2 4 0
 4 4 8 9 8 5 3 9 1 8 0 2 7 9 7 2 4 8 5 4 0 8 9 1 4 6 9 3 5 5 2
 3 7 3 0 5 3 5 6 1 5 6 6 6 0 4 1 4 7 1 2 5 7 1 6 7 8 1 5 2 9 3
 1 2 2 2 6 5 2 4 8 6 8 4 2 1 7 1 3 1 8 6 0 1 6 5 6 6 3 7 7 2 7
 8 3 9 9 5 0 0 4 2 1 3 1 2 0 9 2 0 5 4 8 6 5 1 2 1 9 5 2 7 1 1
 4 7 6 5 7 1 9 3 4 2 8 6 2 2 5 1 8 9 7 0 5 4 3 1 5 5 2 5 1 9 5
 4 0 0 3 0 4 2 5 3 6 9 3 5 2 9 7 5 7 0 2 6 6 1 4 9 7 8 2 9 6 3
 6 5 8 9 4 0 8 5 7 3 0 0 0 7 8 1 7 5 1 6 5 7 7 8 7 6 0 3 4 8
```

IN MAV'S WORKSHOP

BY IAN MAVRIC

BUILDING THE SUPER MICRO PART 3

IN THE LAST ISSUE I TALKED ABOUT HOW TO ADD FOUR INTERNAL DISK DRIVES INTO THE "SUPER MICRO", A MAXED OUT MODEL 4 WITH 128K, SPEED UP KIT, RS232, HI-RES GRAPHICS ETC. IT'S A MODERN-DAY VERSION OF THE C-U SERIES OF THE SAME NAME[1]. THIS MONTH I TALK ABOUT HOW TO UTILISE THE 80-TRACK DISK DRIVE IN POSITION #0 AND BOOT OUR FAVOURITE OPERATING SYSTEMS: LDOS, LS-DOS AND NEWDOS.

OVER A BEER WITH JOHN BENSON WE WERE DISCUSSING THE PROS AND CONS OF SETTING UP THE FOUR DISK DRIVES IN VARIOUS WAYS. MY PREFERRED SYSTEM WAS TO SET UP DRIVES #0, #1, #2 WITH 80-TRACKERS AND #3 WITH A 40-TRACKER. I REASONED THAT THIS WOULD LEAVE A 40-TRACK DRIVE FREE IF SOMEONE SENT ME A REGULAR TRS-80 MODEL I/III/4 DISK I COULD JUST READ IT FROM DRIVE #3. IN LDOS I COULD EVEN USE THE CONV/CMD PROGRAM TO READ DISKS OFF TRSDOS 1.3 DISKS. JOHN MADE A GOOD POINT THAT BY SETTING UP DRIVE #0 WITH A 40-TRACKER AND THE REST 80-TRACKERS, ANY OS COULD BE BOOTED IN DRIVE #0, EVEN TRSDOS 1.3 DISKS. MY METHOD EXCLUDES TRSDOS 1.3, BUT ALLOWS AN 80-TRACK BOOT DISK, IN WHICH I HAVE LS-DOS 6.3.1, PROFILE 4+, EDAS, PRONTO, LESCRIPT, DOUBLE DUTY, HYPERDRIVE AND PRO-WAM ALL ON THE ONE DISK. SO THERE ARE POSITIVES AND NEGATIVES TO EITHER SETUP. AS YOU KNOW FROM LAST MONTHS ARTICLE I WENT WITH MY PREFERRED SETUP AND SO THAT WOULD REQUIRE 80-TRACK BOOT DISKS. HERE'S HOW YOU DO IT:

3A. TEMPORARILY SET UP YOUR MODEL 4 WITH A 40T #0 AND 80T #1

IF YOU ONLY HAVE THE ONE MACHINE THEN YOU'LL NEED TO SET UP YOUR SYSTEM TEMPORARILY WITH DRIVE #0 AS A 40 TRACKER AND DRIVE #1 AS AN 80 TRACKER. SINCE I'D ALREADY SET UP THE SUPER MICRO AS ABOVE, RATHER THAN PULL IT ALL APART AGAIN, I PLUGGED AN EXTERNAL 80 TRACK DISK DRIVE INTO MY MODEL 4D AND MADE THE BOOT DISK USING THAT SYSTEM, THEN WALKED OVER TO THE SUPER MICRO AND TESTED IT.

3B. LDOS AND LS-DOS

BOOT THE SYSTEM AND FORMAT AND BLANK DISK IN THE 80 TRACK DRIVE:

FORMAT :1 (CYL=80,SIDES=2)

WHICH FORMATS A 80 TRACK DOUBLE-SIDED DISK TO A CAPACITY OF 720K, BEWARE THIS TAKE MUCH LONGER THAN THE OLD 184K OR 360K FORMATTING YOU'RE USED TO. USE DISKS RATED TO 96TPI DOUBLE-DENSITY IF YOU CAN FIND THEM, AND REMEMBER HIGH-DENSITY 1.2MB DISKS WON'T WORK. THEN:

BACKUP SYS0/SYS:0 :1 (S,I)

WHAT THIS DOES IS PUTS SYS0/SYS ON THE FIRST TRACK OF THE BOOT DISK WHICH IS WHERE IT NEEDS TO BE TO MAKE A BOOTABLE DISK. THEN:

BACKUP SYS/SYS:0 :1 (S,I,Q=N,NEW)

TRANSFERS EVERYTHING ELSE TO THE 80 TRACK DISK. IF ITS GONE TO PLAN, YOU SHOULD NOW BE ABLE TO REMOVE THIS DISK TRY IT IN A SYSTEM WITH AN 80 TRACK DRIVE #0. THE PROCESS IS THE SAME FOR LDOS

AND LS-DOS. IF IT DOESN'T WORK, YOU NEED TO START AGAIN FROM SCRATCH. THE IMPORTANT THING HERE IS TO PUT SYSØ/SYS IN THE RIGHT PLACE ON THE NEW DISK. ONCE ITS ALL WORKING, MAKE BACKUPS (QFB:Ø :1 FOR LDOS OR DISKCOPY:Ø :1 FOR LS-DOS).

3C. NEWDOS/8Ø V2.Ø (ALSO WORKS ON WARRICK SANDS NEWDOS/9Ø)

SINCE PDRIVE IS THE KEY TO SETTING UP THE DRIVE PARAMETERS, LETS SAY THAT DRIVE #1 IS SET UP WITH YOUR 8Ø-TRACKER, THEN THE FOLLOWING WILL CONFIGURE THE DRIVE PARAMETERS PROPERLY:

PDRIVE,Ø,1,TI=A,TD=G,TC=8Ø,
SPT=36,TSR=Ø,GPL=8,DDSL=35,
DDGA=6,A

ONCE THAT IS DONE SIMPLY PUT A BLANK DISK IN DRIVE #1 AND ISSUE THE COMMAND:

COPY,Ø,1,,CBF,FMT

THIS FORMATS THE DISK AND THEN COPIES THE FILES OVER IN ONE MOTION. THE CBF PART IS IMPORTANT BECAUSE OTHERWISE NEWDOS WILL ATTEMPT TO DO A SECTOR-BY-SECTOR MIRROR IMAGE COPY WHICH WILL NOT WORK. TRUST ME, I'VE TRIED. SINCE I'M NOT A BIG NEWDOS USER THIS ONE HAD ME STUMPED UNTIL SCOTT KEVILL SUPPLIED ME WITH THE ANSWER, SO ALL THIS PART OF THE ARTICLE SUBMITTED WITH THANKS TO SCOTT.[2]

NEXT TIME: PART 4: FINISHING IT OFF... HI-RES GRAPHICS, RS232, AND A SPEED-UP MOD.



IN MAV'S MONITOR WORKSHOP

EVERYTHING YOU WANTED TO KNOW ABOUT TRS-8Ø MODEL I, III AND 4 CRTS

THE MONITORS ON TRS-8ØS ARE PRETTY BASIC AND EASY TO UNDERSTAND, SO I'LL EXPLAIN THEM ALL GIVEN MY 3Ø YEARS EXPERIENCE WORKING WITH THE SYSTEMS. THESE SHOULD HELP YOU WITH KNOWING WHAT YOU ARE LOOKING AT AND WHAT TO EXPECT WHEN BUYING REPLACEMENT PARTS FOR YOUR MONITORS AS THEY WEAR OUT.

FIRST, SOME DEFINITIONS:

TELEVISION GRADE CRT: A TUBE WITH FAST DECAYING PHOSPHOR[3] (P4 NORMALLY) USED IN BLACK & WHITE TELEVISIONS BECAUSE IT WORKS WELL WITH FAST CHANGING IMAGES, AS YOU WOULD SEE ON A TELEVISION. RAPID STROBING OF THE IMAGE CAUSES EYE-ACHE IF USED AT SHORT DISTANCE FOR LONG HOURS (SUCH AS SITTING IN FRONT OF A TRS-8Ø).

COMPUTER GRADE CRT: A TUBE WITH MEDIUM DECAYING PHOSPHOR (P4 FOR B&W, P31 FOR GREEN, AND P55 FOR AMBER) USED MAINLY IN COMPUTERS. THE SLOWER DECAY-RATE MAKES THE STROBING OF THE TEXT LESS TIRING ON THE EYES.

MODEL I MONITORS: THERE ARE THREE TYPES AND ALL ARE PARTIALLY GUTTED TELEVISION SETS.

26-1201: THE "ORIGINAL" TRS-8Ø MONITOR IS ACTUALLY AN RCA B&W TELEVISION, AND IS FITTED WITH AN RCA 12VBNP4 OR 12VBYP4 TV-GRADE PICTURE TUBE (OBVIOUSLY). SEE NOTE 1.

26-9201: IS A B&W TRS-8Ø MONITOR MADE BY SANYO OR

TOSHIBA (I'VE YET TO CONFIRM WHICH ONE) AND SOLD IN THE EUROPEAN AND AUSTRALIAN MARKETS, AND IS ALSO A GUTTED TELEVISION. HOWEVER THE ELECTRONICS ARE A GENERATION MORE REFINED THAN THE RCA AND SO WE GET A BETTER PICTURE FROM THE TOSHIBA 310DMB4 TELEVISION-GRADE PICTURE TUBE. SEE NOTE 2.

26-9202: IS THE LAST AND BEST MONITOR TANDY MADE FOR THE MODEL I, IT FINALLY HAS A COMPUTER-GRADE 310HDB31 GREEN CRT WHICH PRODUCES A VERY SHARP IMAGE, ALMOST AS GOOD AS A MODEL III.

MODEL III: TV GRADE B&W CRTS AND ANALOGUE ELECTRONICS

ON THE MODEL III THEY FITTED ONE PICTURE TUBE, A TELEVISION-GRADE RCA 12VCLP4 WHICH IS THE SAME BASIC DESIGN AS THE MODEL I'S ORIGINAL MONITOR, BUT WITH A MUCH MORE SOPHISTICATED ANALOGUE VIDEO BOARD (AS COMPARED TO THE MODEL I MONITOR), A QUITE GOOD PICTURE IS PRODUCED. SEE NOTE 3.

MODEL 4: B&W AND GREEN COMPUTER-GRADE CRTS AND DIGITAL ELECTRONICS (MOSTLY)

ON DISK BASED MODEL 4S THEY FITTED A NEWLY DESIGNED HI-QUALITY DIGITAL VIDEO BOARD, AND A COMPUTER-GRADE 12VCMP4 B&W CRT FOR US MACHINES, AND A 12VCMP31 GREEN CRT FOR EXPORT COMPUTERS. AFTER ABOUT A YEAR OF THIS THE INSTALLED THE 12VCMP31 ON ALL MODEL 4 DISK SYSTEMS US AND NON-US, TO COINCIDE WITH THE NEW MODEL 4 GATE ARRAY MOTHERBOARD. SEE NOTE 4.

BY CONTRAST, CASSETTE-BASED MODEL 4S GOT THE MODEL III VIDEO SYSTEM OF AN ANALOGUE VIDEO BOARD COUPLED WITH A TELEVISION-GRADE 12VCLP4 CRT. SEE NOTE 3.

OCCASIONALLY YOU FIND AN EARLY MODEL 4 (AND MODEL 12) WITH AN RCA M31-334GH GREEN CRT FITTED, THIS IS AN INTERCHANGEABLE EQUIVALENT TO THE 12VCMP31.

MODEL 4P/4D: RE-DESIGNED DIGITAL ELECTRONICS BOARD

IT MAY SEEM STRANGE THAT I CATEGORISE THE 4P AND 4D TOGETHER, BUT IT'LL BE CLEAR IN A MOMENT. TANDY RE-DESIGNED THE MODEL 4 DIGITAL VIDEO BOARD TO FIT IN THE SMALLER 4P CASE, THE CABLE TO THE MOTHERBOARD NOW CONNECTS TO THE "SIDE" OF THE VIDEO BOARD AND NOT THE "BOTTOM". THEY ALSO TWEAKED THIS BOARD TO PROVIDE A SUPERIOR PICTURE ON THE SMALL 4P SCREEN.

WHEN TANDY DECIDED TO KILL OFF THE 4P DUE TO POOR SALES, TCA (TANDY COMPUTER ASSEMBLY, IN TEXAS) FOUND THEY HAD A SURPLUS OF THESE SMALLER VIDEO BOARDS AND SO THESE THEN BECAME THE VIDEO BOARD USED IN THE MODEL 4D. THIS IS WHY THE 4D HAS A SUPERIOR PICTURE TO ALL PREVIOUS TRS-80S, HOWEVER TANDY COMPLICATED THINGS BY CHANGING THE PINOUTS ON THE WIRING LOOM FROM THE MOTHERBOARD TO THE BRIGHTNESS/CONTRAST CONTROLS AND VIDEO BOARD ON THE 4P/4D. THE RESULT OF PUTTING THIS BOARD INTO AN EARLY M4 OR AN M3 IS AN OVERDRIVEN RASTER AND NO VISIBLE TEXT ON THE SCREEN.

CRTS FOR THE 4P WAS THE M24-302W/P B&W TUBE ON EARLY US 4PS, THEN THE M24-302GH/P GREEN SCREEN TUBE ON ALL EXPORT 4P MODELS AND ON US GATE ARRAY 4PS, AND THE 4D GOT THE SAME 12VCMP31 GREEN TUBE AS THE GATE ARRAY MODEL 4.

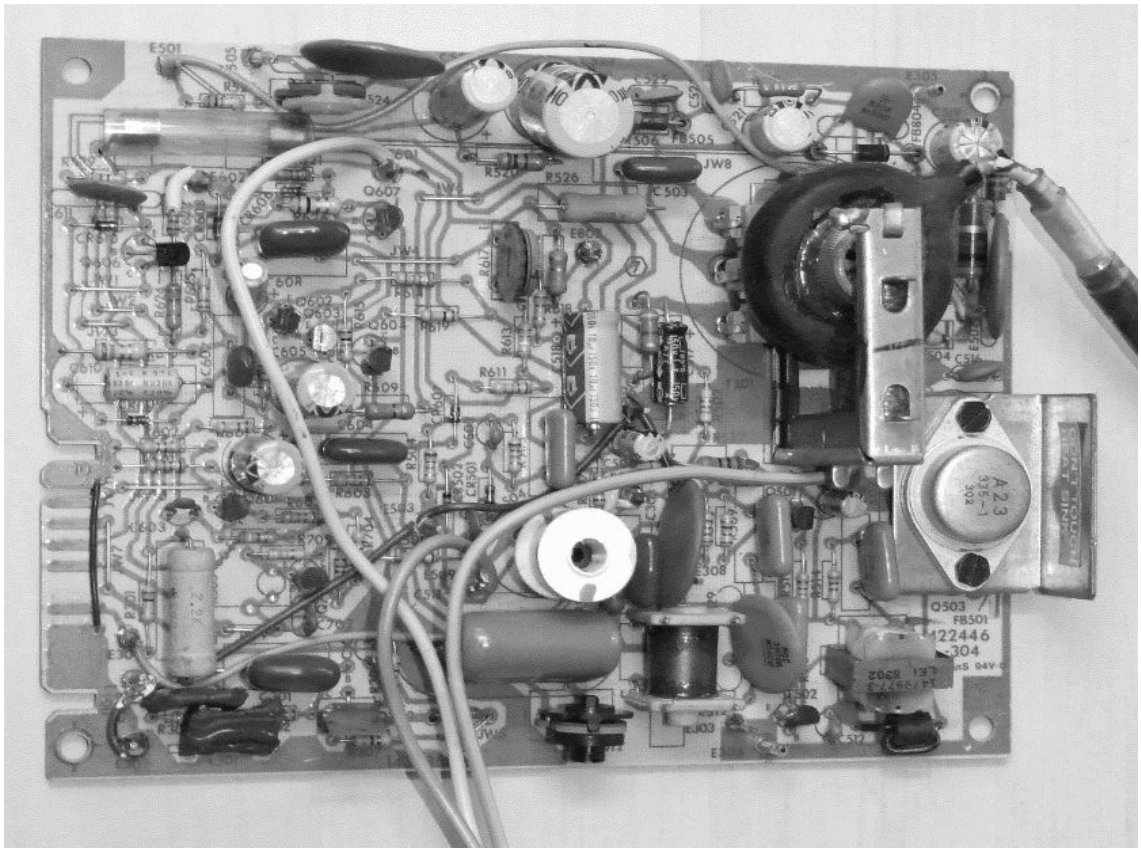
NOTES:

1. THESE MONITORS CAN BE UPGRADED WITH A COMPUTER-GRADE GREEN OR AMBER TUBE, BUT THE RESULT IS USUALLY LESS IMPRESSIVE THAN HOPED, DUE TO THE RATHER ARCHAIC ELECTRONICS IN THE MONITOR, THE DESIGN OF WHICH DATES BACK TO THE EARLY 1970S. I HAVE AN AMBER CRT IN MINE AND WHILE IT WORKS WELL, IT HAS NO WHERE NEAR THE SHARPNESS OF THE SAME TUBE FITTED TO A MODEL III OR 4.
2. THESE MONITORS CAN BE UPGRADED TO A COMPUTER GRADE AMBER OR GREEN CRT WITH GOOD RESULTS.
3. THIS CAN BE IMPROVED MARKEDLY BY FITTING A COMPUTER-GRADE CRT (SUCH AS THE 12VCMP31) AND INSTALLING NEW ELECTROLYTIC CAPACITORS ON THE VIDEO BOARD TO BRING THE SIGNALS BACK INTO SPEC. IN FACT CAPS DEGRADING OUT OF SPEC ON ALL VIDEO BOARDS IS THE PRIMARY REASON THE PICTURE LOSES FOCUS, FOLLOWED BY THE TUBES JUST PLAIN WEARING OUT.
4. THE 12VCMP4 CAN BE REPLACED BY A 12VCMP31 FOR THOSE WHO WANT TO CONVERT A US B&W MODEL 4 TO A GREEN SCREEN.

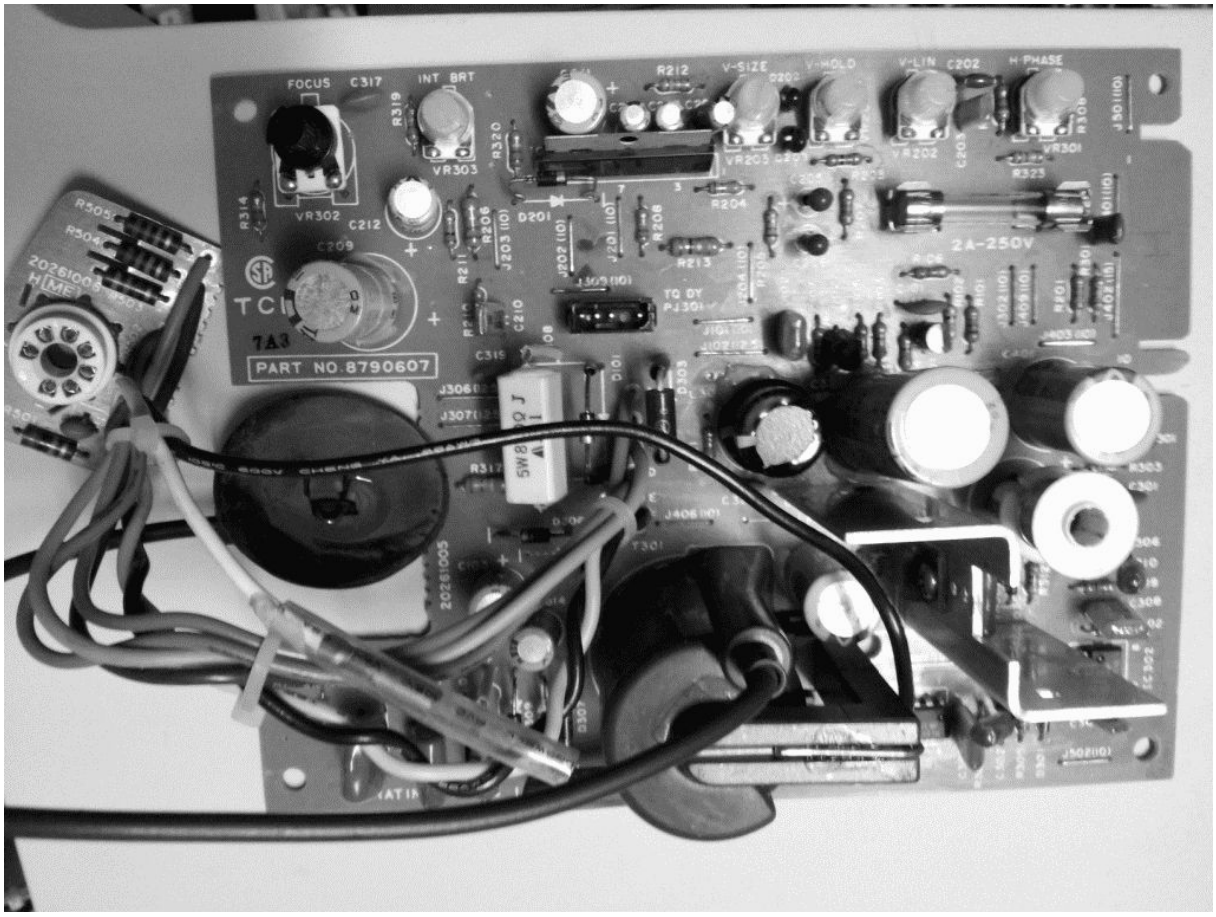
IANM@TRS-80.COM

REFERENCES

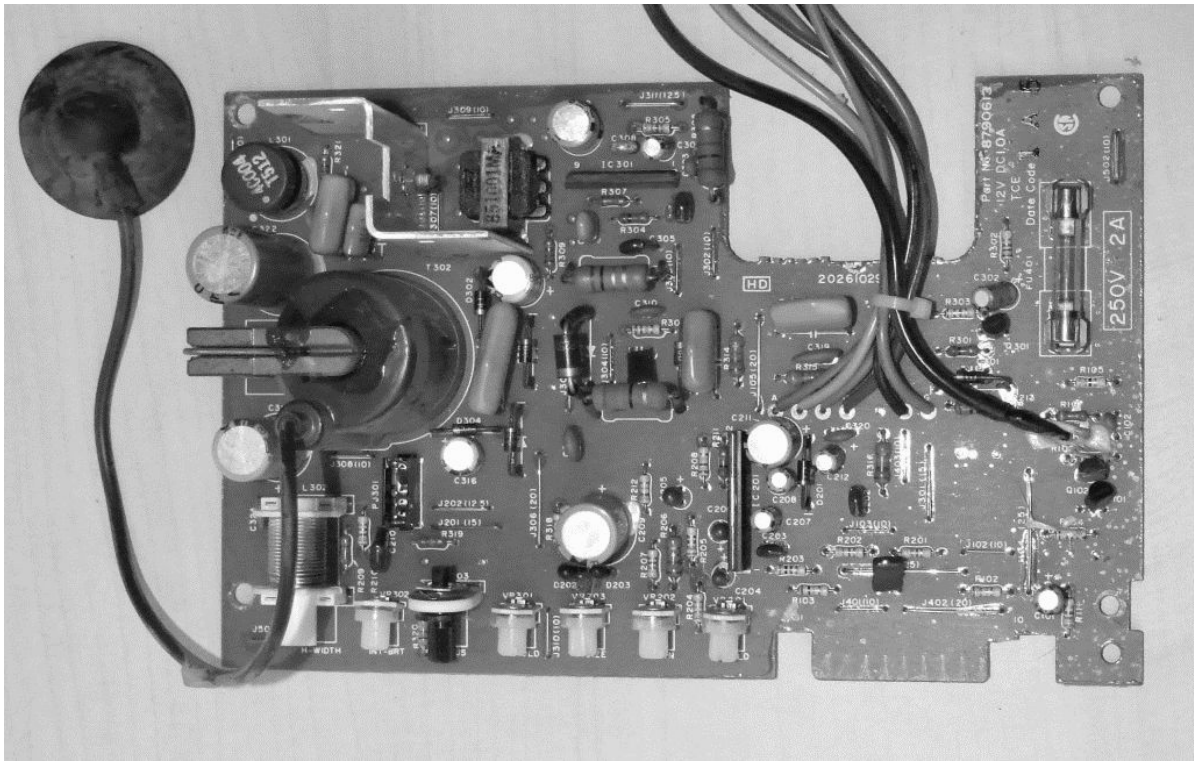
1.
"BUILDING THE SUPER MICRO", COMPUTER USER, FEBRUARY 1984 PP. 14-16; MARCH 1984 PP. 14-18; APRIL 1984 PP. 36-38; MAY 1984 PP. 26-30.
2.
HTTP://TECH.GROUPS.YAHOO.COM/GROUP/TRS-80/MESSAGE/53973.
HTTP://EN.WIKIPEDIA.ORG/WIKI/PHOSPHOR



A MODEL 3, ANALOGUE VIDEO BOARD



A MODEL 4 LARGE VIDEO BOARD



A MODEL 4P/4D SMALL DIGITAL VIDEO BOARD

(CONTINUED FROM PAGE 1)
I'VE REPLACED, JUST FOR THIS ISSUE, AT THE READY PROMPT WITH A FEW WORDS OF APOLOGY. I'M ALWAYS DELIGHTED TO RECEIVED ANY TANDY RELATED ARTICLE FOR ANY MODEL. SO PLEASE, DON'T THINK THAT TRS8BIT IS FOR M1'S ONLY. IT'S JUST THAT MY EXPERIENCE, THOUGH VERY LIMITED, IS M1 ORIENTATED.

I CAME ACROSS ANOTHER HI-RES UPGRADE ARTICLE WHILE SEARCHING THROUGH OLD EDITIONS OF NATGUG NEWS. I HOPE IT MIGHT HELP SOMEONE OUT THERE, OR AT LEAST JOG SOMEONE'S MEMORY WHO MIGHT BE ABLE TO IDENTIFY THE HARDWARE.

KEVIN SMITH WROTE A NICE CALENDAR PROGRAM, ORIGINALLY FOR THE NASCOM 2. I WAS SO IMPRESSED WITH IT, I'VE CONVERTED IT TO RUN ON MY MODEL 1.

NOW THE ONE-LINERS.
BRIAN RAITER HAS PROVIDED US WITH 2 GRAND COMPILATIONS OF VARIOUS ONE-LINERS THAT HE'S FOUND THROUGH THE YEARS. AS HE SAYS, ALTHOUGH HE'S NOT THE AUTHOR OF ALL OF THE PROGRAMS, HE'S WRITTEN A PROGRAM TO COMBINE A NUMBER TOGETHER. THIS IS AVAILABLE FROM THE MUPPETLABS WEBSITE. THEY DO RUN INDIVIDUALLY, SO I'VE BROKEN THEM DOWN INTO SINGLE ITEMS JUST AS THEY APPEAR ON A M1 SCREEN.
BRIAN HAS ALSO WRITTEN A 'NEW' ONE-LINER, 'TWIRL', JUST FOR THIS OCCASION!
ALTHOUGH, AS HE SAYS, ON A STANDARD M1 IT DOES RUN RATHER SLOWLY. HOWEVER, SPEED HAS NEVER BEEN AN ISSUE WITH ONE-LINERS AND IT DOES PRODUCE A SUPER SCREEN DISPLAY.

MAV. HAS COME UP WITH A REALLY USEFUL ONE-LINER TO ENABLE YOU TO FIND WHICH LOWERCASE MOD YOUR MACHINE HAS.
E.T.FONEHUME HAS PRODUCED A SEQUENCING PROGRAM WHICH IS RATHER CLEVER. NICE TO HEAR FROM YOU AGAIN E.T.
THE OTHERS, I'VE EITHER MANAGED TO REDUCE FROM A MULTI-LINE PROGRAM OR HAD A BRAINWAVE ALL OF MY OWN! (I HOPE YOU'LL EXCUSE THE 2 LINES IN THE CHESS PROGRAM)

I'VE PLACED ALL EIGHT OF THE NEW ONE-LINERS TOGETHER AS A FEATURE SHOWING NOT JUST THE PROGRAM BUT A SCREEN DISPLAY AS WELL.

MAV CONTINUES HIS SERIES OF ARTICLES WITH PART 3 OF 'BUILDING THE SUPER MICRO'.
HE'S ALSO PRODUCED A VERY PROFESSIONAL AND INFORMATIVE VIDEO SHOWING HOW TO CHANGE A CRT SCREEN.
THERE'S BEEN A LINK ON OUR WEBSITE FOR SOME TIME NOW FOR YOU TO USE, IT'S HTTP://WWW.YOUTUBE.COM/WATCH\?V=9SMXSMSSCZU
HIS WEBSITE IS ALWAYS WORTH A VISIT. HAVE YOU SENT A PHOTO OF YOUR SYSTEM TO HIM YET?

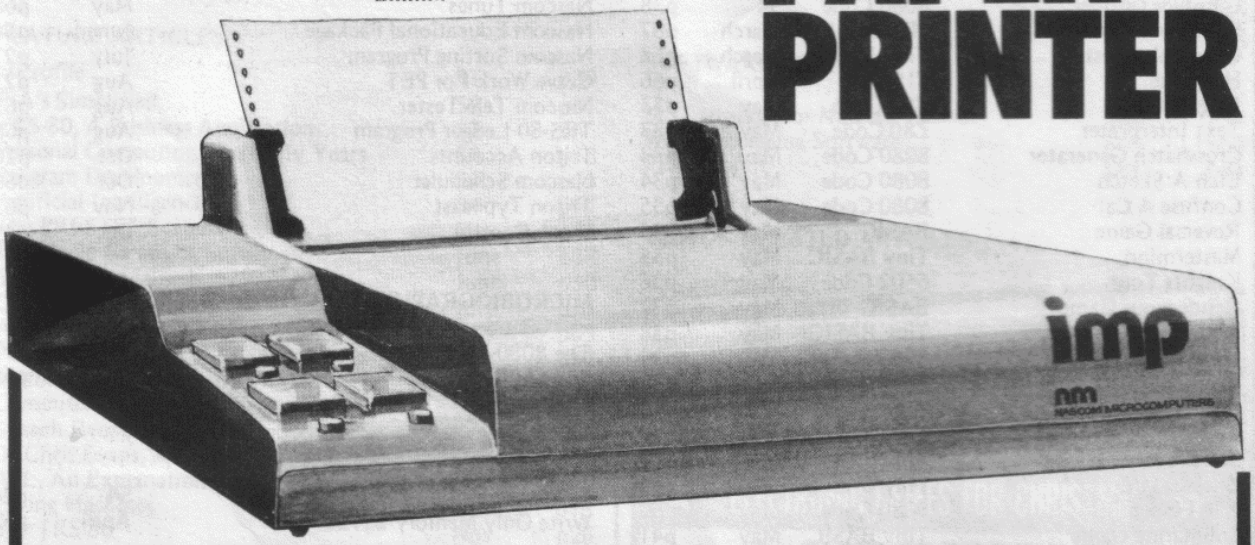
I HOPE YOU FIND THIS EDITION ENJOYABLE. ONCE AGAIN, THANKS TO ALL OF YOU WHO HAVE SENT IN YOUR CONTRIBUTIONS.
I'M LOOKING FORWARD TO THE SEPTEMBER EDITION ALREADY, SO ANY ARTICLES YOU CAN SEND IN WILL BE MOST WELCOME.
IN THE MEAN TIME TAKE CARE-



DUSTY

nascom imp

PLAIN PAPER PRINTER



BOXED AND BUILT FOR ONLY £325 PLUS VAT

FEATURES

- Serial RS232 interface.
- 80 characters wide.
- Bi-directional printing.
- 60 lines per minute.
- 10 line print buffer.
- 96 character ASCII set. (includes upper/lower case, $,#, £)
- Automatic CR/LF.
- 8½" paper (pressure feed).
- 9½" paper (tractor feed).
- Tractor/pressure feed.
- Baud rate from 110 to 9600.
- External signal for optional synchronisation of baud rate.

The Nascom IMP plugs straight into a Nascom 1/2 but is **usable with all other micro systems.** Parallel option will be available shortly.

TO NASCOM MICROCOMPUTERS LTD
92 BROAD STREET
CHESHAM
BUCKS
Tel: 02405 75155

nm Nascom Microcomputers

NM/CT/2

Please send me.....Nascom IMPs at £325 each plus VAT plus £2.50 p&p

NAME _____

ADDRESS _____

_____

_____

_____

ACCESS/ BARCLAYCARD NO _____

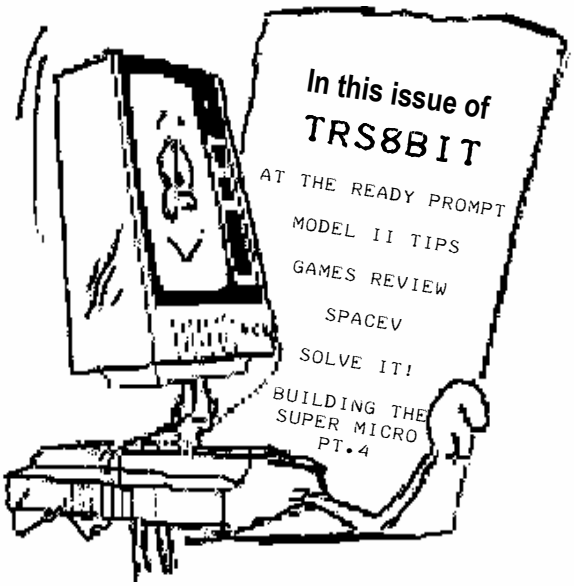
TRS8BIT



HELLO EVERYONE AND WELCOME TO THE SEPTEMBER 2012 EDITION OF TRS8BIT. I'M NOT QUITE SURE EXACTLY WHAT HAPPENED TO THE SUMMER, BUT AS THE NIGHTS ARE NOW DRAWING IN, IT'S SAFE TO ASSUME THAT WE MUST HAVE HAD IT!

FIRST, LET ME THANK EVERYONE WHO EMAILED ME WITH SUCH NICE COMMENTS ON THE LAST ISSUE. IT SEEMS THE ONE-LINERS AMUSED QUITE A NUMBER OF YOU.
I JUST THINK THAT THEY ARE SUCH FUN, BEING QUICK AND EASY TO PRODUCE AND SOMETHING THAT, WITH MODERN COMPUTING HAVING SUCH VAST AMOUNT OF MEMORY, IS NOW, SO VERY RARE.

I'M PLEASED TO REPORT THAT MIGUEL DUTRA'S WEBSITE IS BACK WITH US. MIGUEL HAS REVAMPED MUCH OF THE INFORMATION ON THE CP-500 AND IS WELL WORTH A VISIT.



IAN MAVRIC HAS PRODUCED A VIDEO SHOWING HOW TO CHANGE OR UPGRADE A CRT SCREEN. THERE WAS A LINK TO YOUTUBE ON OUR SITE. FOR FURTHER DETAILS, PLEASE CONTACT HIM VIA HIS WEBSITE.

IRA HAS REPORTED THAT VERNON B HESTER HAS RELEASED, INTO THE PUBLIC DOMAIN, ALL HIS MULTIDOS SOFTWARE. IRA HAS ALSO UPGRADED HIS CP/M PAGES AND HAS JUST ABOUT COMPLETED HIS TANDY CATALOGUE DATABASE. THIS IS TRULY A REMARKABLE ACHIEVEMENT. JUST ABOUT EVERY CATALOGUE NUMBER FROM TANDY HAS INFORMATION AND A PHOTOGRAPH WHICH IS NOW AVAILABLE ON DISPLAY. A VERY IMPORTANT REFERENCE FOR US ALL TO BENEFIT FROM, FOR YEARS TO COME.

GNOTSOFT HAVE RELEASED, ON THE ANDROID MARKET PLACE, A TRS-80 ANDROID-NIM LOOK ALIKE FOR YOUR ANDROID TABLET OR PHONE.

I NOTICED, ON THE TRS-80 FORUM, SCOTT GOLDMAN HAS OBTAINED A LOTHAREK SD FD AND IS HOPING TO HAVE IT RUNNING IT ON HIS MODEL 4. I'M LOOKING FORWARD TO HEARING HOW THING GO FOR HIM AND WHAT, IF ANY, PROBLEMS HE ENCOUNTERS. HAS ANYONE TRIED THIS OUT ON A MODEL 1? AGAIN, I'D LOVE TO HEAR DETAILS.

KEVIN WARNER, FROM AUSTRALIA, HAS CONTACTED ME. HE HAS SET UP A NEW TRS-80 SITE. HE'S FEATURING A SET OF BASIC PROGRAMMING TUTORIALS, WHICH, IF YOU'RE

(CONTINUED ON PAGE 27)

AT THE
READY
>
PROMPT



WHEN WRITING ANY BASIC PROGRAM, GAME OR OTHERWISE, WHICH INVOLVES THE DIRECTIONAL ARROW KEYS, IT'S A SIMPLE MATTER TO PEEK MEMORY LOCATION 14400 TO SEE WHICH, IF ANY, OF THE ARROW KEYS ARE BEING PRESSED. THERE IS A ONE-LINER, BELOW, THAT WORKS FINE IN LEVEL 2, JUST TO GIVE YOU THE IDEA, IN CASE YOU'RE NEW TO THIS SORT OF THING.
PRESSING THE UP, DOWN, LEFT AND RIGHT KEYS RETURNS 8, 16, 32 AND 64 RESPECTIVELY. (YOU CAN PRESS 2 KEYS AT ONCE AND GET A COMBINATION, I.E. UP AND DOWN TOGETHER RETURN 24!)
DON'T FORGET, YOU'LL NEED TO PRESS 'BREAK' TO EXIT THE PROGRAM.

IF YOU'VE RESERVED A PORTION OF MEMORY ON POWER UP, THERE'S NO NEED FOR COMPLEX CALCULATIONS TO SEE HOW MUCH MEMORY IS CURRENTLY AVAILABLE ON YOU MODEL 1. FOR EXAMPLE, IF YOU SET MEMORY TO 15572, JUST TYPE ?15572-MEM AND THE ANSWER IS DISPLAYED. MEM ACTS JUST LIKE A VARIABLE!

AND FINALLY, FOR THIS ISSUE, HERE'S A TIP I PICKED UP FROM HENRY SHEPHERD MANY YEARS AGO. THIS RATHER CLEVER LITTLE TIP UNFORTUNATELY REQUIRES DISK BASIC FOR FUNCTION DEFINITION. DON'T FORGET, FURTHER INFORMATION REGARDING CONTROL CODES IS ON PAGE C/1 OF THE LEVEL 2 MANUAL.

THE FUNCTION DEFINITION COMMAND USED IN CONJUNCTION WITH THE CONTROL CODES CAN BE MOST USEFUL WITH FORMATTED DISPLAYS.
CHR$(29) MOVES THE CURSOR TO THE BEGINNING OF A LINE AND CHR$(25) ADVANCES IT ONE POSITION TO THE RIGHT.
SO IF YOU DEFINE A FUNCTION AS:-

```
DEF FNC$(N)=CHR$(29)
+STRING$(N,CHR$(25))
```

THEN IN ANY LINE FNC$(N) WILL MOVE THE CURSOR TO THE 'NTH' POSITION AND START PRINTING THERE WITHOUT ERASING ANYTHING ALREADY ON THE LINE, AS TAB(N) WOULD DO.
THIS IS VERY USEFUL FOR FILLING IN SPACES IN TABLES OR AMENDING DATA WHEN YOU (OR THE PROGRAM) DOES NOT KNOW WHICH LINE IS IN USE AND CONSEQUENTLY CAN'T USE PRINT@.

--== o0o ==--

```
READY
>LIST
10 A$=INKEY$:IF A$=""THEN10ELSEPRINTPEEK(14400);:GOTO10
READY
>RUN
 8  16  32  64
```

MODEL II TIPS

DERRICK ROWE

IT DOES NOT SEEM THAT LONG AGO THAT I FIRST USED A 16K LEVEL 2 WITH THE WELL-HATED CTR-41 FOR STORAGE. THE COMPANY WHERE I WORK NOW HAS OVER 50 MODEL 1'S MOST OF WHICH ARE BEING USED AT OUR BRANCHES FOR PRODUCING LIFE QUOTATIONS. SURPRISINGLY, MOST OF THE PROBLEMS ARE MECHANICAL - THE SOFTWARE SIDE OF THE SYSTEM HAS BEEN ALMOST TROUBLE FREE. TWO LESSONS HAVE BECOME APPARENT:-
1) ANY USE OF FOREIGN EQUIPMENT IS CERTAIN TO BRING DISASTER.
2) MAINTENANCE IS A MUST AS THE MODEL 2'S EIGHT INCH DISK DRIVES DO REQUIRE REGULAR ATTENTION (PROBABLY ABOUT 1-2 TIMES/YEAR). TANDY NOW PROVIDE A VERY GOOD MAINTENANCE SERVICE WHICH IF COMBINED WITH COMPUTER INSURANCE OFFERS TOTAL PROTECTION AGAINST LOSS OF USE - VITAL FOR BUSINESS USE.

THE MOST TRAUMATIC MOMENT WAS PRODUCED BY TANDY'S INTRODUCTION OF DOS 2.0A. THIS HAS MAJOR DIFFERENCES FROM THE 1.2 VERSION. THESE WERE SUFFICIENT TO MAKE US CONSIDER CP/M BUT ONLY FOR A MOMENT AS THE HORRORS OF CP/M'S DOCUMENTATION TOGETHER WITH THE MUCH TALKED ABOUT BDOS ERRORS CONVINCED US TO STICK WITH TRSDOS. ON A MORE SERIOUS NOTE THERE ARE SOME VERY SIGNIFICANT BENEFITS IN 2.0:-
1) THERE IS A COPY DIRECTORY AUTOMATICALLY MAINTAINED IN THE EVENT OF A DISK PROBLEM. THIS IS A VERY MAJOR BENEFIT WHICH MORE THAN JUSTIFIES ALL THE CONVERSION PROBLEMS. BUT WILL SOMEONE IN FORT WORTH TELL ME WHY MODEL II SCRIPSIT EXCLUDES THIS OPTION?

2) REMOVED REQUIREMENT TO INITIALISE (I) ON CHANCE OF DISKS ALTHOUGH THIS SLOWS DOWN I/O. UNDER PROGRAM CONTROL IT IS POSSIBLE TO RUN AT THE ORIGINAL SPEED USING A SNAPP PATCH (I MENTION UNDER PROGRAM CONTROL AS THIS PATCH RE-INSTATES THE NEED FOR I)

3) THE DIRECTORY NOW SHOWS THE LAST UPDATE DATE AS WELL AS THAT OF FIRST CREATION.

4) BACKUP WILL CONTINUE THROUGH DISK FLAWS ALLOWING MAXIMUM RECOVERY OF DATA.

5) THERE IS A MOVE COMMAND ALLOWING SELECTIVE COPYING E.G. MOVE */BAS:1 TO 2

6) THE WILD CARD SPECIFICATION IS ALLOWED FOR KILL AND DIR

7) OTHER BENEFITS COVER DISK SPOOLING TYPE AHEAD BUFFER, HELP COMMAND FOR SYNTAX INFORMATION, IMPROVED TERMINAL PROGRAM AND BETTER ERROR REPORTING IN BASIC. NOT LONE AGO LEON ASKED FOR TIPS ON HOW TO OVERCOME SOFTWARE PROBLEMS WITH 1.2 PROGRAMS. SILENCE WAS THE GREAT REPLY BUT THE LATEST NEWS IS THAT MOST SUPPLIERS ARE NOW SUPPLYING REVISED PROGRAMS. THERE ARE NOW A NUMBER OF SOFTWARE PACKAGES AND THE FOLLOWING LIST COVERS THE PACKAGES WHICH I HAVE USED. I WOULD STRESS THAT ALL COMMENTS ARE MY PERSONAL OPINION BASED ON MY OWN EXPERIENCE WHICH MAY NOT BE EQUALLY RELEVANT TO ALL APPLICATIONS.

ELECTRIC PENCIL:- EXCELLENT FOR BASIC WORD PROCESSING. PRODUCES TREMENDOUS QUALITY OUTPUT WITH THE DIABLO VERSION USING PROPORTIONAL PRINTING. UNDER 1.2 IT IS VERY EASY TO CORRUPT DISKS BY CHANGING DISKS WITHOUT AN I. UNDER 2.Ø ONLY THE DIABLO VERSION OF THE PROGRAM WILL RUN AND THEN ONLY IN A RESTRICTED WAY. THE PROGRAM MUST BE ENTERED WITH REPEATED USE OF CTRL&F. AT THE PRINT MENU THE USE OF ESC WILL ALLOW THE PROGRAM TO WORK. THIS APPEARS TO OVERCOME THE TYPING PROBLEM DUE TO THE REVISED KEYBOARD ENTRY ROUTINE.
EMBEDDED PRINT COMMANDS NO LONGER OPERATE CORRECTLY E.G. .C OR .L1ØØ.
FILES CAN NOT BE KILLED FROM PENCIL.
THE SYSTEM MUST BE RESET AFTER USING PENCIL.

SCRIPSIT:- LOTS AND LOTS OF WORD PROCESSING FACILITIES WHICH REALLY NEED SPECIAL FUNCTION KEYS TO DO THEM JUSTICE. THE METHOD OF OPENING DOCUMENTS IS AS GOOD IF NOT BETTER THAN THE DEDICATED SYSTEMS. THERE IS A USEFUL MERGE FACILITY FOR AUTOMATIC MAILING WHICH HAS THE 'READERS DIGEST' APPROACH TO INSERTING VARIABLES INTO A STANDARD LETTER. A SPELLING CHECKER, WHICH IS AWAITED, IF EFFECTIVE, WILL MAKE THIS A VERY COMPETITIVE PACKAGE.

IN CONCLUSION SCRIPSIT IS MUCH BETTER THAN PENCIL FOR DEDICATED WORD PROCESSING BUT UNLIKE PENCIL IT CAN NOT BE USED ON ANY OTHER FILES. PENCIL CAN BE USED ON BASIC PROGRAMS SAVED UNDER 'A' OR ON FILES RECEIVED FROM A REMOTE COMPUTER AND SAVED BY TERMINAL. I HAVE USED THIS FACILITY TO COPY FILES FROM A PHILIPS WORD PROCESSOR INTO PENCIL ON THE MODEL 2.

BASIC COMPILERS:- TANDY'S OWN COMPILER IS NOT AS CLEVER AS IT LOOKS. THERE IS MAJOR PROBLEM IN THAT THE BASIC IS NOT STANDARD MICROSOFT, EVEN AFTER ALL THE PAINS OF CONVERSION I AM TOLD THAT THERE ARE NOT ANY SPEED BENEFITS. I REMAIN CONVINCED THAT THE CALL SUBROUTINE FACILITY AND COMMON DATA AREAS ARE TWO USEFUL FACILITIES. IT IS SAD THAT OVERALL THIS PACKAGE DOES NOT SUCCEED.

MICROSOFT'S OWN IMPLEMENTATION OFFERS TREMENDOUS SPEED BENEFITS WHICH TRANSFORM THE SYSTEM INTO A REAL COMPUTER!. THE LIMITATIONS ARE THAT THE COMPILED PROGRAMS USE ABOUT 5 TIMES AS MUCH DISK SPACE AND THAT THE MAXIMUM PROGRAM SIZE IS ABOUT ONE THIRD OF THE INTERPRETER DUE TO MEMORY REQUIREMENTS OF THE L8Ø LOADER. 2.Ø DOS USERS LOSE 2K OF MEMORY DUE TO A RE-LOCATION OF 'TOP'. RACET COMPUTERS PROVIDE A PROCEDURE APTLY NAMED 'TOP' WHICH WILL RESET THIS MARKER. THERE ARE SOME DIFFERENCES WITH THE INTERPRETER IN THAT ARRAYS MUST BE SPECIFICALLY DIMENSIONED - A=1Ø:DIM B(A) IS ILLEGAL AND ERASE IS NOT PRESENT.

SNAPPS XPRINT:- THIS ALLOWS A SELECTION OF BASIC PROGRAMS TO BE PRINTED FROM DOS. THESE FILES CAN BE BOTH COMPRESSED AND SAVED AS 'A' E.G. XPRINT */BAS:1 WILL LIST ALL /BAS FILES ON DRIVE 1. OVERALL VERY USEFUL.

DISC SPOOLERS:- THESE DO NOT SEE VERY USEFUL OWING TO THE LIMITATIONS OF FLOPPY DISK TRANSFER SPEEDS. IT SHOULD NOT BE OVERLOOKED THAT DISK SPOOLING PUTS A LOT OF STRESS ON THE DISK SYSTEM WHICH CAN LEAD TO AN INCREASE IN DISK FAULTS. UNDER BASIC, TANDY'S SPOOL SLOWS DOWN THE KEYBOARD. FOR THE SYSTEMS WITH WHICH I HAVE BEEN INVOLVED, SNAPP'S DISK SPOOLER DOES NOT SEEM TO SPEED UP APPLICATIONS DUE TO DISK CONTENTION. TANDY'S SPOOLER COMES WITH 2.Ø AND IS MOST EFFECTIVE WHEN IMPLEMENTED WITH SCRIPSIT. THERE SEEMS LITTLE POINT IN USING ANYTHING ELSE.

MEMORY SPOOLERS:- SNAPP PROVIDE VARIOUS MEMORY SPOOLERS. AMAZINGLY THE 'TOPSOOL' VARIANT PROVIDES A 2K BUFFER WITH NO LOSS OF USER MEMORY. THIS IS OF SIGNIFICANT VALUE FOR THOSE APPLICATIONS WHERE PROCESSING ALTERNATES WITH PRINTING AND RUN TIMES CAN BE HALVED BY USING THIS FACILITY.

SNAPP'S EXTENDED BASIC:- A LOT OF USEFUL PROGRAMMING AIDS WHICH WILL BE IMMEDIATELY FAMILIAR TO USERS OF NEWDOS. WHY THESE FEATURES ARE NOT IN TRSDOS ESCAPES ME.
RACET COMPUTERS DISK SORT/ MERGE:- IF YOU WANT TO SORT FILES THIS IS ESSENTIAL AND ARGUABLY BETTER THAN A MEMORY APPROACH IN THAT YOUR SORT HAS FAR MORE SCOPE IN TERMS OF EVENTUAL CAPACITY.

VISICALC:- FOR NON-PROGRAMERS THIS REMAINS THE MOST EFFECTIVE SYSTEM FOR USING COMPUTERS, ON THE STOCKS ARE VERSIONS TO HANDLE DATA MANAGEMENT AND TIME SERIES FORECASTING. FOR SIMPLE APPLICATIONS (ALTHOUGH DON'T UNDERESTIMATE THE POWER OF VISICALC'S CALCULATIONS) THIS PROGRAM CAN OFTEN BE THE FASTEST SOLUTION EVEN FOR THE EXPERIENCED PROGRAMMER.

--=== oØo ===--

TRS 80 SOFTWARE

from the leaders in innovative software

We were first for the TRS 80 with Re number Basic

We were first with Level III

We were first with the Howe Monitors 1, 2 and 3.

We were first with Newdos

And we are going to go on being first
SEND SAE FOR LIST

A. J. HARDING
(MOLIMERX)
28 Collington Ave.,
Bexhill, E. Sussex.
Tel. (0424) 220391

BARCLAYCARD VISA

Buy it with Access



FOOTBALL MANAGER

A GREAT NEW GAME from

Addictive Games

for ZX81, ZX80, TRS80, VIDEO GENIE

JUST LOOK AT THESE FEATURES

★ 4 Divisions ★ Promotion & Relegation ★
★ F.A. Cup ★ Full League Tables ★
★ Transfer Market *(Buy and Sell Players)* ★
★ Pick your Team for each Match *(but watch out for Injuries!)* ★
★ As many Seasons as you like ★
★ 7 levels of Play *(from Beginner to Genius!)* ★
★ Managerial Rating *(tells you how good you REALLY are!)* ★
★ Save Game Facility *(continue again another time)* ★
All this and much much more packed tightly into 16K RAM

"IF YOU LIKE FOOTBALL YOU'LL LOVE THIS GAME"

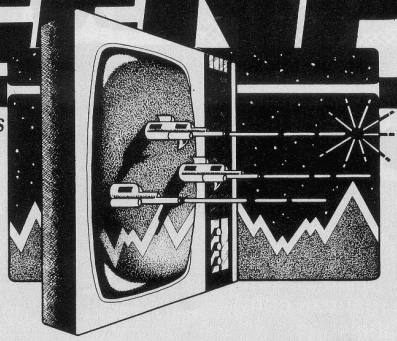
HARDWARE REQUIRED:-

ZX81/ ZX80	TRS80/ Video Genie
8K ROM 16K RAM	LEVEL II 16K RAM

To Order send Cheque/ P.O for £9.95 made payable to:
ADDICTIVE GAMES
at:
P.O. BOX 278
CONNIBURROW
MILTON KEYNES
MK14 7NE
PLEASE STATE COMPUTER

SCREENPLAY

This month Dick Olney reviews games for the TRS-80 Model I.



The games I've been looking at this month all run on the well-established Tandy TRS-80 microcomputer. I was supplied with a basic Model I carrying 16k of RAM (it uses the Z80 processor) and including a CTR-80A cassette deck. This configuration would cost £399 (inc VAT) and plugs into a domestic television. The TRS-80 was an early entrant in the micro scene and has built up a considerable share of the market. It is a fully expandable and versatile machine and, though the lack of colour or high resolution graphics severely limits its capacity as a games unit, it has built up a considerable stock of this type of software.

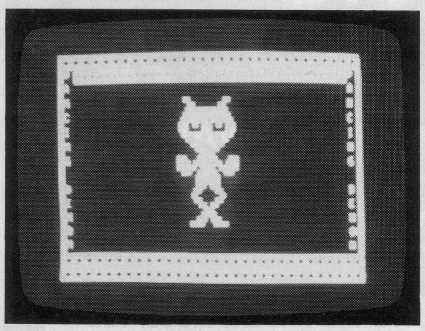
Tandy has more recently brought out its colour computer, which I shall be looking at later on in the year, but on the Model I the graphics-oriented games are necessarily much less sophisticated than on the machines I have previously reviewed. Many of the games include simple sound effects which can be directed through a standard hi-fi system or — as in my case — you can use a mini-amplifier box which costs £7.49. I have looked at a varied assortment of games which I believe is fairly representative of the vast selection available.



GAME: Pyramid of Doom
SUPPLIER: Adventure International
PRICE: £12.50

This is a standard Adventure game set (as the name suggests) in and around an ancient pyramid. I have to admit it took me five hours even to get inside the pyramid, though the frustration of this served mainly to increase the satisfaction of my eventual success. Most of the objects and rooms inside the pyramid are much as you'd expect (a notable exception being the giant oyster!), though there are plenty of original problems for you to wrap your mind around. I would have expected a slightly more extensive configuration for the money and the 'astral guide' is a bit ingratiating and humourless compared with others I've seen, but nevertheless this is a well designed program from what must be considered an expert in the field. The package gives an 'average completion time' of one month. I've no idea of exactly how this is arrived at — but either way its an exaggeration, I would think about 50 hours' continuous play is nearer the truth.

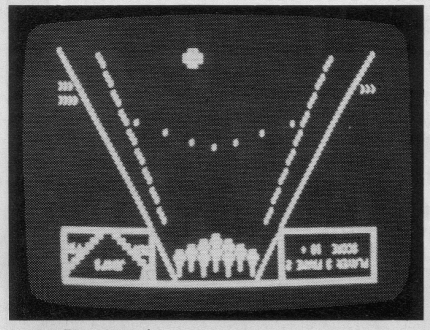
PRESENTATION: ****
COMPLEXITY: *****
VALUE FOR MONEY: ***



GAME: Dancing Demon
SUPPLIER: Tandy
PRICE: £6.95

More of an entertainment than a game really — but unusual enough to demand a mention. Using simple alphabetic codes, you develop dance routines for a character looking more like a friendly alien than a demon, who inhabits a simple stage (the graphics are basic but effective). Fundamental tunes are superimposed on the performance, again using an alphabetic code (which is simple once you get used to typing 'A' for 'C' and 'B' for 'C#' etc). All in all you have 25 notes and 26 different stops to concoct a string of 248 action/note elements, and its quite easy to create all sorts of bizarre results (hence giving your support to the 'Inane uses for new technology' lobby!). This one looks like it was originally written for sales demonstrations and later marketed to prevent salesmen giving it away with the machine. It does inspire overwhelming desires to give demonstrations to friends (neighbours, traffic wardens, cats — well anybody really!) and this, along with its novelty value, probably makes it worth the seven quid; although I'm sure I'd get heartily sick of it before very long.

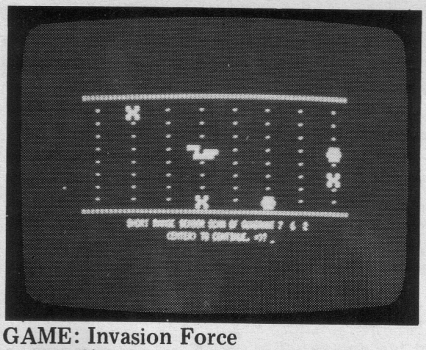
USE OF GRAPHICS: ****
PRESENTATION: *****
ADDICTIVE QUALITY: **
VALUE FOR MONEY: ****



GAME: Tenpins
SUPPLIER: Molinerx
PRICE: £9.50

As you might expect, this program simulates ten-pin bowling. Given a view up the alley towards the pins, you line up a ball (actually a sort of chunky cross shape) on the base line using the arrow keys, and bowl it with the space bar. The manual claims that the force you exert can be increased by holding down the space bar longer, but I found that this had minimal effect. Once moving, the balls direction can be further modified and as it nears its target you can give it a limited amount of spin. Scoring is standard — each game consists of ten two-ball frames — and up to four players can take part. I was surprised that there is no facility for getting players' initials on the score-board since this is such a common and simple feature on games of this type and always adds to the fun, somehow. The graphics and sound are both as good as you can expect on this machine but the game itself is rather uninspired, both in design and execution.

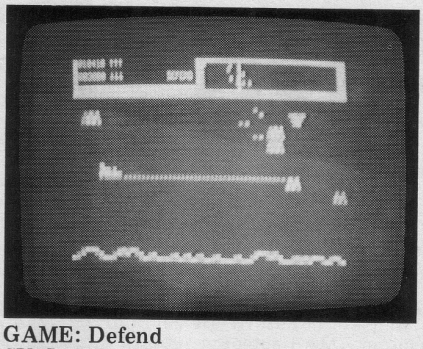
USE OF GRAPHICS: ****
PRESENTATION: **
VALUE FOR MONEY: **



GAME: Invasion Force
SUPPLIER: Tandy
PRICE: £11.95

Tandy's version of Star Trek combines all the standard features with limited real time action and rather poorly designed graphics. Status reports, long and short range sensor scans and power distribution are all displayed at the same time on a rather overcrowded screen — nice idea, but my feeling is that it makes the game less interesting. The battleground is a two-dimensional 10 x 10 matrix of 100 quadrants, and the object is simply to destroy as many 'Jovians' as possible. The standard beam and projectile weaponry (here referred to as 'Masers' and 'Triton' missiles) are complemented by 'Antimatter' pods and an experimental ray which — if you're lucky — can be used to destroy groups of enemy craft. One other interesting feature is the ability to control the power distribution between *all* the major ship's functions — thus making the most economical use of your energy. Movement and battle happen in real time, but because of the design I'm not sure that the game is necessarily enhanced by this attribute. All in all, if you want a game of this genre (which is undoubtedly the case), then you can probably do better than this one.

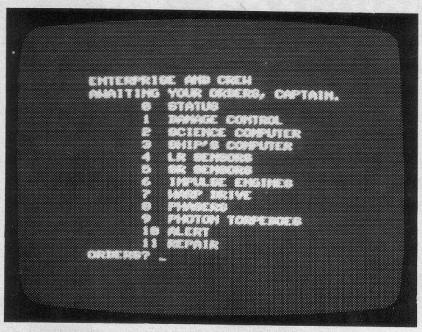
USE OF GRAPHICS: *
PRESENTATION: **
RESPONSE SPEED: **
VALUE FOR MONEY: **



GAME: Defend
SUPPLIER: Molimerx
PRICE: £14.95

This is a real-time arcade type game with some similarities to both Defender (see March issue) and Scramble — this month's Arcade ace. You fly a jet-like craft moving at a steady horizontal speed (well, in fact everything else moves and you stay still!), using the arrow keys. Mostly this involves controlling vertical displacement, since moving horizontally across the screen is of little use (you generally need to be as far back as possible from the approaching objects) and does not provide acceleration or the ability to actually reverse motion. Each phase of the game is in three parts. Firstly you must fight off hordes of alien craft (which mindlessly fire missiles), then navigate a meteor storm, and finally wend your way through a rocky tunnel. In the first part of the game a forward-firing beam is controlled by the space bar and 'smart bombs' are activated by the '@' key. The latter destroy all aliens on the screen, though their remains will still obliterate you if you collide with them, giving the facility limited value. The response time, graphics and sound effects are all quite good but, if anything, the game is rather too easy. Admittedly I've had considerable practice with Defender, but I'm sure that most players would quickly tire of the tasks involved, particularly the tunnel.

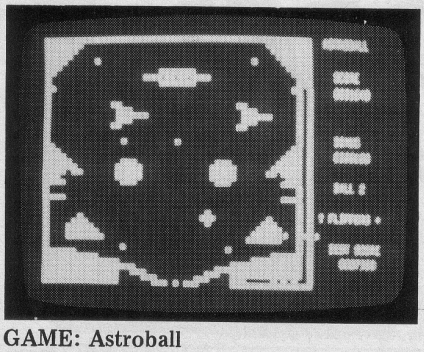
USE OF GRAPHICS: ***
PRESENTATION: **
RESPONSE TIME: **
ADDICTIVE QUALITY: **
VALUE FOR MONEY: **



GAME: Star Trek 3.5
SUPPLIER: Adventure International
PRICE: £9.50

The latest version of the old favourite. Your mission is to explore all 'M-type' planets in a section of the galaxy divided into 192 quadrants, killing any Klingons you might meet on the way. The galaxy is conceived as three-dimensional, thus giving an 8x8x3 matrix. All of the standard Star Trek features are included — short and long range scan, computer facilities, warp and impulse engines, phasers and photon torpedoes. You also have an option for 'alert stations', which in fact determines the power given to the deflector shields. The long range scan representation of a three-dimensional galaxy is confusing to begin with, as are the headings you use with the impulse engines (running zero through nine anticlockwise, where zero is west). Still, there's plenty for both novice and expert to get their teeth into with good graphics and limited sound effects. If you use a TRS-80 and don't yet own one of the many versions of this game available I'd recommend this one — well worth the money.

USE OF GRAPHICS: ***
PRESENTATION: ****
VALUE FOR MONEY: *****

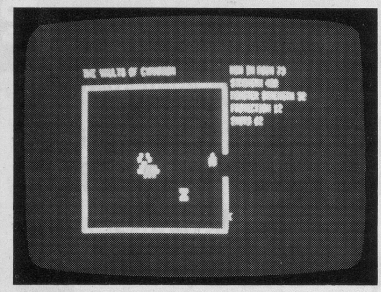


GAME: Astroball
SUPPLIER: Molimerx
PRICE: £13.77

Astroball is a pinball simulator with standard flippers operated by two of the arrow keys. Despite the limitations of the machine's graphics, the screen layout is well designed and has all the types of targets and obstacles you'd expect on a pinball machine, including side lanes. The response time is more than adequate, and quite high scores can be obtained, especially as it's always possible to get extra balls. Five skill levels are provided to vary the speed of the game. Even high resolution colour machines, however, can never really capture the full attraction of a mechanical pinball and — although, of course, few people actually have such machines in their homes — it's one of those games that I just don't feel computers do very well. Only real pinball addicts would want to buy this and then (despite this particular game's many assets) I suspect they'd find it a poor substitute for the real thing.

USE OF GRAPHICS: ***
PRESENTATION: *
RESPONSE TIME: ***
VALUE FOR MONEY: *

SCREENPLAY



GAME: Vaults of Cymarron
SUPPLIER: The Software House
PRICE: £20

As the name suggests, this is an adventure game — but with a difference in that much of it is graphics-oriented real time action with sound effects. The instruction manual, though clearly produced on a low budget, gives an amusing background to the game with illustrations that look like they belong in a 'head bangers' broadsheet. The game starts on the pleasure planet 'Babylonia' where, having sampled 'many of the grosser forms of self indulgence' available there, you find yourself heavily in debt (apparently exacerbated by the massive inflation rate). You are presented with a simple menu representing the planet's 'Presto' information service. Using this you discover the size of your debt and are made an offer by the 'Cred Mutual Assoc' allowing you to pay it off by retrieving some of the valuable artefacts contained in the Vaults of Cymarron (the remains of an ancient shopping precinct). You are allowed a limited amount of cash with which to purchase supplies from the pawnbroker, where a certain amount of bargaining takes place. Your choice of supplies is critical to the main part of the game, and above all it's essential to buy plenty of food. After visiting the pawnbroker you go on to the spaceport from where — having chosen the direction from which you will enter the vaults — you set off for Cymarron. At this point the second part of the program is loaded in from the cassette.

After choosing a skill level (which determines the strength of the monsters) you are eventually plunged into the first room. Commands are entered using single keys such as 'L' for look and 'F' for fight, covering a fairly standard range of adventure vocabulary. All of the rooms are square with up to four exits and contain various obstacles, monsters (only ever one per room) and useful or valuable artefacts. You move your character with the arrow keys and can fire missiles with the space bar (assuming you bought a gun and needle pak from the pawnbrokers). I would suggest that you save the game as soon as possible after entering the vaults, since otherwise if you get killed quickly you'll have to go through all the prelininaries again, which can be very irritating.

The vaults are quite extensive, so there's plenty to explore, though the game does rather lack variety. It is of course not as cerebral as your average Adventure game, and hence much less demanding; but the real-time action makes it less predictable the second time around, perhaps giving it a more lasting attraction. Vaults of Cymarron is nicely presented and well designed, if a little pricey.

USE OF GRAPHICS: ****
PRESENTATION: *****
RESPONSE TIME: ****
VALUE FOR MONEY: ***

Conclusion.

Undoubtedly the best offerings available for the TRS-80 are the classic adventure and Star Trek variants, of which there are many. These are rather more serious games than the arcade type real-time affairs and, as a result, tend to be aimed at a more adult audience. Nevertheless, many of the latter are available for this machine and, though of course not nearly as impressive as their high resolution colour counterparts, they tend also to be comparatively inexpensive. This is obviously not a machine one would buy for its games alone, but it still has plenty to offer the prospective 'screenplayer'. Next month I shall be amusing myself with Commodore's colourful gamey machine, the VIC-20.

The Adventure International software reviewed this month was supplied by Calisto computers of Birmingham.

aculab floppy tape,

The tape that behaves like a disc,
For TRS-80 LEVEL II.

Connects directly to TRS-80 Level 2 Keyboard. Operating and file handling software in ROM. 8 commands add 12 powerful functions to Level 2 BASIC. No buttons, switches or volume controls. Full control of all functions from Keyboard or program. Daisy chain multiple drives. Certified digital tape in endless loop cartridges. Reads and writes in FM format at 9000 Baud. Soft sectored with parity and checksum error detection for highly reliable operation—just like discs. Maintains directory with up to 32 files on each tape, tapes may be write-protected. Supports Basic and machine-language program files, memory image and random access data files. 12 character filespecs —: "FILENAME/EXT:d" (d is drive no. 0—7). Automatic keyboard debounce. Full manual with programming examples and useful file-handling routines.



COMMANDS (usually followed with a filespec and possible parameter list).
@SAVE, @LOAD, @RUN —for BASIC programs, machine language programs and memory image files. @GET, @PUT —moves a 256-byte record between a random access file and BASIC's data buffer. @KILL —removes a file from the directory and releases tape sectors for immediate re-use. @LIST —displays file directory along with sector allocation and free sectors. @NEW —formats tape and creates a blank directory.

Master drive with PSU, Manual and a selection of tapes. £167-00 +£2-00 pp+vat.
Slave drives with PSU £122-00 +£2-00 pp+vat.
(Export orders pp charged at cost)

For further information, Telephone 0525 371393
aculab Ltd.
24 Heath Road, Leighton Buzzard, Beds. LU7 8AB

ALBION SOFTWARE

LAMMAS ROAD,
LONDON, E10 7QT.

Telephone 01-539-7071/2/3/4

Telex 27950 Quote Ref 3635 Dly
Answerback: MONREF G

DUPLITAPE

A copy program which can copy any length program or data-file from tape to tape. Duplitape will run on a computer with 4K of RA.M. , but will copy any length of program including Adventures, Machine Code & Protected Tapes. Duplitape is not a simple copy program but is designed to alter and reposition tape pulses so that difficult tapes will load easily. It requires two cassette recorders.

CHAIN-LINK

This program allows long BASIC programs to be split up so that they will run in a small memory machine. It allows variables including strings to be passed from one program to another. It has a built in MERGE for BASIC programs, and allows two BASIC programs to be kept separate in memory if needed. (16K L II)

THE WIZARD'S CASTLE.

A fantasy adventure with graphic maps. The objective is to find treasures in a 500 room castle. The castle has monsters which have to be fought together with other hazards. Every time the program is run the castle will be different. Magic spells can also be used by some characters. Suitable for all Dwarves, Elves, Hobbits and Humans . (16K L II)

ACULAB DATA-BASE

A general purpose data base to run with an Aculab Floppy Tape under Extended BASIC (which must be obtained from Aculab Ltd.) The use of the P.I.M.S. manual by SCELBI Publications is reccomended. Functions supported include Help Command; Arithmetic; Searching; Sorting; Listing; Printer output if required; Printing of 5-line address labels; Commas are allowed in the text lines. As this is intended for present owners of PIMS the instructions are only briefly summarised except for new commands and general operating procedures.(16+ K L II)

PROFESSIONALS HELP-MATE

This will be available in late May. This program will take a number of time shee from clerks showing the time spent for each client, and produce sheets for each client showing what work has been done for them. HELP-MATE will also be able to price the hourly rates for each clerk. No serial numbers are used, all input being done by name of the client or clerk.
No price has been fixed for this yet.

ALL THESE PROGRAMS ARE SUPPLIED ON CASSETTE AT £9-50 EACH. THEY ARE NOT DISK COMPATIBLE AND WILL ONLY RUN FROM CASSETTE,

PLEASE STATE WHETHER YOU REQUIRE TRS-80 OR VIDEO-GENIE VERSIONS FOR DUPLITAPE.

SPACEV/CMD

Laurie Shields

The following is an account, that others night find interesting and might like to follow, of getting Spacev/cmd to run direct from disc.

Before I was asked if there was anything that I could do, the cassette program had already been put onto disc using LMOFFSET or some other similar utility, but to run the game it was necessary to Load Spacev/cmd, go to Level2 Basic, and then execute a System, /xyz (Enter) sequence of commands.

Normally I would have used Zen and Superzap to tackle this sort of problem but just for the challenge I decided to solve it using just the utilities supplied with TRS2.3 Dos and one of the Edtasm type of Disc assemblers.

The usual reason why some machine code cassette programs don't work when they are loaded from disc is that there is a conflict with some part of the Dos. left over and still active, after control has passed to the program.

The most frequent culprit is the Dos keyboard routine. If the normal Level2 keyboard scan could be restored when the program loads than all the trouble might be over.

The Level2 manual tells us how the Z80 finds out where the keyboard scanning program, known as the driver, is located. It is in part of the TRSSO memory map called the Keyboard Device Control Block and the address of the driver program is located in memory bytes 16405+1 and 16405+2.
So simply power-up Level2 with the break key down and:
PRINT PEEK(16406),PEEK(16407)
Back comes the response from our user friendly computer (if you don't believe me just try it on a Sharp) of 227 and 3, or E3 and 03 in Hex. This means that to check the keyboard Level2 calls a routine in ROM at address 03E3H.
(remembering the Z80 always gets its addresses back to front)

Now we come to the simplest machine code program ever written and anybody can understand it. The object is to load 227 into address 16406 and 3 into the next one. That's it, so if you've never used you disc Edtasm before then now is the time.

From Dos ready just key EDTASM (Enter] and you should get the prompt: * Since we want to insert some text we key I (Enter) and Edtasm will respond by printing 00100 for the line number, in a similar fashion to AUTO 100 in BASIC
You now enter the following lines of code using the right arrow to tab across to the various columns and terminating each line with [Enter]. Since we want to start by putting a 227 into location 16406 we tell this start or origin address to

Edtasm using the ORG statement. After that we use the define byte DEFB instruction for the 227 and the 3 and the program is finished. All that is then required is an END and the number 402DH which means that after the program has loaded control passes back to the Dos at this address.

It should look like this:

```
00100 ORG     16406
00110 DEFB    227
00120 DEFB    3
00130 END     402DH
```

After the fourth line press the Break key to escape the insert mode and key: A/NO [Enter] which asks the assembler to Assemble the source file but No Output to disc or cassette. The response should be:

```
4016            00100     ORG      16406
4016 E3         00110     DEFB      227
4017 03         00120     DEFB      3
402D            0013U     END 4     02DH
00000 TOTAL ERRORS
```

If you have any errors then they must be corrected before the next stage, which is to key:

A (Enter) Edtasm will ask: OBJECT FILL TO DISC OR TAPE ? you reply D.

In response to OBJECT FILESPEC? answer BOUNCE/CMD (Enter) and with luck you will create a very useful CMD file that you can call from DOS which will make you all nostalgic about the good old days when you got three characters for the price of one key stroke.

Now for the magic, from Dos ready key:
BOUNCE (Enter) and then,
SPACEV (Enter) You are saving the Universe an we know it.

However the excessive effort of calling both programs might he too much for some disc users and for their benefit we will combine them.

For a number of reasons its a lot easier to tack things on the end of a machine code program rather than at the start so lets see what our little program looks like on the disc and then have a look at the end of Spacev as well.

Although TRS2.3 doesn't have a Superzap it does have a utility called Diskdump/bas and this will tell us all we need.

Run it from BASIC giving Bounce/cmd for the filespec and your printer will list out

the first (and only with this file) sector.
The first line is all we need and it should look like this:

0 01 04 16 40 E3 03 02 02 2D 40 FF FF FF FF FF FF !...@....-@......!

In simple language the meaning of these bytes is as follows:

Byte Value Meaning

0 01 Wake up Dos there's some something coming.
1 04 In fact there's 4 Bytes coming.
2 16 Right hand half of address where you have to put them.
3 40 Left hand half of address.
Combining these two we get the Load address of 4016H. Which is of course 16406. Easy isn't it ?
4 E3 This is 227 decimal, and it gets loaded into 4016H.
5 03 Likewise this 3 gets loaded into the next address 4017H. That's the last of the four bytes we had to sort out. What happens next ?
6 & 7 02 02 Thanks for your help Dos you can go back to sleep.
After you've told the Z80 to Goto the following address:
8 & 9 2D 40 Just like bytes 2 6 3 they combine to give an Execution address of 402DH.

Nothing else on this disc sector is of any meaningful value.

Now we know all there is to know about the layout and meaning of machine code programs on disc sectors; well not quite all but what's left is just frills and trappings.
So full of confidence we now have to find the end of Spacev and add our new bit onto it. Where do we find the end ? Easy get the Dos to tell us via the Directory, just key DIR (A) (Enter) and in among the other information on the video you will get:

SPACEV/CND LRL= 256 / EOF= 12 / SIZE 3 GRANS

The LRL=256 tells us that each record of the program is 256 bytes, i.e. a full sector, this makes it easy for us as had they been shorter than 256 We would have had to do some sums. The EOF=12 tells us that the file ends in sector 12, bearing in mind that-the Dos does some of its sums by starting at 0 not 1.

Therefore there are 13 sectors to the file and as we are going to use Diskdump to get at it we must remember that Basic starts counting at l, usually.

Power up Basic, run Diskdump again asking for the 13th sector of file Spacev/cmd; the printout should be:

```
 0 20 28 4D 4F 4C 49 4D 45     52 58 29 20 20 20 20 20   ! (MOLIMERX)   !
16 45 4E 54 45 52 20 10 20     53 54 41 44 44 41 52 44 !ENTER    STANDARD!
32 20 20 20 4F 46 20 20 20     50 4C 41 59 20 20 20 20 !    OF       PLAY  !
48 28 30 20 20 3D 20 20 46     41 53 54 29 20 2D 2D 20 !(0 = FAST) --    !
```

64 28 39 20 20 3D 20 20 53 4C 4F 57 29 20 02 01 11 !(9 = SLOW) ... !
80 EE 7B F3 21 00 70 11 53 43 01 EE OB ED BO C3 53 !...!...SC..... S!
96 43 02 02 EE 7B 42 20 41 53 52 43 20 4F 20 52 54 !C.... B ASRC 0 RT!
etc. up to byte 255.

We want to find the end of the program and as we now know it is signified by the two bytes 02 02 followed be the execution address.
Careful examination of the printout shows these to be located at byte 97 with the next two bytes being EE 7B - the execution address 7BEEH.

To add our new bit of program what we have to achieve is a gap of 4 bytes between the program and the exec address, insert 01 04 16 40 and the lob is finished. We used to retain the original execution address of Spacev/cmd rather than putting the 402DH we had with Bounce so that the Z80 will start to execute Spacev after it is loaded.
As we haven't got a Superzap we can only get at this disc sector to alter it as a Random Access file through Basic.

Summarising the problem:
l. Open the file Spacev/cmd for random access.
2 Field the buffer so as to allow us to manipulate bytes 97 to 106
3. Get the 13th record or sector.
4. Move bytes 97 - 100 to 102 - 106.
5. Change bytes 97 - 100 to 01 04 16 40.
6. Put the modified sector to the file.

When finished a check on the 13th sector using Diskdump should show the following, where xx indicates don't care:
Before:
96 43 02 02 EE 7B xx xx xx xx xx xx xx xx xx

Move bytes 97 - 100
96 43 xx xx xx xx xx xx 02 02 EE 7B xx xx xx xx xx

Insert:O1 04 16 40 E3 03:
New Program
96 43 01 04 16 40 E3 03 02 02 EE 7B xx xx xx xx xx

Here is a simple Basic program to do the job:

```
10 CLEARl000: CLS:PRINT
20 INPUT"LOAD DISC WITH SPACEV/CND PROGRAM ";X$
30 OPEN "R", 1, "SPACEV/CMD"
40 FIELD l, 97 AS DD$, 1 AS DO$, 1 AS D1$, 1 AS D2$,
   1 AS D3$, 1 AS D4$, 1 AS D5$, 1 AS D6$, 1 AS D7$,
   1 AS D8$,1 AS D9$, 149 AS DX$
```

```
50 GET 1, 13
60 LSETD6$ = DOS: LSETD7$ = D1$: LSETD8$ = D2$: LSETD9$ = D3$
70 LSETDO$ = CHR$(1):  LSETDl$ = CHR$(4):    LSETD2$ = CHR$(22):
   LSETD3$ = CHR$(64): LSETD4$ = CHR$(227): LSETD5$ = CHR$(3)
80 PUT 1, 13
90 END
```

This technique will probably work with other programs that have been put to disc with an offset loader and don't work properly.
One that comes immediately to mind is MON3 It still uses the Dos keyboard and all sorts of funny things can happen if you try to use Mon3 from disc to load a System tape into low memory.
(Do I hear lots of voices saying 'so that's what was happening !')

A simple check with any of these programs would be to run Bounce first and if this cures the problem then add this little piece of code at the end.





The *BEST* in TRS-80s

Call The Right Stuff

Ask for Ian
The number is +61 416 184 893

That's The Right Stuff
And he's in Melbourne







http://ianmav.customer.netspace.net.au/trs80/

Solve-it, A Game

By Peter Stone

When it came to writing this article, it's fair to say that I suffered both a writing & a programming block for quite a while. To be fair, a large proportion of my TRS80 related resources, had to be thrown away when I was forced to downsize from a three bedroomed house to a single bedroomed flat, in the space of 20 days, two years ago. I didn't get rid of everything, I kept a number of items, which has formed the basis for a number of articles. Over the last two years, I've come to realise though that some things I needed got left behind, however I've also had a number of Aha moments while sorting through piles of things.

In volume 5 issue 3, I wrote about the time when TRS80 model1s provided me with a living, when I worked at the local Polytechnic. Something I didn't mention in the article, was the CTUK we set up. CTUK stood for **C**omputer **T**own **U**nited **K**ingdom. The idea was introduced by Personal Computer World magazine, & was based on an American idea known as CTUSA (Computer Town USA). Both the senior technician & myself talked it over, agreed it would be a good idea for a community based project, & got permission from the head of department to set it up. We had an advantage in that we had 16 machines we could use. A further plus was that we borrowed a set of computer based learning tapes from the local Tandy shop for evaluation. There were six tapes providing 12 lessons on aspects of BASIC programming, & my first task was to transfer the tapes to floppy disk. A further set of lessons covered disk basic, & came on floppy disk, these were also copied. I will gloss over the grind that was checking each of the taped lessons & converting them to work on a disk system. I think I can safely say that the CTUK was a success. We had at least 12 people a week turning up, plus one or two that had their own TRS80s & wanted to learn things like assembler. One of these guys, they were all male, actually bought his own copy of a stripped down version of LDOS, (we used Newdos80 V2), the version of LDOS came on one disk if I recall correctly. After going through all the lessons, everyone wanted more things to do. Some started writing their own programs, some needed a bit of prodding - which brings us to this article.

I was going through a box that had lain undisturbed for two years under a pile of other boxes, these contained the overflow of my kitchen equipment, & I had assumed that it to also contained kitchen equipment. But on checking I found it contained my Eagle annuals, (1958 to 1965, very collectable, the 1965 annual going for £45), & a couple of ring binders. In the back of one of the ring binders, was an envelope, & inside were listings & other documents from my days at the Poly. I found out later that my sister had pushed the envelope in the back of the ring binder in a whim, during the last push before I left the house.

I examined the listings, & found out that one was a program from the CTUK days

for the less confident of the people there. It had been decided, by the senior technician, that a program listing would be provided, which could be entered & modified by the programmer. My task had been to come up with something suitable. I remember at the time, thinking back to a game I had called Mastermind, manufactured by Parkers, I believe. I'm sure that some of you can remember how it was played, however I'll briefly recap. It consisted of a plastic board, lots of pegs in six colours, & two sets of smaller pegs in black & white. It required two people to play, one chose a code of four coloured pegs, & the other had to guess the correct sequence. You had hints as to how well you had done, with the white & black pegs, a black peg indicated a correct coloured peg in the right position, & a white peg indicated a correct coloured peg in the wrong position. The location of the white & black pegs bore no relation to the positions on the main playing area. My main gripe with the physical game was that you needed two people to play it, something that wasn't always possible, so a suitable game for a program. I recall starting to write it, & then coming across a similar program in a book, so I combined the two. The listing is shown below.

```
10 CLEAR 100:D=6:P=4:L=12
20 DIM G$(L),G(P),C(P),B(L),W(L)
30 RANDOM
40 GOSUB 1200
50 GOSUB 200:GOSUB 270
60 PRINT"guess number ";G;
70 INPUT A$
100 IF LEFT$(A$,1)="S" OR  LEFT$(A$,1)="s"  THEN 400
110 IF LEFT$(A$,1)="Q" OR  LEFT$(A$,1)="q" THEN 500
120 GOSUB 600
130 GOSUB 700
140 GOSUB 1000
150 IF B(G)=P THEN 2000
160 G$(G)=A$
170 G=G+1:IF G>L THEN 2200
180 GOTO 60
200 G=1:C$=""
210 RETURN
270 FOR J=1 TO P
280 R=RND(D)
290 C$=C$+MID$(STR$(R),2,1)
300 NEXT
310 PRINT "Code chosen"
320 RETURN
400 IF G=1 THEN PRINT "you've made no guesses yet":GOTO 60
410 CLS: PRINT,"summary"
420 PRINT "No.     Guess   Black   White"
430 FOR J=1 TO G-1
440 PRINTJ;TAB(7);G$(J);TAB(17);B(J);TAB(24);W(J)
460 NEXT:PRINT
```

```
470 GOTO 60
500 PRINT
510 PRINT"you've Quit "
520 PRINT:PRINT "My code was ....";
550 FOR K=1 TO 1000:NEXT
570 PRINT C$:PRINT
580 GOTO 2090
600 IF LEN(A$)<>P THEN 680
610 FOR J=1 TO P
620 R=VAL(MID$(A$,J,1))
630 IF R<1 OR R>D THEN 680
640 NEXT
650 RETURN
680 PRINT"Illegal entry.....Try again"
690 GOTO 70
700 B=0:W=0
710 FOR J=1 TO P
720 G(J)=VAL(MID$(A$,J,1))
730 C(J)=VAL(MID$(C$,J,1))
740 IF G(J)=C(J) THEN B=B+1:G(J)=0:C(J)=0
750 NEXT
760 FOR J=1 TO P: IF C(J)=0 THEN 820
770 H=0:FOR K=1 TO P
780 IF C(J)=0 THEN 810
790 IF C(J)<>G(K) THEN 810
800 H=1:G(K)=0:C(J)=0
810 NEXT K:W=W+H
820 NEXT J
830 RETURN
1000 B(G)=B:W(G)=W
1010 PRINT"Guess No. ";G;"--black=";B;" white =";W
1020 RETURN
1200 CLS
1210 PRINT"**** Solve it ****
1220 PRINT
1230 PRINT"The aim of the program is to figure out a";P;"digit code"
1240 PRINT"using the numbers 1 to";D
1250 PRINT"The number shown by"
1260 PRINT"'black' indicates correct digit(s) in the right position(s)"
1270 PRINT"'white' indicates correct digit(s) in incorrect position(s)"
1280 PRINT
1290 RETURN
2000 PRINT
2010 PRINT"you guessed the code in";G;"guesses"
2020 IF G<5 THEN B$="Outstanding!"
2030 IF G=4 OR G=6 THEN B$="Pretty good"
2040 IF G=7 THEN B$="Not bad"
```

```
2050 IF G=8 THEN B$="Not too great"
2060 IF G=>8 THEN B$="Bad!"
2070 PRINT"....That's ";B$
2090 INPUT "want to try again";A$
2100 IF LEFT$(A$,1)="Y" OR LEFT$(A$,1)="y"  THEN 40
2120 PRINT:PRINT"Coward":PRINT
2130 END
2200 PRINT
2210 PRINT"That's your Limit of";L;"Guesses"
2220 PRINT"My code was ";C$
2240 GOTO 2090
```

The program was designed to be easily modified, with everything placed in variables. The main variables are as follows.

D	Number of possible digits in each position of the code (ie a digit from 1 to D)
P	Number of positions in the code
L	Limit of number of guesses that can be made
G$	Array in which guesses are saved
G,C	Work arrays for analysing guesses
B,W	Number of black & white responses for this guess
R,H	Work variables
G	Counter of the number of guesses made
A$	User's response
C$	Secret code chosen by program
J,K	Loop variables
B$	String with message about the level of the user's performance

The main routines & subroutines are listed below.

10 - 50	Initialises variables, displays introduction, chooses secret code
60 - 140	Get guess from user, analyses reply, displays result
150	Determines if user made a correct guess
160 - 180	Saves guess, adds one to guess counter, determines if limit on number of guesses was exceeded
200 - 210	Subroutine to initialise variables
270 - 330	Subroutine to choose secret code & inform user
400 - 470	Subroutine to display summary of guesses so far
500 - 580	Subroutine to display secret code if the user quits
700 - 830	Subroutine to determine the correct number of black & white responses to the guess
1000 - 1020	Subroutine to display the number of black & white responses to the guess
1200 - 1340	Subroutine to display the title & introduction
2000 - 2130	Subroutine to rate the user's performance after a correct guess & ask about another game

2200 - 2240 subroutine to display secret code after user exceeds the limit on the number of guesses

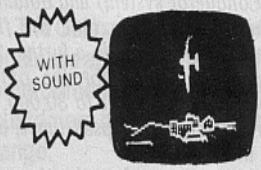
There are a number of modifications that can be made to this program. The most obvious would be to tidy up the display, using print@ statements, Another might be to use a more graphical display. The original game had a layout along the lines of CCCC wwbb, where C is a code peg, & w & b are black & white pegs. Another option might be to offer an easy, intermediate or advanced version of the game, along the lines of;

Easy D=3 & P=3,
Intermediate D=6 & P=4
Advanced D=8 & P=5

Of course, the guess limit would have to be modified to suit. When modifying or debugging the game, it is useful to add the line 305 Print C$, to the program.

ARCADE ACTION for TRS-80 Model I & III AND VIDEO GENIE

Dealer enquiries welcome



Dont forget if you can write good software for TRS 80, Pet, Apple, then we can offer top royalties for marketing both in Europe and the U.S.

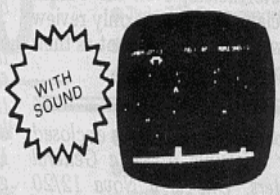
AIR ATTACK

Presenting a new dimension in Arcade games: The THIRD, Defend an East Coast Town against Fighters, Bombers and V.1's.

It's the best Arcade style game available, anywhere

£12.95

TRS-80 Model I & III. Level 2 Cassette 16k

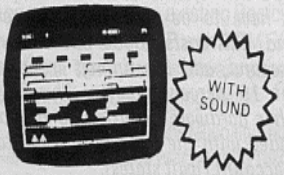


MOONBASE RESCUE

Somewhere in the Dagma system a group of Astronauts need your help. Can you steer your L.V.A. down through showers of meteors to the moons surface and return to dock with the Mother Ship

TRS-80 Model I Level 2 Cassette 16k

£8.95



FROG RACE

Afrogs life is a hard one; They've built a six lane Motorway right next to the crocodile infested river, that you must cross to get home!

Can you help a poor frog out, and maybe pick-up a girl-frog on the way.

£5.95

TRS-80 Model I & III Level 2, Cassette 4k & 16 k

*All prices include VAT and P/P.

ALGRAY SOFTWARE

Algray House, 33 Bradbury Street, Barnsley, South Yorkshire S70 6AQ

Telephone: (0226) 83199

140 PCW

IN MAV'S WORKSHOP
by Ian Mavric

BUILDING THE SUPER MICRO PART 4

In the last issue I talked about how to format the 80-track LDOS, LS-DOS and NEWDOS/80 boot disks for the "Super Micro", a maxed out Model 4 with 128K, speed up, RS232, hi-res graphics etc. It's a modern-day version of the C-U series of the same name[1.] This month I finish off the project just in time for its new owner to collect it.

Looking back through among the last issues of 80micro, and the Computer New 80 eras it was pretty obvious that people who were still using Model 4s still did two main things with them... communications and hi-res graphics. In 2012 this seems rather ambitious given that we are all used to the internet and colourful SVGA graphics, but in 1989 when all we had were dial-up BBSs and the choice for graphics was CGA and EGA on an XT or 286 computer, the Model 4 did indeed acquitted itself pretty well. Dial up modems were all the rage and in '89 2400 baud was a pretty good speed, in fact is an ideal speed for BBS work with a Model I or III, not just a Model 4. And for graphics, remember back then speedy colourful animation measured in FPS didn't exist, nor did MPEG, AVI, etc. so displaying hi-res pictures on the screen was a popular pass-time, and again the Model 4 excelled at that as well. With that in mind, lets put the finishing touches on the Super Micro.

4a. Installing an RS232C board

If your Model 4 started off as a 16K cassette machine, or a single disk drive 64K or a very early dual drive Model 4 (before the RS232C came standard with the computer), you'll need to install one, and fortunately its pretty easy to do. In fact the time to do it is when the motherboard is out and while you are installing the disk controller during a disk upgrade.

The Radio Shack RS232C kit (26-1148) comprises of the board itself, a power cable and a white spectra-strip to connect to the motherboard. Chances are after all these years if you find an RS232C board it might be missing its spectra-strip or the strips is broken and unusable - if so use the same method with gold-pin strip-headers and an IDE cable to connect the board to the motherboard. If the power cable is missing make up one which is a duplicate of the disk-controller power cable - they are the same.

Once installed the best way to test functionality of the board is to loop pins 2 and 3 together and load up your favourite terminal program (mine is MAL, but many others like Lynxterm, Fasterm, and Modem 80 are just as good) and see if you get an echo of what you are typing on the screen then you are done, set to communicate. If you are interested in downloading TRS-80 software from the internet into your Model 4, I suggest you watch this video, which uses the RS232C port to download software from a regular Windows 7 PC:
http://www.youtube.com/watch?v=EM_SjUpGX5k

4b.High resolution graphics board
History: High resolution graphics for the TRS-80 started with a very expensive kit by Radio Shack for the TRS-80 Model III (cat no. 26-1125). It had very little hardware support and was super expensive, so not many were sold. I have one in my Model III and it comprises of a large add-on board with 32K of graphics memory, and a CRT controller chip. In terms of complexity its fascinating because its designed as a add-on for a motherboard which wasn't designed with a hi-res graphics addition in mind at design stage. Only one program from Radio Shack properly support it: Dow Jones Market Analyzer (cat. no. 26-1606) and I recommend people to have a look at that program if they get a chance. A non-Radio Shack program which made good use of the Model III hi-res graphics kit was xT.CAD by Microware. Again, well worth a look.

When it came time to design the Model 4, Radio Shack had decided to design its motherboard for easy fitment of a hi-res graphics board, this also meant the graphics board could be designed and made much more cheaply than the Model III kit, sold for less and hopefully sell many more of them than the Model III version. Software support of the Model 4 hi-res board (26-1126) is therefore much better, and its backwardly compatible with the Model III board (when the computer is operating in Model III mode) which means those software packages which run on the hi-res Model III also run on the hi-res Model 4.

To install the board its really simple, open the Model 4 in the usual manner and turn it around so you are looking at the motherboard.

Model 4 (non-gate array 26-1068 and 1069):
Locate a long DIL connector marked J10 (down in the lower left-hand quadrant of the motherboard) and remove the jumper joining pins E14/15 together. Insert the graphics board into J10 and you are done.
Model 4 (gate array 26-1068A and 1069A) and 4D (26-1070):
On the gate array Model 4 and 4D there is a jumper on long DIL connector J12 joining pins 16 and 18 together - remove that jumper and install the hi-res board onto J12 and you are done.
Model 4P (all models non-gate array 26-1080 and gate array 26-1080A):
Locate long DIL connector J7, and jumper E4/E5 and remove it. Install the hi-res board onto J7 and you are done.

The quickest way to test the hi-res board is with Tandy's Model 4 diagnostic package which can be downloaded from David Cooper's web site[2].

4c. Quick'n'dirty speed-up mod for early Model 4s
In an issue of Sydtrug News[3] they mentioned how to make early Model 4s run up to 16% faster by carefully removing the PAL chip in U3, bending pin 7 out sideways so it doesn't go into the socket, and re-inserting the PAL chip into U3. What this does it takes away the wait-state which makes the Model 4 mode run

at approx 3.37MHz. We know the Model 4 is supposed to be a 4MHz machine and by taking away this wait state the computer speeds up to a proper 4MHz. Note: this only works on early Model 4s, of those I've tried it on about 1 in 4 doesn't work, so try it and see, if the computer doesn't start, or runs unpredictably, just bend the pin back and return the computer to normality.

IN MAV'S HARD DRIVE WORKSHOP

Lets make the Super Micro even more super - add a hard drive

Around the time the 2nd part of the Building The Super Micro series was published, a gentleman approached me saying it was just the sort of computer he'd been looking for and was keen to buy it. Like a lot of TRS-80 enthusiasts, he saw the potential to install four drives in a Model III or 4 was always a possibility, but had never gone to the trouble to research the requirements and proceeded to complete the project. A deal was eventually cut and the computer was picked up by its happy new owner. A few weeks later he contacted me again and said he'd always wanted a hard drive on a TRS-80 and wondered if I had any on hand to sell.

As luck would have it one showed up in eBay.co.uk and with the help of our very own Dusty, when I won the auction and had it sent to Dusty who would look the unit over for me, and then send it on to me in Australia.

The TRS-80 Primary hard drive comprises of 5 parts: 1) the Tandon MFM hard drive itself, 2) the controller (HDC), 3) a power supply from a Model 4, 4) the front panel assembly, and 5) the box which holds it all and integral cooling fan. On the side of some of the Tandon MFM hard drives are a couple of "G-force" sensors. They turn red if the hard drive has been subject to enough force (ie. dropped) to damage the platters or heads. Basically if they are red then its a pretty accurate indicator that the hard drive itself damaged, or if it does work it won't work for long. I had Dusty check for this condition and the sensors had turned red. I explained how to remove the hard drive (unplug its cabled, unbolt it from the chassis, and clip the three wires) and the resulting "empty" TRS-80 Primary hard drive is now 1/2 its original weight and much cheaper to send to Australia. I waited patiently and six weeks later the unit arrived in my Melbourne workshop. Thanks again to Dusty for his help.

MAV MODIFIED HARD DRIVES 101

TRS-80 hard drives are one of the most misunderstood and powerful pieces of hardware... they were very expensive at the time, and not many people know properly how to make them reliable. They were not even that reliable back when

they were new, but that was mainly due to the quite variable QC at Tandon with their MFM hard drive units. They were questionable back in the early 80s, and downright untrustworthy by 2012. To make a reliable TRS-80 hard drive that Tandon has got to go. It also helps if you buy the right hard drive for the application.

There are two types of hard drive, a Primary and a Secondary. The Primary, as the name implies, comes first. It has the HDC inside it and you always need one of these and only one in the system. Tandy's solution for people who needed more hard drive space was the Secondary hard drive. It doesn't contain the HDC, it only has a smaller power supply (a 35W one from a Model III as it happens), and is daisy chained to the Primary. You can add up to three Secondary drives to a TRS-80, but you must have a Primary drive to start with. You cannot add a Secondary drive to a TRS-80 on its own and make it work. It won't. I'm stating this clearly in case you have an opportunity to buy a Secondary hard drive, as they are often cheaper than Primary hard drives, but you need the Primary, as a minimum, to get a TRS-80 Hard Disk system up and running.

The power supply in a TRS-80 Primary hard drive is a Model 4 power supply and as such suffers from the same ailments as those fitted to a Model 4, namely that the filter capacitors crack up and smoke out and they get dry solder joins around the output connector. Before I even start one of these hard drives up I replace Caps C1, C2 and C13 and re-solder TB1 and TB2 with a nice hot soldering iron. On the subject of the power supply, if one comes from the USA which has a 120V unit it can be replaced with a 240V one by simply dropping it in but be aware of the case-cooling fan, its also A/C operated and the 120V version will blow up if 240V is applied to it. As a rule of thumb, if you are upgrading or downgrading the power supply in a TRS-80 hard drive, do the same with the cooling fan.

The front panel has lights for Activity and a switch with internal light for Protecting the contents of the hard drive. In their wisdom Tandy decided to use small incandescent globes which can be quite annoying to find when they blow. I have found that you can replace the globes with green or red LEDs but you quickly find there is way too much current and the LEDs burn out after a couple of hours of use, so install in series with the LED a 120ohm 1/2W resistor and it will work perfectly.

Finally, the Tandon hard drive itself, it can be replaced with most MFM hard drives if you know what to do with those three wires soldered to the Tandon hard drive. Rather than re-invent the wheel, all the information needed to put the wires in the correct places on the HDC board are contained in an article by Roy T. Beck in CN80[4]. For recommendations on which MFM drives work best in the TRS-80 hard drive, I find most half-height 5.25" MFM and all 3.5" MFM hard drives work very well, a big improvement over the ancient 5.25" MFM hard drives, the newer drives have larger disk capacity, quieter operation and quicker track-to-track stepping. It makes an already fast device even faster. If you are

looking for a list of good (and bad) vintage MFM hard drives I suggest you check the Red Hill web site[5].

SOFTWARE MAKES THE MOST OF ALL THAT SPACE

Tandy had Logical Systems Inc. write software drivers for their TRS-80 hard drives. In the Tandy "universe" hard drives are one of two capacities: either 5Mb or 15Mb. So the plain drivers will treat all hard drives as either 4-head 153-cyl (5Mb) or 6-head 306-cyl (15Mb). What happens if you install say a 44Mb hard drive using Roy's Three Wires modifications and attempt to use the Tandy drivers? You get either 5Mb capacity if you use TRSHARD3/DCT or 15Mb capacity if you use TRSHD15/DCT. That sure is a waste of space if you have a much larger 44Mb hard drive installed. What we need is a driver which allows us to input our own numbers of heads and cylinders.... then we could use any hard drive to its maximum capacity. Fortunately Misosys had the same thing in mind when they wrote RSHARDx, a hard disk drivers package for Model III and 4 computers running LDOS 5.3.x and LS-DOS 6.3.x You can find RSHARDx on Tim Mann's web site, and the comprehensive manual is also there[6]. I have used this package on all hard drives I restore and sell since 1992.

ianm@trs-80.com

REFERENCES:

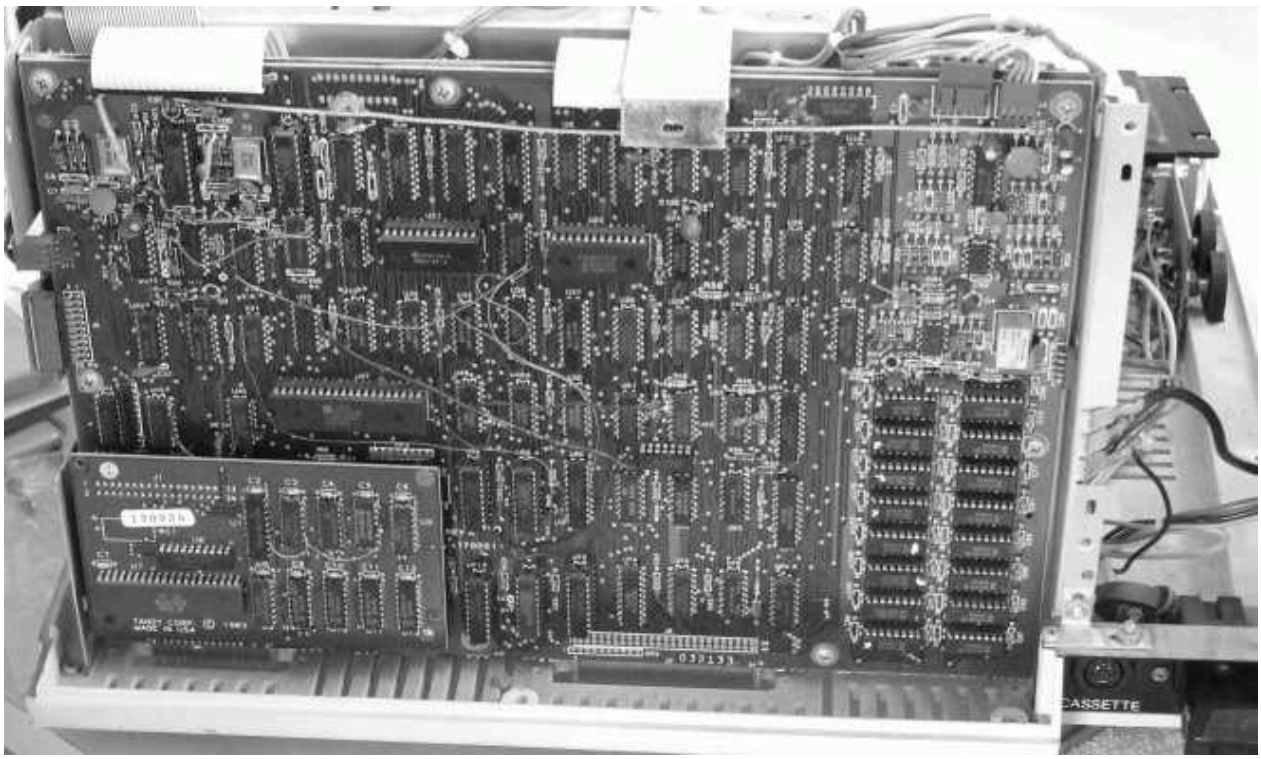
1. "Building The Super Micro", Computer User, February 1984 pp. 14-16; March 1984 pp. 14-18; April 1984 pp. 36-38; May 1984 pp. 26-30.
2. http://www.trs-80.us/trs80_m1_m4diag/downloads.html
3. Sydtrug News, Vol. 6, No. 7, March 1986, pp. 11 "Fast Fix for Mod 4 Slows"
4. Computer News 80, Vol. 4, No. 8, pp.19-23, "My Adventures with Hard Drives Part 8 - The Three Wires"
5. http://www.redhill.net.au/d/d-a.html
6. http://tim-mann.org/misosys.html



Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.



Completed 4 drive M4 Super Micro with Hard Drive



HI-Res board installed on a M4

(CONTINUED FROM PAGE 1)
NEW TO BASIC PROGRAMMING, SHOULD BE VERY HELPFUL. YOU CAN USE THE LINK I'VE SET UP ON OUR WEBSITE TO CHECK IT ALL OUT.

IRA HAS JUST ANNOUNCED THAT A NEW CONVERSION PROGRAM, 'TELEDISK TO DMK' IS AVAILABLE, WRITTEN BY AARON BROCKBANK. IF, LIKE ME, YOU'D NO IDEA WHAT TELEDISK FORMAT WAS, I'VE FOUND ON THE WEB, A FEW WORDS TO EXPLAIN ALL!

TELEDISK WAS A POPULAR DISK ARCHIVAL TOOL AROUND 1990. THERE ARE A NUMBER OF ARCHIVES, I.E. UPDATE AND METALAB, THAT CONTAIN TELEDISK IMAGES OF DEC RX50 DISKS FOR RAINBOWS, DECMATES, AND PRO3XXS.

TELEDISK WAS ORIGINALLY RELEASED AS A SHAREWARE PROGRAM BY SYDEX. ITS STILL AVAILABLE AS A DISK ARCHIVAL TOOL IN A NUMBER OF MSDOS PROGRAM ARCHIVES. THE LATEST VERSION SEEMS TO BE 2.12, WHICH CONTAINES FILES DATED 12/12/1990. AS NEAR AS I CAN TELL SYDEX STILL HAS A WEB SITE BUT THE TELEDISK PRODUCT WAS PURCHASED BY NEW TECHNOLOGIES, INC IN 2000. THEY HAVE A TELEDISK WEB PAGE, BUT ARE NOT EVEN MARKETING THE CURRENT VERSION, 2.22, TO THE GENERAL PUBLIC. IT IS DEFINATELY NO LONGER A SHAREWARE PROGRAM!

I'M PLEASED TO REPORT THAT I'VE MANAGED TO RE-CONTACT LAURIE SHIELDS AFTER MANY YEARS. FOR ANY OF YOU WHO DON'T REMEMBER LAURIE, HE WAS THE CHAP WHO BROUGHT US THE 'ZEN' EDITOR ASSEMBLER IN MANY GUISES AS WELL AS A MASSIVE NUMBER OF OTHER INSPIRATIONAL IDEAS, PROGRAMS AND Z80 ASSEMBLER TUTORIALS THROUGHOUT THE COUNTRY AT MANY ONE-DAY WORK SHOPS HELD BY NATGUG. AS A POINT OF INTEREST, IT WAS LAURIE WHO FIRST INTRODUCED ME TO ASSEMBLER PROGRAMMING TOO!

ANYWAY, DOWN TO BUSINESS: WHAT IN STORE FOR YOU IN THIS ISSUE.
AT THE READY PROMPT MAKES A RETURN, WITH TIPS ON ARROW KEYS AND DEFINE FUNCTIONS.

RECENTLY, ON THE TRS-80 FORUM, 8" DISK WERE THE FLAVOUR OF THE MONTH, SO I'VE INCLUDED SOME MODEL 2 8" DISK TIPS FROM DERRICK ROWE. LIKE DERRICK, I TOO FOUND THE HARDWARE SOMEWHAT 'FLAKY'.

THERE'S A TANDY GAMES REVIEW FROM PCW.

I'VE FOUND AN ARTICLE ORIGINALLY FROM NATGUG NEWS, IN 1981, BY LAURIE SHIELDS ENTITLED 'SPACEV'. IT REALLY SHOWS THE BRILLIANCE OF THE GUY GETTING AROUND LOADING PROBLEMS FROM CASSETTE TO DISK, WITH THE MOST BASIC SET OF SOFTWARE TOOLS. WELL WORTH A READ.

PETER STONE HAS, YET AGAIN, COME UP WITH A BRILLIANT BUT DIFFERENT TYPE OF ARTICLE, A GAME, ENTITLED 'SOLVE-IT'.

MAV HAD PRODUCED PT.4 OF 'BUILDING THE SUPER MICRO'. HE'S ALSO WRITTEN ABOUT UPGRADING AN OLD TANDY HARD DRIVE WHICH TO MY MEAGRE DIY ABILITY IS QUITE ASTOUNDING!

I HOPE YOU FIND BOTH THE ARTICLES AND ADVERTS ENJOYABLE-
TILL DECEMBER,
TAKE CARE

DUSTY



ComServe
computer shop



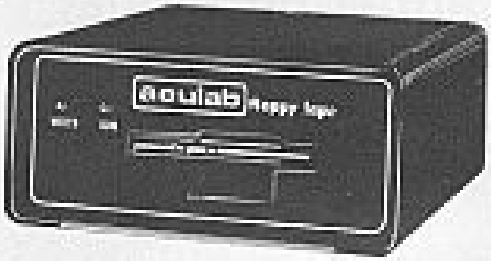
"HOBBYIST" GENIE

VIDEO GENIE with 16K RAM, ●3 manuals ● Leads ● demonstration tape ● cover ● head cleaner ● demagnetiser ● 2 joysticks ● sound kit with speaker installed ● demonstration games tape for sound and joysticks and graphics. 1 year guarantee.

£369.00 *inc VAT and carriage*

ACULAB SYSTEM

VIDEO GENIE with 32K RAM ● 5 manuals ● leads ● demonstration tape ● cover ● cleaning tape ● Aculab drive ● power supply ● 11 cartridges ● extended basic with disk like commands ● compress, merge and renumber ● 12" monitor ● expansion lead.



£666.00 *inc. VAT and carriage*



DISK SYSTEM

GENIE with expansion box, 48K RAM ● 5 manuals ● twin 40 track drives ● 35 track compatible ● cover ● cleaning tape ● cleaning disk ● 10 diskettes ● Newdos 80 operating system ● 12" monitor.

£1190.00 *inc. VAT and carriage*

CENTRONICS 737-2	EPSON MX80FT	GP 80
£399	£430.	£230
inc VAT, cable & carriage. Proportional spaced printer.	inc VAT & carriage. Quality printer with graphics	inc. VAT & carriage Hobby printer with Hi-Res graphics

ComServe

98 TAVISTOCK STREET, BEDFORD, BEDFORDSHIRE
TELEPHONE (0234) 216749

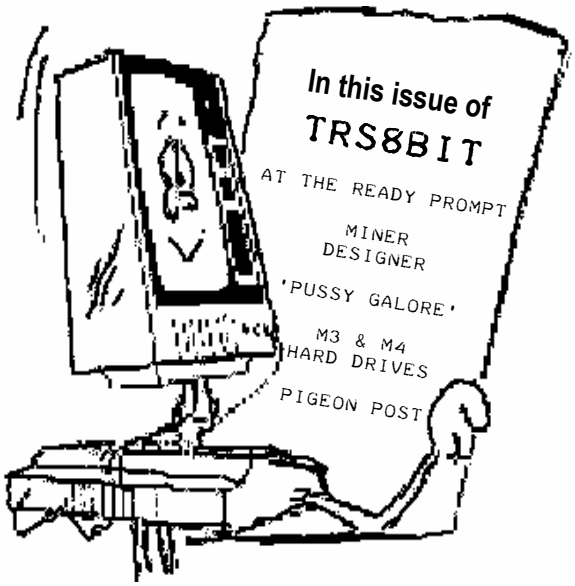
TRS8BIT



WELL, HERE WE ALL ARE WITH YET ANOTHER XMAS RUSHING TOWARDS US! 2012 THIS TIME, BUT IT DOESN'T SEEM 2 MINUTES SINCE THE LAST ONE EH!

IT PRACTICALLY GOES WITHOUT SAYING, **DEE & I WISH YOU ALL A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR**! WE LOOK FORWARD TO YOUR CONTINUED SUPPORT AND INTEREST THROUGH 2013 (AND ONWARDS OF COURSE!).

THIS ISSUE COMPLETES OUR 6TH YEAR, AND TO BE HONEST, I'M ANTICIPATING VOLUME 7 WITH AS MUCH, IF NOT MORE, ENTHUSIASM, AS I HAD FOR VOLUME 1! IT SEEMS INTEREST IN ALL THING 'TANDY' IS CONTINUING TO GROW AND ANYTHING WE CAN DO TO HELP IT ALONG IS JUST FINE WITH ME!



MAV'S BEEN BEAVERING AWAY, PRODUCING A NUMBER OF VIDEOS, WHICH HE HAS PLACED ON 'YOUTUBE'. YOU CAN CHECK OUT HIS WEBSITE FOR FURTHER DETAILS. THEY REALLY ARE 'FUTURE CLASSICS'. HE SEEMS TO PULL THESE OLD MICROS APART, BUT, MOST OF ALL, HE MANAGES TO GET THEM BACK TOGETHER WITH NO EXTRA BITS LEFT OVER!

EBAY UK HAS BEEN RELATIVELY QUITE FOR THE PAST 3 MONTHS AS FAR AS TANDY HARDWARE IS CONCERNED. THE MAIN ADVERTS HAVE BEEN FOR ASTRONOMICALLY PRICED CASSETTE SOFTWARE.

IRA HAS HAD QUITE A FEW CONTACTS FROM AUTHORS PLACING SOFTWARE IN THE PUBLIC DOMAIN. AS ALWAYS, HIS SITE IS WELL WORTH CHECKING OUT WHEN YOU HAVE TIME.

I'VE HAD TWO RECENT REQUESTS FOR HELP EXTRACTING SOFTWARE FROM MODEL 1 DISKS, TO PC FORMAT, IN THE LAST FEW WEEKS. THE ONLY HELP I'VE BEEN ABLE TO OFFER IS TO DIRECT ANY REQUESTS TO EITHER IRA IN THE STATES OR TO MAV IN AUSTRALIA. I'M NOT AWARE OF ANYONE IN THE UK WHO HAS A 'CATWEASEL' SETUP. IF YOU HAVE AND CAN HELP, PLEASE CONTACT ME AND IN FUTURE I'LL PASS ON YOUR DETAILS.

HAVING HAD AN EMAIL NATTER WITH BOTH MAV AND PETER STONE, I'VE DECIDED TO CHANGE, SLIGHTLY, THE FORMAT OF TRS8BIT IN ORDER THAT 'TECHNICAL' ARTICLES CAN HAVE A MORE UP-TO-DATE FONT INSTEAD OF THE TELETYPE

(CONTINUED ON PAGE 19)

AT THE
READY
>
PROMPT



THESE FEW WORDS CAME ABOUT THROUGH PURE COINCIDENCE! HOWEVER, THE FURTHER I INVESTIGATED, THE MORE INTERESTING THINGS BECAME!

IT ALL STARTED WHEN I CAME ACROSS A 'LETTERS TO THE EDITOR' ITEM IN NATGUG NEWS FROM ANDREW DONALD. IT JUST AMUSED ME TO HAVE A SOLUTION TO A PROBLEM, I DIDN'T KNOW I HAD!
ANDREW WROTE :-

"In Practical Computing for October a small routine was given which enabled cassette data files to be written and read (PRINT# and INPUT# functions), with an increased speed.
A factor of 2 was claimed. It works by writing only 8 bytes as the data item leader rather than the 256 specified by the ROM.
I was recently asked to have a look at the routine since it did not appear to work.
I found that the routine was written for the old ROMs and should work as stated for those.
To enable it to operate with new ROMs it is necessary to change the value of 150 appearing in line 30 to 121.
It seems a handy routine for anyone using tape based data files)."

THAT, IN ITS SELF, WAS INTERESTING ENOUGH, BUT ANDREW WENT ON AND WROTE :-

"If we have any cassette based ROM zappers in the group they might be interested to learn that exactly the same effect may be achieved by changing the FF at address 0288H to 07.
On the subject of ROM zapping I am aware that a number of people have now replaced their ROMs with EPROM's. If your system has a pound sign whilst in BASIC, then this may be used in place of the floating $ in the Print Using function by changing 29 to 60 at each of the following addresses. 102C, 10EC, 2D2E, 2D3D.
This is a function which we in Britain have almost abandoned simply because we don't use dollars (yet)."

I CAN'T REMEMBER HEARING OF ANYONE REPLACING THEIR ROMS WITH EPROMS AT THE TIME, BUT AS THE EMULATORS WE NOW USE PICK UP THE ROM DATA FROM FILES, IT SEEMS A REMARKABLY GOOD AND RELATIVELY SIMPLE IDEA TO IMPLEMENT.

AS LUCK WOULD HAVE IT, I JUST HAPPENED TO HAVE A COPY OF THE ACTUAL EDITION FROM OCTOBER 1981 OF 'PRACTICAL COMPUTING'. IT'S A BIT 'DOG EARED' I MUST ADMIT, BUT HERE IS A PRINT OUT OF THE PROGRAM, WHICH, BY THE WAY, WAS WRITTEN BY FRANK JOHNSON.

```
0 REM** PRINT# TO TAPE DRIVER ROUTINE **

10 CLEAR500:B#=PEEK(16562)*256+PEEK(1656
1):IFB#>32767THENB=-65536+B#ELSEB=B#:REM
EMBER TO RESERVE AT LEAST 15 BYTES BEFOR
E RUNNING: MORE DATA IS NEEDED TO ENABLE
 PRINT# TO DISK AS WELL AS TAPE
20 FORA=B-19TOB+15:READC:POKEA,C:IFPEEK(
A)<>CTHEN40ELSENEXT:B%=B-19:POKE16842,19
5:POKE16843,PEEK(VARPTR(B%)):POKE16844,P
EEK(VARPTR(B%)+1):PRINT#-1,"A":REM THIS
SECTION SETS UP PROGRAM WHICH DELETES TH
E PROGRAM & RETURNS TO READY
30 DATA42,177,64,35,34,203,65,17,205,255
,25,34,160,64,205,77,27,195,24,26,254,35
,192,193,205,254,1,6,7,205,137,2,195,150
,32:REM  THE PROGRAM ALWAYS RESIDES IN M
EMORY FROM ONE BYTE PAST RESERVED ADDRES
S
40 PRINT"NOT ENOUGH MEMORY RESERVED FOR
THIS ROUTINE"
```

At 18, star-programmer Matthew Smith says he's over the hill.
Peter Connor, 27, takes notes and worries about his pension.

MATTHEW SMITH
THE MINER DESIGNER

When I arrived at Liverpool's Lime Street Station one grey December afternoon in 1983 the first thing I noticed about Matthew Smith was that he wasn't wearing any socks. Hadn't he earned anything from writing the top-selling Spectrum game *Manic Miner*? Couldn't he afford socks?

As I later found out, my worries were unfounded: he certainly had made some money. But Matthew is not the whizz-kid breed of programmer, the adolescent brat who trades in his push-bike for a Ferrari on his 17th birthday.

Instead, he is an unassuming, but confident, teenager who is ready for even more success as *Jet Set Willy*, the next instalment in the *Manic Miner* saga, hits the nation's micros.

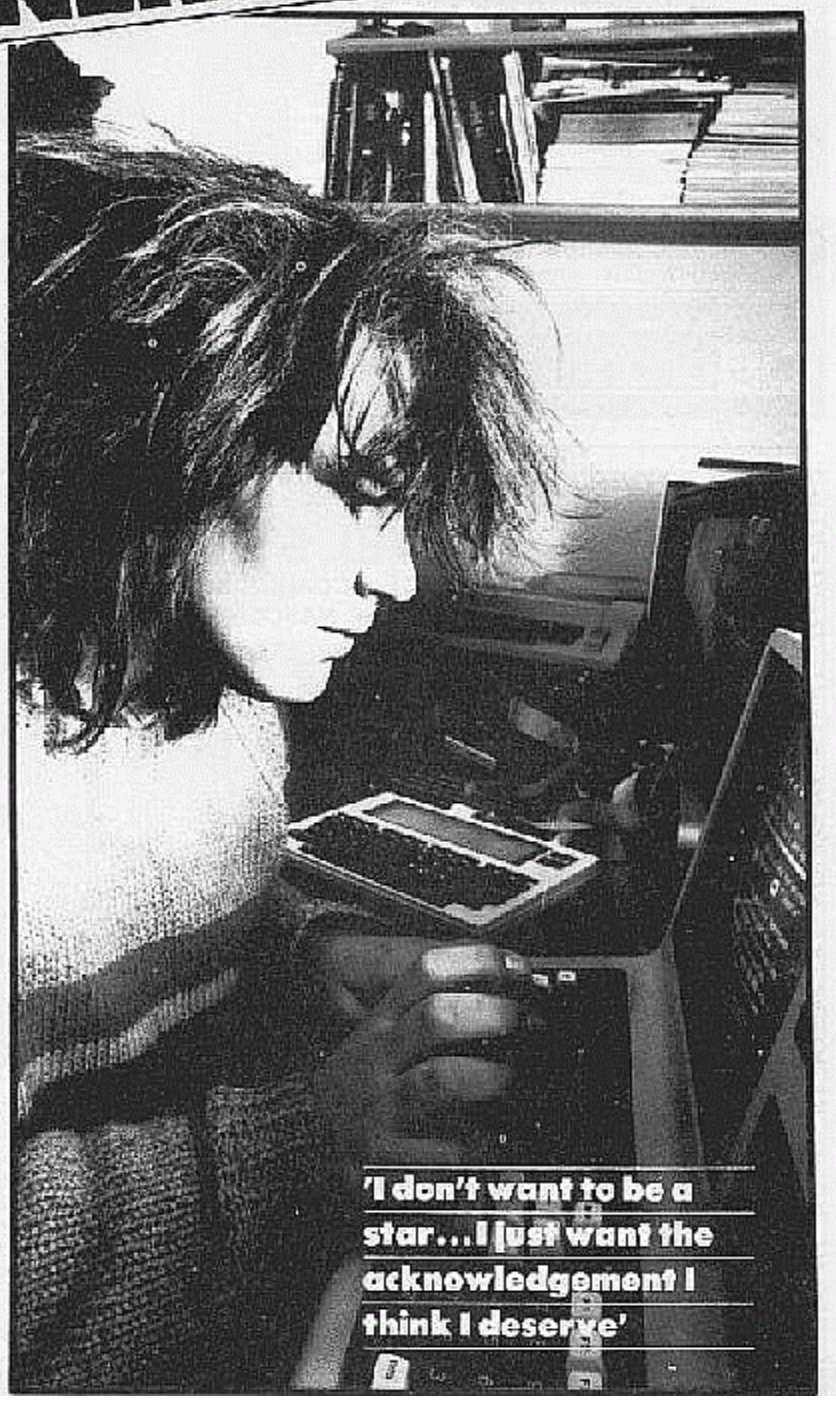
At the offices of Software Projects, the new company in which Matthew is a major shareholder, he told me how he'd first become interested in computing: 'I'd been keen on electronics, building transistor radios and so on, and five years ago my parents gave me a Tandy for Christmas. It only had 4K of memory and unextended Basic, but it got me hooked.'

Like many others involved in Liverpool computing Matthew joined the Tandy users' group based in the local dealer's shop. 'We used to hang around after closing,' says Matthew. 'Fortunately the manager was an enthusiast as well.'

He pursued his interest at home too, and at the age of 14 wrote his first commercial program, a version of *Galaxians* called *Delta Tower One*. 'It took me weeks to write it. I think it sold 13 copies and I made about £50 profit.

'It was just an average game,' he added modestly.

Average or not, the game showed him that there was a living to be made in programming. Through his contacts in the Tandy shop he began to write business and accounting programs for companies in the Liverpool area. It was this freelance work which saw him through the period after leaving school



in October 1982.

'I didn't like school,' says Matthew, 'and it didn't like me.' Still, he managed to pick up nine or 10 'O' levels.

It was only in April of last year that Matthew first got the Spectrum which has brought him fame and fortune. By this time he'd become associated with Bug-Byte, and it was for them that he wrote *Styx*, his first successful game program. In this perilous journey to the Hades of Greek mythology an experienced eye might be able to detect the germ of *Manic Miner*.

Before starting on the Herculean task of creating *Manic Miner* Matthew went on an 'inspirational skiing trip', which seems odd since he describes himself as 'naturally lazy'. It must have done the trick, though, because he was able to come back and spend eight solid weeks on the new program.

He followed a daunting routine, getting up in the evening, working until lunchtime, sleeping for a few hours and then going back to the computer. It doesn't sound like laziness to me.

Matthew's approach to programming is straightforward: 'When I got the Spectrum I worked out the limitations of the machine and then decided to write a program to go to the limits.'



'I'm not interested in flash cars, the money has just made it easier to be lazy.'



He actually wrote *Manic Miner* on a Model 4 Tandy with 5 Megabyte hard disk and then transcribed it on to the Spectrum. 'It's more efficient like that,' says Matthew, 'because the Tandy and the Spectrum both have the Z80 processor, and there's no waiting for loading and saving.'

I asked him if he had had any artistic help with *Manic Miner*'s impressive graphics. 'No,' he replied, 'I drew them out myself on graph paper. Art was one of my better subjects at school.'

Manic Miner, though, was not originally Matthew's idea. It came first of all from Alan Maton, his partner in Software Projects, who suggested the idea of a figure tunnelling down. Matthew turned this upside down and the result was the game all Spectrum owners know and love.

Their first thought for a name was *Miner 49'er* but, like many great inventions, someone else got there just before them. Whatever the name, Matthew was confident of success: 'I was pretty sure it was a winner as soon as I'd finished it.'

He took a two-month holiday after *Manic Miner*'s completion in which, he says, 'I just went to bed at nights.' But there has been little rest for him since his return to Liverpool. In September a follow-up to *Manic Miner* was 'discussed over meals in various restaurants'. And so, over the tablecloth, *Jet Set Willy* was born.

Matthew has found things rather difficult with *Willy*. 'To program you really have to enjoy sitting at a computer all night,' he said, 'but I'm getting too old to manage the hard regime.'

However, despite the weight of his 18 years, Matthew is pleased with what he's achieved in *Jet Set Willy*. 'I think I've got about three times as much into the program as in *Manic Miner*.' He describes it as being in the style of an 'arcade adventure' with graphics developed on his own special software.

Again he's confident of success: 'It'll reach number one,' he predicts, but he doesn't think it will stay there as long as *Manic Miner* because 'there's too much good competition around nowadays.'

Jet Set Willy is bound to bring Matthew even more publicity, yet the role of ace programmer is one he's unwilling to play. 'I don't particularly want to be a star,' he says. 'I just want the acknowledgement I think I deserve for writing good programs.'

Part of this recognition is likely to be a lot more money – to go with the £20,000 or so he has already banked from *Manic Miner*. But there won't be any extravagances: 'I'm not interested in flash cars and that kind of thing,' he says. 'The money has just made it a bit easier to be lazy ...'

After another skiing trip Matthew expects to start thinking about a follow-up to *Jet Set Willy* – perhaps to be called *Willy Meets the Taxman*. He also intends to start programming on the Spectravideo and other machines using the MSX operating system; another massive market for miner Willy.

Looking further ahead Matthew's future is uncertain. He is sure, though, that he won't always be a games designer. 'Programming games won't keep me going for the rest of my life,' he says.

So what will? He hopes to take a closer interest in the business side of Software Projects and doesn't rule out the possibility of studying Computer Science 'when I'm too old to program'. That might not be too far away since he says 'I'm a bit over the hill already'.

And what about those socks? Obviously Matthew Smith has enough pocket money to buy a few thousand pairs.

'I only wear them occasionally,' he says, 'when it's very, very cold.' Somehow, writing *Manic Miner* seems to have warmed the parts that other computer games don't reach. ●

THE XMAS CAT

ONCE AGAIN, IT'S TIME FOR DEE AND I TO SEND YOU OUR ANNUAL CHRISTMAS CARD.

AS USUAL, I'VE TRIED TO THINK OF AN ORIGINAL PRINT, SO THIS YEAR, WE'RE SENDING THE CARD TO YOU IN THE SHAPE OF THE 'ORIGINAL', LUCKY, XMAS CAT!

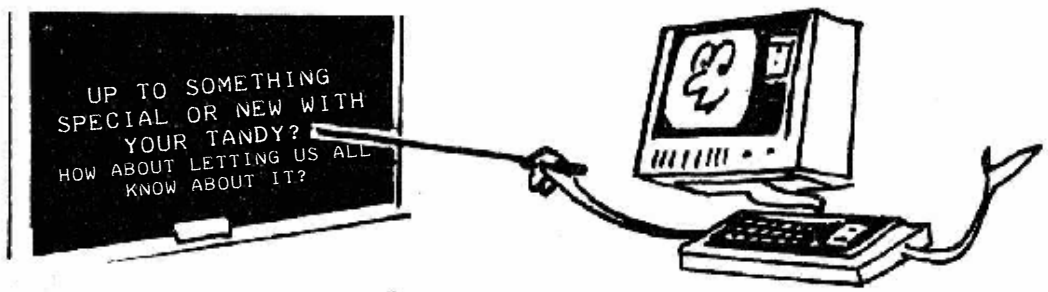
YES! I KNOW, AS ALWAYS, YOU'LL NEED TO SQUINT AT IT A BIT BUT AT LEAST THE THOUGHT IS THERE EH!

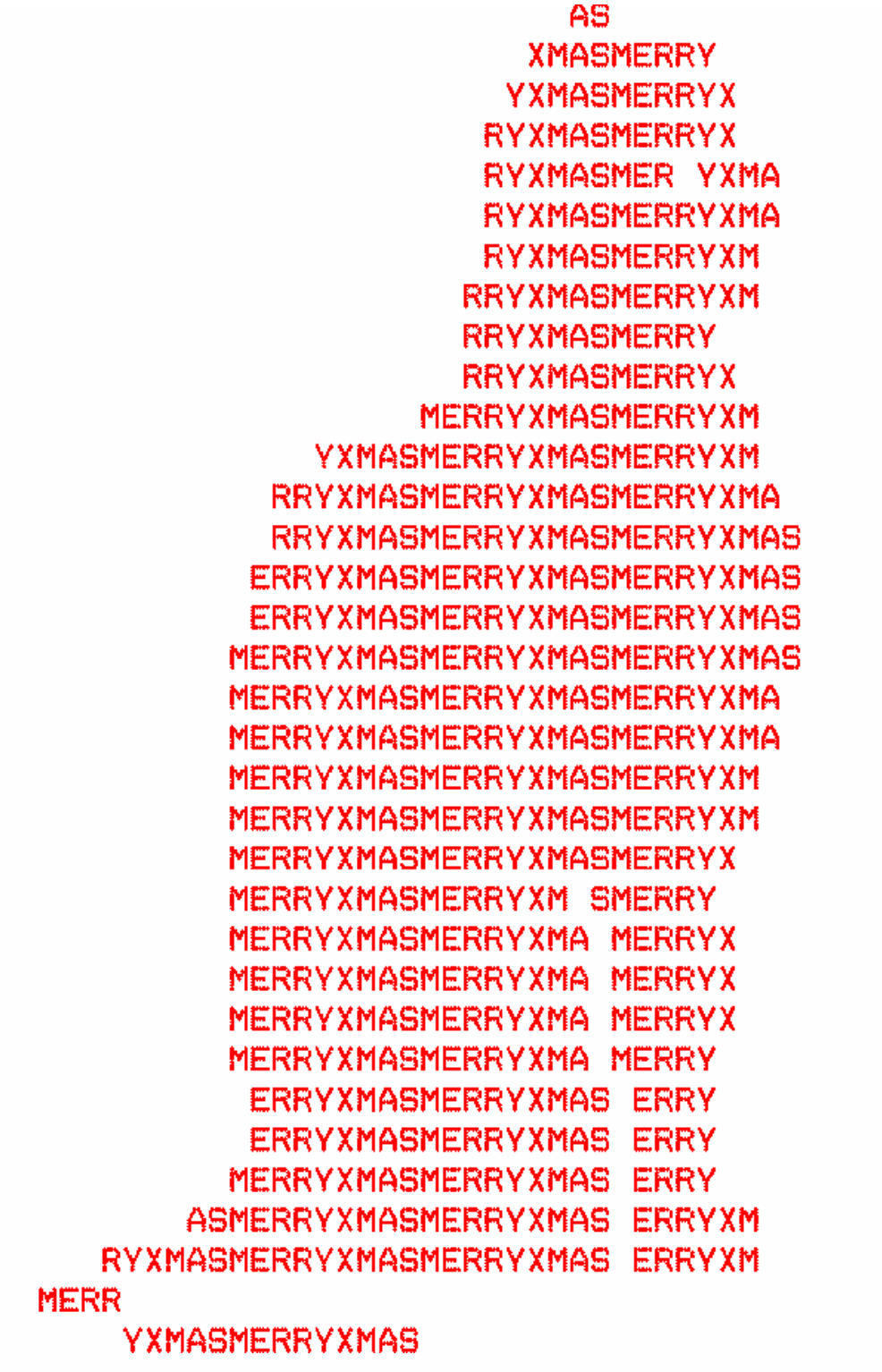
THERE'S A COMPLETE PROGRAM LISTING BELOW, WITH 'PUSS' ON THE NEXT PAGE. IT'S MORE OR LESS SIMILAR TO LAST YEARS PROGRAM, (APART FROM THE DATA STATEMENTS OF COURSE!) IT WORKS FINE ON MY 'REAL' 16K SYSTEM AND, OF COURSE, MATTHEW'S SUPER EMULATOR TOO.

```
10 CLEAR500
20 M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMAS"
30 GOSUB 210
40 READ S
50 IF S=0 GOSUB 150
60 READ L
70 T$=MID$(M$,S,L)
80      X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256

90 REM THIS AND THE NEXT LINE ARE NOT NEEDED ON SYSTEMS
                        WITH 16K OR LESS

100      IF X>32767 THEN X=X-65536
110          FOR Y=1TOL
120          POKE X+S+Y,ASC(MID$(T$,Y,1))
130          NEXT Y
140 GOTO 40
150 LPRINT P$
160 GOSUB210
170 READ S
180 IFS=0THENLPRINTP$:READS
190 IF S=99 THEN END
200 RETURN
210 P$=STRING$(45,32)
220 RETURN
230 DATA 26,2,0,24,9,0,23,11,0,22,12,0,22,9,32,4,0
240 DATA 22,14,0,22,13,0,21,14,0,21,12,0,21,13,0,19,16,0
250 DATA 14,21,0,12,24,0,12,25,0,11,26,0,11,26,0,10,27,0
260 DATA 10,26,0,10,26,0,10,25,0,10,25,0,10,24,0,10,16,27,6,0
270 DATA 10,17,28,6,0,10,17,28,6,0,10,17,28,6,0,10,17,28,5,0
280 DATA 11,17,29,4,0,11,17,29,4,0,10,18,29,4,0,8,20,29,6,0
290 DATA 4,24,29,6,0,1,4,0,5,14,0
300 DATA 99
```





JUST TO WISH ALL OUR READERS A VERY
MERRY CHRISTMAS AND A HAPPY NEW YEAR

THANKS FOR YOUR HELP, SUPPORT AND
ENCOURAGEMENT THROUGHOUT THE YEAR(S).

IT REALLY WOULDN'T BE THE SAME WITHOUT YOU!

AVAILABLE ON CASSETTE

★ ASTROFACT ★

★ FOR THE TRS-80 LEVEL II ★
REQUIRES 16K MEMORY

: A NEW PROGRAM IN BASIC AT A SPECIAL INTRODUCTORY PRICE :

ONLY

£6.00 (**U.K.** inclusive of postage, packing and VAT. | **OVERSEAS** inclusive of postage and packing. No VAT required)

BUY NOW FOR XMAS

IDENTIFY THE PLANETS AND STARS
PREDICT MOON'S POSITION ON YOUR OWN SKYLINE
IS IT A FIRST QUARTER OR A FULL MOON NEXT SUNDAY?
WHERE IS VENUS IN THE SKY ON CHRISTMAS DAY?

Astrofact requests the input of year, month, day, time, also asks for latitude (North or South of Equator) and longitude (East or West of Grenwich) Astrofact then calculates and displays for each of the planets.

MERCURY: VENUS: MARS: JUPITER: SATURN: URANUS: NEPTUNE: PLUTO

The following data which is displayed sequentially as calculated and then finally a complete list in tabular form:
Altitude in degrees above (or below) the horizon: Azimuth (compass bearing)
Right ascension: Declination: Phase: Apparent magnitude (brightness)
Distance from the Earth; for any date or time 1975-2001 incl.
Also **SUN** as above plus angular diameter,
And **MOON** as above plus angular diameter, and "age" (eg first quarter)

Additionally, Astrofact gives Altitude any Azimuth of an coordinates of right ascension and declination input by you, and displays also right ascension, declination, azimuth and altitude for the most important stars and star groups, examples: Sirius, Pleiades (Alcyone), Leo (Regulus), etc.

Orders despatched first class post with explanatory leaflet enclosed.

D. H. WOOLLEY (QUORN) LTD,

UNIT 2, WINDMILL ROAD,
LOUGHBOROUGH,
LEICESTERSHIRE,
ENGLAND.

AN IDEAL PRESENT FOR ALL AGES AND EXCEPTIONAL VALUE FOR JUST SIX POUNDS

TELEPHONE: (0509) 217215.
TELEX: 34338.

IN MAV'S WORKSHOP
by Ian Mavric

EXPANDED DISCUSSION ON TRS-80 MODEL III AND 4 HARD DRIVES

Last issue[1,] I completed my series on Building The Super Micro with a short summary about how to add a hard drive to a TRS-80 Model 4. I received many favourable comments from people and a few requests to expand on these amazing devices, so this issue I'll delve more the history of the hard drive as applied to TRS-80 Model III and 4 computers. First... some history:

IN THE BEGINNING THERE WAS NO HARD DRIVES

When the Model I came out in August 1977 it had cassette storage and after it was on the market for about 5 minutes Radio Shack realised if they wanted people to take their computer seriously it would need to be able to use the newly-invented 5.25in "Mini" floppy disk drive. So by mid 1978 the Expansion Interface and disk drive was on the market, and for a few years people were happy with that. Also around this time a company called Shugart invented a hard disk drive which was a much miniaturised version of a mini/mainframe computer hard drive, which could be easily interfaced to small computers, had a capacity of about 5Mb, but had one serious drawback - price.
By 1982 these hard drives had been perfected and the price reduced to a level that serious businesses could consider affording. Also around this time the level of sophistication in TRS-80 software and the amounts of data some people needed to store was reaching a point where floppy disk was no longer convenient or fast. Radio Shack noted this and produced a hard drive for the Model III, it held 5Mb, was very fast, and cost a cool four thousand dollars[2.] It carried the catalogue number 26-1130 and is a big black metal box proudly proclaiming "FIVE MEG DISK SYSTEM".
In 1985 the 5Mb drive was looking small so it was phased out and replaced with a 15Mb unit in a shiny-white metal box to match the Model 4, and this became the popular 26-4155W "FIFTEEN MEG DISK SYSTEM"[3.] There were other drive capacities sold (10Mb, 12Mb, 35Mb and 70Mb) but this discussion will focus on the 5 and 15Mb units, since they are the most commonly found these days.

WHAT TO LOOK FOR WHEN BUYING A TRS-80 HARD DRIVE

Hard drives do come up on eBay and vintage computer forums from time to time, the most common being the 26-1130 5Mb Primary drive and the 26-4155W 15Mb Primary drive. You can basically use any TRS-80 Primary drive and they are easy to identify straight away by their keyswitch to power-up the unit. Avoid Secondary drives (easily identified by 2 red lights, one green light, and no power keyswitch) unless you already have a working Primary drive, and steer clear of those big 8Mb hard drives from the TRS-80 Model II business computer - they

can't be made to work in a Model III/4 application.

Original owners of hard drives paid a premium for the privilege of owning one, often costing more than the computer they hooked them up to. For this reason although they were a relatively low-volume seller for Tandy. Those who bought them were often reluctant to throw them away, so it always surprises me when they turn up out of peoples attics and garages. Most will not be stored properly, and a lot of work can be needed to get one up and running. They key may be lost, the cables or connectors may have corrosion, and chances are the software is also lost. You need to factor all this in when buying a used hard drive and whether it is easier to just buy a refurbished one from a reputable seller. For those with a technical bent, there is nothing really super complicated inside the unit and that anyone who can use a soldering iron, cable crimper and oscilloscope can get them up and running successfully. Some of the stuff I'm about to cover now was already mentioned last issue, so if you are feeling a bit of deja vu, that's why.

IT'S MISSING THE KEY, CABLES, AND SOFTWARE - BUT I GOT THE DRIVE AT LEAST!

A lot of drives which are pulled out of attics are missing everything. What arrives is the hard drive box, and nothing else. Where do we start if the key is missing? Fortunately the cabinet is held together with three Philips-head screws along the back, and once they are out you can see that the keyswitch is just an easily replaceable assembly, so if the key is missing I replace it with Jaycar's 240V mains keyswitch (cat. no. SM1031, $12.95)[4] and now you can power the unit up. For the interface cable you need to make one up, it has a 50-pin card-edge connector on one end and a 50-pin DIL female socket on the other end, and about 90cm of 50-wide ribbon cable. If you are stuck for one I make them and sell them, however most decent electronics supply shops carry the parts.

Finally for the software to drive the unit, I suggest Misosys RSHARD5/6[5], which I have used for 20 years now, it works on any size hard drive, and there are even patches available for those who wish to use a hard drive, via an adaptor (26-1132) on a Model I. The original RS drivers from the early 1980s are also useable, but they don't have the flexibility of RSHARD. Once you have the key, interface cable, power cable (luckily its just a standard IEC cable), and software, we are now ready to delve inside the drive and make some checks and upgrades prior to using the drive. Do these now to save many headaches down the road.

THE HARD DRIVE ITSELF

Some call it the Winchester, others call it the Bubble, I prefer to call it the MFM hard drive, but whatever you call it, it's the device inside the big black or white box which holds all the information. If you have a 5Mb hard drive the original unit is a Tandon TM-602S, which has 153 cylinders and 4 heads. If you have a 15Mb hard drive it will come with a Tandon TM-503, which has 306 cylinders and 6 heads. When new, both hard drives are very quiet and the head stepping noise is barely

perceptible. As they wear the bearings get noisier and the head stepping makes a distinctly ping-ping sound. Use this to gauge the amount of use the drive has had. Very noisy drives tend to have had a long hard life, and as such may be about to develop bad sectors. Only time and use can confirm that. Like all hard drives, once these old MFM drives develop bad sectors it life and usefulness are pretty much over, so in a future article I'll show how to substitute the Tandon MFM with something else.

CHECKS FOR BOTH THE BLACK 26-1130 AND WHITE 26-4155W DRIVES

Once you have the hard drive in front of you there is a number of things to do which can be checked now and will save some grief down the road. After you remove the top of the hard drive cabinet (three screws at the back) you will see a controller board with cables running to it. The 26-1132 will have a large HDC board with a very large IC in the middle marked 8x300 (see photo). I'm going to call this the 8x300 board from now on and in all future articles. If you have a 26-4155W drive you will have either an 8x300 board (early units marked "Radio Shack") or a new revised small HDC which has a WD1010 IC on it (later units marked "Tandy"). This board I will refer to as the WD1010 board from now on. Under this chassis which holds the HDC is the hard drive itself on the left, and the power supply on the right. The chassis is held down by four screws and once removed the chassis lifts up easily after you have unplugged all the cables - its a good idea to mark where they all went.

Looking at the hard drive it may or may not have excessive shock sensors (see photo), if they have turned red then there is good evidence that they drive may have been handled roughly enough for the heads to be damaged or the platters to be marked by the heads crashing into it, so prepare for the worst - the drive itself may need to be replaced. Looking inside the chassis you will see the power supply. Those familiar with the insides of a Model 4 will instantly recognise it's the same power supply. The same Rifa filter capacitors which give problems in the Model 4 also give problems in the hard drive, so now is a good time to replace them. For hard drives which contain the Astec version of the power supply I replace Caps C1, C2 and C13 and re-solder TB1 and TB2 with a nice hot soldering iron. On the Tandy version of the power supply (120V) I replace C32 and C33, and re-solder J2 and J3.

Using a multimeter or even a simple continuity tester, confirm both the Activity light and Protect light works. The Protect light in particular can be a real nuisance which is fails because it completes the circuit for the hardware write-protect. When the globe blows the hard drive thinks the disk is write-protected. To fix this I put a 1/2W 470ohm resistor across the light leads so that even if the light blows the hardware write-protect will still function normally. While on the subject of blown globes, if they fail and you can't find replacements (they are hard to find and expensive) they can be replaced with ordinary LEDs but because the current to drive globes is much higher than LEDs, the LED needs a 1/2W 100ohm resistor placed in series with it.

FURTHER TESTS FOR THE 8x300 HDC BOARD[6]

While the unit is open if you have access to a multimeter you can check these two points: U32 Pin 14, check for a voltage of 2.4V to 3.0V and then check TP8 for a voltage of 2.5V +/- 0.5 volts. Hang a frequency counter off TP9 and check for a 10.00MHz output. If these three conditions aren't met then the HDC will never work - they are required to modify the frequency modulation within a very specific window that is the essence of reading and writing to MFM hard drives. To correct the voltage at TP8, adjust R18. To correct the frequency at TP9, adjust C33. If you aren't getting the correct voltage at U32 Pin 14, replace U32, R22 or R24 or other components as needed.

THOSE OF YOU WITH THE WD1010 HDC BOARD

Life is much simpler for those of you who have found a hard drive with the smaller WD1010 HDC board, I have never seen one which doesn't work - they are very robust, and using digital timing for locking in the frequency and voltages, there are no adjustments to float off spec.

BEFORE YOU CONNECT IT TO THE COMPUTER

So you've got it all back together and the moment of truth has arrived, time to power the old unit up. Plug in the power and turn the key. You should immediately hear the hard drive spin up, and feel a breeze from the cooling fan. On 8x300 HDCs the Activity light will be out for 15-20 seconds and then turns green. On the WD1010 HDCs the Activity light shines green for the first 15-20 seconds and then goes out when the drive has finished its self-test.

CONNECT IT TO THE COMPUTER WILL YA!

All that is left now is to connect it to the TRS-80 Model III or 4 and set the unit it. Connecting the computer using a standard cable will have the card-edge (computer) end exiting from the rear of the TRS-80 and the female end exiting down from the hard drive ("Computer In" connection). On a TRS-80 system the hard drive is always powered up first, then the computer, so turn on the hard drive and a few seconds later power up your TRS-80. It should boot your favourite operating system from floppy disk as it normally would. This is another handy test that you have the hard drive cables plugged in properly because if you get them wrong the computer won't boot. Check your cables and try again.

The software side of this will fill a whole article so I'll leave it for next time, but for the moment I won't leave you in the lurch. If you have made it this far and wish to test your hard drive, you need the appropriate Tandy disk set to configure and format the hard drive, for the 5Mb hard drive you need set 26-1130 which comprises of two disks: Hard Disk Operating System Initialization and Hard Disk

Operating System. For the 15Mb hard drive you need disk set 26-1133, which is the same pair of disks but has patched drivers to format extra tracks and heads on the larger hard drive. If you need either of these disk sets to get started, send me an email.

NEXT TIME I will discuss setting up a hard drive with Misosys RSHARD5/6 drivers and how to set up a dual boot hard drive for Model III mode hard drive operation on a Model 4.

ianm@trs-80.com

REFERENCES:

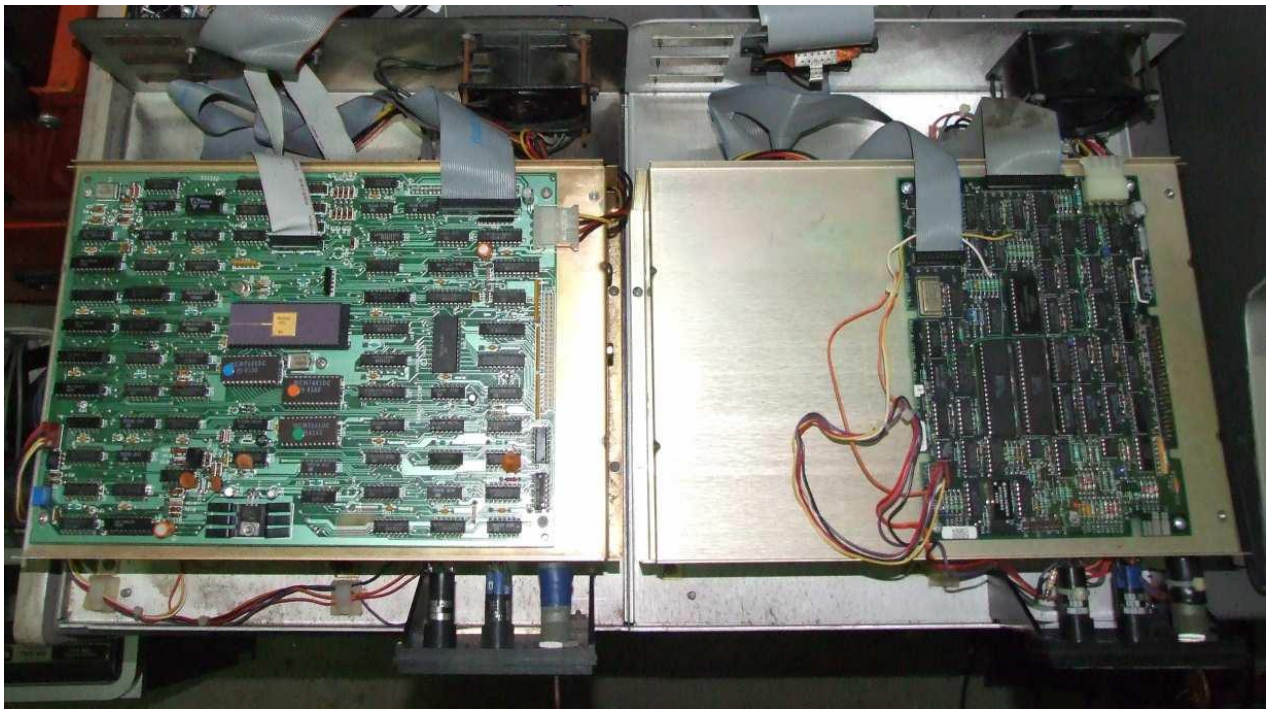
1. "Building The Super Micro - Part 4", TRS8Bit, September 2012 pp. 14-16.
2. "TRS-80 Catalog" RSC-8, Sept 1982, pp. 48.
3. "TRS-80 Catalog" RSC-10, Sept 1984 pp. 19.
4. http://www.jaycar.com.au/productView.asp?ID=SM1031
5. http://tim-mann.org/trs80/doc/rshard.pdf
6. "Service Manual for TRS-80 5-Meg Hard Disk", Radio Shack, 1982, pp.34

DISCUSSION QUESTIONS:

1. Why do you think Tandy chose a keyswitch for powering the unit up and down?
2. Can you think of a situation where a hardware write-protect would be useful?
3. What kind of large data volume applications would be well suited to a TRS-80 hard drive?
4. Why did Tandy choose LDOS has its Hard Disk Operating System?
5. As the Model 4 matured Tandy released more hard-disk based applications and lowered the price of the computer and hard drive - Why was this necessary?
6. Although there were no Model I hard disk specific applications, Tandy felt compelled to market an adaptor kit to attach the 5Mb drive to the Model I - Why would they have done this?
7. Given the cost, how many TRS-80 hard drives do you think were sold?



Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.




L D O S - The Logical Disk Operating System - Ver 5.3.1
Copyright 1991 MISOSYS, Inc., All rights reserved
LDOS
Sun, Nov 11, 2012
Time ?
LDOS Ready
Radio Shack
TRS-80
Microsoft Project 2000



TRS-80 MODEL IV, III/1 HARD DISK SYSTEM



Hard Disk Drive Adds 5 Megabytes of Storage to Your Model IV, III & I

3995.00
Primary Hard Disk System

3595.00
Secondary Hard Disk System

- Add Up to 3 Secondary Hard Disk Drives For a Total of 20 Megabytes of Storage
- Compatible With Most Model III/I Software
- Large Capacity and High Speed Make It Ideal For Inventory, Accounting, Data Base Filing
- Operating System Enhances Familiar TRSDOS Commands With Convenient Hard Disk Features
- BACKUP's Special Features Allow Easy Selection Of Specific Files and Fast Backup of Large Files to Multiple Diskettes

I/III [A] Our 5 megabyte Hard Disk System gives you vastly increased data storage—with exceptionally fast access and ease of use. Winchester technology can provide dramatic improvements in speed. The advanced operating system's many superior, user-friendly features make file handling a breeze. Store over 1000 different files per disk system or create files as large as the available space. The directory is alphabetized to make locating and accessing files quick. High-level communication is available through an advanced communication package that allows direct downloading to disk. A print spooler is included, too. The Job Control Language allows you to compile and later execute a predefined series of commands, including screen and audio alerts, conditionals, and user interaction to make many large-scale operations virtually automatic. BACKUP, in addition to ordinary backup functions, permits backing up only those files that have been modified, or modified in certain date ranges, or even those that do or do not already exist on the destination disk. When backing up from hard disk to diskettes, the system will span a large file across multiple diskettes and allow easy identification of diskettes when recreating a file on the hard disk. Other features include keyboard type-ahead for fast data entry, a versatile printer output formatting program, key redefinition, and more. Extensive user's manual included.

Primary Drive. Includes Hard Disk Operating System with BASIC. 26-1130 ..... 3995.00

Secondary Drive. Installation required (not included). 26-1131 ..... 3595.00

Model I Adapter Kit. Includes Adapter, cable and diskettes. Note: May require installation (not included). 26-1132. ..... 69.95

PIGEON POST

THIS IS THE RECENTLY ANNOUNCED STORY OF A CARRIER PIGEON DISPATCHED BY THE D-DAY INVASION FORCE TO RELAY SECRET MESSAGES BACK ACROSS THE CHANNEL, BUT WHO NEVER MADE IT HOME TO ITS BASE.

WITH ALL OUR RECENT ARTICLES ON CODES ETC. I THOUGHT THIS WOULD BE OF INTEREST. HAS ANYONE HAD A GO AT CRACKING THE MESSAGE ON THEIR MODEL 1 (AND BEATING GCHQ)? HOWEVER, THE POOR BIRD GOT STUCK IN A CHIMNEY ONLY TO BE DISCOVERED 70 YEARS LATER, IT'S SECRET COMMUNIQUé STILL ATTACHED TO ITS SKELETON IN A RED CAPSULE.

THE MESSAGE IS SO SECRET THAT IT IS WRITTEN IN CODE LONG SINCE FORGOTTEN BY THE SECURITY SERVICES.

NOW THE GOVERNMENT COMMUNICATIONS HEADQUARTER (GCHQ) IN CHELTENHAM, GLOS, IS HOPING TO DECIPHER THE NOTE IN A BID TO UNRAVEL THE MYSTERY.

THE ROYAL PIGEON RACING ASSOCIATION BELIEVE THE BIRD PROBABLY EITHER GOT LOST, DISORIENTATED IN BAD WEATHER, OR WAS SIMPLY EXHAUSTED AFTER ITS TRIP ACROSS THE CHANNEL.

ITS REMAINS WERE DISCOVERED BY DAVID MARTIN WHEN HE RIPPED OUT A FIREPLACE AND FOUND IT IN THE CHIMNEY WHILE RENOVATING HIS HOME IN BLETCHINGLEY, SURREY.

HISTORIANS BELIEVE THE BIRD WAS ALMOST CERTAINLY DISPATCHED FROM NAZI-OCCUPIED FRANCE ON JUNE 6, 1944 - DURING THE D-DAY INVASIONS.

DUE TO WINSTON CHURCHILL'S RADIO BLACKOUT, HOMING PIGEONS WERE TAKEN ON THE D-DAY INVASION AND RELEASED BY ALLIED FORCES TO INFORM MILITARY GENERALS BACK ON ENGLISH SOIL HOW THE OPERATION WAS GOING.

"IT'S A REAL MYSTERY AND I CANNOT WAIT FOR THE SECRET MESSAGE TO BE DECODED," MR MARTIN SAID.

"IT'S REALLY UNBELIEVABLE."

IT IS THOUGHT THAT THE BIRD WAS DESTINED FOR THE TOP SECRET BLETCHLEY PARK, WHICH WAS JUST 80 MILES FROM MR MARTIN'S HOME.

THE MESSAGE WAS SENT TO X02 AT 16:45 AND CONTAINED 27 CODES, EACH MADE UP OF FIVE LETTERS OR NUMBERS.

THE DESTINATION X02 WAS BELIEVED TO BE BOMBER COMMAND, WHILE THE SENDER'S SIGNATURE AT THE BOTTOM OF THE MESSAGE READ SERJEANT W STOT.

EXPERTS SAID THE SPELLING OF SERJEANT WAS SIGNIFICANT, BECAUSE THE RAF USED J, WHILE THE ARMY USED G.

DURING THE WAR, CODE BREAKERS WORKED AT BLETCHLEY PARK ROUND THE CLOCK IN TOP SECRET - DECIPHERING NAZI CODES INCLUDING ENIGMA.

IT WAS ALSO HOME TO A

CLASSIFIED MI6 PIGEON LOFT, MANNED BY TRAINER CHARLES SKEVINGTON.

COLIN HILL, A VOLUNTEER FOR THE ROYAL PIGEON RACING ASSOCIATION SAID: "THE MESSAGE MR MARTIN FOUND MUST BE HIGHLY TOP SECRET.

"THE ALUMINIUM RING FOUND ON THE BIRD'S LEG TELLS US IT WAS BORN IN 1940 AND WE KNOW IT'S AN ALLIED FORCES PIGEON BECAUSE OF THE RED CAPSULE IT WAS CARRYING - BUT THAT'S ALL WE KNOW.

"WE SUSPECT IT WAS FLYING BACK TO MONTY'S HQ OR BLETCHLEY PARK FROM NAZI OCCUPIED NORMANDY DURING THE INVASION.

"I CAN ONLY PRESUME IT BECAME EXHAUSTED AND ATTEMPTED TO REST ON AN OPEN CHIMNEY - WHERE IT VALIANTLY PERISHED.

"THIS IS A VERY SPECIAL MESSAGE BECAUSE MOST WERE NOT WRITTEN IN CODE. THIS IS WRITTEN LIKE AN ENIGMA MESSAGE SO MUST HAVE BEEN VERY SECRET AND VERY IMPORTANT."

PIGEON ENTHUSIASTS - COMMONLY KNOWN AS "FANCIERS" - WERE CALLING FOR MR MARTIN'S MYSTERIOUS MILITARY BIRD TO BE POSTHUMOUSLY DECORATED WITH THE DICKIN MEDAL - THE HIGHEST POSSIBLE DECORATION FOR VALOUR GIVEN TO ANIMALS.

THE CRACK TEAM OF BIRDS WERE A SECRET WING OF THE NATIONAL PIGEON SERVICE - WHICH HAD A SQUADRON OF 250,000 BIRDS DURING THE SECOND WORLD WAR.

THEY CAN REACH SPEEDS OF 80MPH, COVER DISTANCES OF MORE THAN 1,000 MILES AND ARE THOUGHT TO USE THE EARTH'S MAGNETIC FIELDS TO NAVIGATE.









The *BEST* in TRS-80s

Call The Right Stuff

Ask for Ian
The number is +61 416 184 893

That's The Right Stuff
And he's in Melbourne





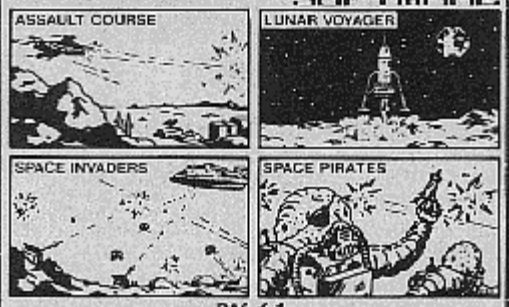


http://ianmav.customer.netspace.net.au/trs80/

MICROMART

VIDEO GENIE & TRS 80 16K LEVEL II

DACC SOFTWARE

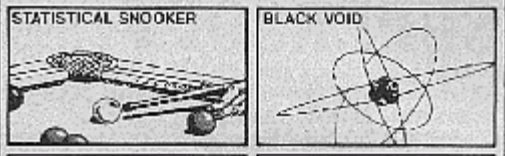


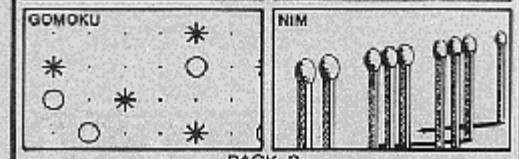
PACK 1



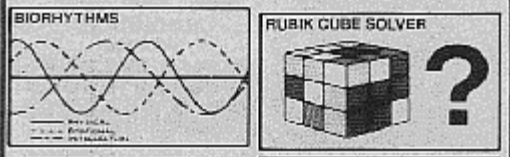


PACK 2





PACK 3





PACK 4

EACH PACK	£9·50	ANY TWO PACKS	£14·50
ANY THREE PACKS	£18·00	ALL FOUR PACKS	£19·50

These four packs (16 games in all) offer the TRS 80 and Video Genie user an enormous variety of excellent games at greatly reduced prices. There are fast, smooth Z80 machine code arcade games like Assault Course, Space Pirates and Space Invaders. There are brain teasers like 3D O's and X's, NIM, Gomoku and a Code Breaker that can even play against you move for move! There are new games as well such as Statistical Snooker, Dracula's Castle and 40-over Cricket.

Promptly delivered and sensibly priced, each tape has a 12 months' guarantee. You can't miss this fantastic offer.

WRITE OR PHONE YOUR ORDERS TODAY!

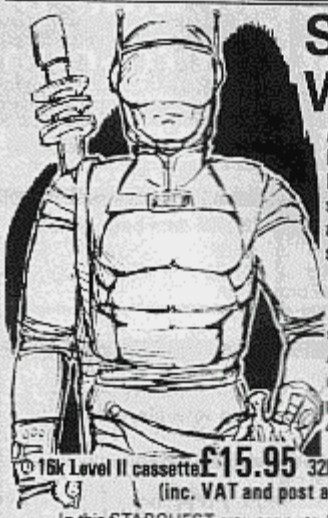
ACCESS ORDERS TAKEN BY PHONE 24 HRS A DAY

Dealer enquiries welcome

24hrs

Lower North St., Cheddar, Somerset.
Tel. Cheddar 515265

TRS-80 & VIDEO GENIE SOFTWARE



STAR WARRIOR

ARMED WITH NUCLEAR MISSILES, BLASTER & POWER GUN, You are on a search and destroy mission against a military dictatorship.

● GRAPHICS! ● Real time ● Two different scenario's ● For ages 10 to Adult ● Complexity: Intermediate ● For one player ● Playing time 10 to 40 minutes ●

16k Level II cassette £15.95 32k TRS DOS disk £17.95
(inc. VAT and post and packing)

In this STARQUEST, you are a starfaring knight bound tightly by your code, a mercenary with a cause. You are a Fury, an interplanetary avenger on a mission of justice and danger, called on by the people of the planet Fornax to free them from the rule of an oppressive military governor. You must take on an entire planetary occupation force—alone! Outnumbered by storm troopers of the Stellar Union, you'll always be on the lookout for hover craft, robot tanks, arillery—even the deadly, rapid firing "Mauler"—nine types of enemy vehicles in all. But you're far from helpless.

When you play "Star Warrior", you'll walk, jump—even fly—over swamps, forests and mountains. Six different sounds and even better graphics than our DUNJONQUEST series make "Star Warrior" more fun, and, the emphasis is on fast action and challenge.

Choose from 19 command options, several suits of powered armour, 5 levels of skill and 2 scenarios. In Scenario One, you must divert your enemy from the main attack. Your aim is to destroy as many military installations and vehicles as you can before you are recalled. You can even set your own time limit, so you can play for as little or as long as you like! In Scenario Two, you must track down and destroy the tyrannical military governor and his staff. You'll use sophisticated electronic direction finding equipment to home in on your targets. You have decoys to fool your enemy, and you are equipped with nuclear missiles, blaster and power gun. The governor is always on the move, so you never know where you'll find him next!

But like your mythical namesake, you are a Fury—a soldier of daring and courage, and the people of Fornax are depending on you!

Rescue at Rigel (R.T.G.) £10.95
As Sudden Smith with Force Shield and Blaster try to rescue prisoners trapped at Rigil. Too many rooms and levels to be bored.

Morloc's Tower (R.T.G.) £10.95
Enter Morloc's Tower and slay said fiend. To do so you need to collect various nic-nacs on the way. Who is Hagedorn?

Datestones of Ryn (R.T.G.) £10.95
Recover Datestones from Rex (the Reaver) and his friends. He is not at all happy that you shall recover them so he has set some traps and sprinkled a few monsters to entertain you in your quest.

Temple of Apshai (R.T.G.) £14.95
200 rooms and 30 monsters – 16 million alter-egos possible! Graphic representation of dungeon room you are in and of monster gobbling you.!

Invasion Orion £13.95
Tactical Space battle - you may set up entirely new scenarios using from 2 to 15 spacecraft or use data provided (on cassette) to play pre-set scenarios.

R.T.G. – Real Time Graphics.

All prices include p & p and V.A.T.

ALGRAY

ALGRAY House, 33 Bradbury Street, Barnsley. South Yorkshire Tel: Barnsley (0226) 83199

(CONTINUED FROM PAGE 1)
STYLE CURRENTLY IN USE.
THIS WILL ALLOW MORE FONT CHARACTERS, SUCH AS THE HASH AND SUPER/SUB SCRIPTS.
I SHALL STILL KEEP THE 'NORMAL' TYPEFACE FOR MY USUAL 'RAMBLINGS' AND ITEMS GENERAL OF INTEREST AS I THINK IT'S SO INDICATIVE OF THE TANDY ERA! (I JUST WISH I'D STILL HAD MY OLD TELETYPE TO PRINT ON).

NOW,,, THERE IS GREAT EXCITEMENT AT THE MILLER HOUSEHOLD AT THE MOMENT.
I'M LEAD TO BELIEVE THAT SANTA MAY VISIT AND LEAVE A TWIN 5 1/4 INCH DISK DRIVE TOGETHER WITH A 'MACINKER' (THESE WERE USED TO RE-INK AND REJUVENATE OLD DOT MATRIX PRINTER RIBBONS!)
THE DISK DRIVES, I'M TOLD, WERE FROM A BBC MODEL 'B' COMPUTER SO I'M NOT SURE HOW, OR EVEN IF, THEY ARE COMPATIBLE WITH MY MODEL 1.
COME THE NEW YEAR I'M SURE I'LL BE EMAILING ALL AND SUNDRY PLEADING FOR HELP AND ASSISTANCE!
AS FOR THE 'MACINKER', I'M TOLD SANTA WILL LEAVE ANY NECESSARY INSTRUCTIONS THAT ARE REQUIRED. WHEN I ENQUIRED ABOUT INKS ETC., I WAS FIRMLY TOLD "NOT TO PUSH MY LUCK"
I'LL KEEP YOU ALL INFORMED OF MY PROGRESS!

AND SO DOWN TO THIS ISSUE.
MAV HAS COME UP TRUMPS AGAIN WITH AN ARTICLE ON TANDY, MODEL 3 & 4, HARD DISKS.

'AT THE READY PROMPT' SHOWS HOW TO REDUCE THE TIME TAKEN IN WRITING CASSETTE DATA FILES, FOR BOTH OLD AND NEW TYPE MODEL 1 ROM'S.
THIS REALLY WAS LIKE FATE PUSHING THE IDEA ALONG AS I CAME ACROSS THE 'FIX' ARTICLE BEFORE I BECAME AWARE OF THE PROBLEM, AND ALSO, I JUST HAPPENED TO HAVE A COPY OF 'PRACTICAL COMPUTING' FROM OCTOBER 1981 AMONG MY BITS AND PIECES!

THE USUAL XMAS CARD IS HERE FOR YOU TO PRINT, USING THE BASIC PROGRAM SUPPLIED, IF YOU WISH. THIS IS THE 6TH SUCCESSIVE YEAR I'VE PRODUCED THIS 'BIT OF FUN' SO I HOPE IT MAKES YOU SMILE.

I FIRST HEARD OF THE 'PIGEON POST' ARTICLE ON A BBC WORLD SERVICE BROADCAST. I WAS QUITE INTRIGUED TO HEAR THAT GCHQ WERE ATTEMPTING TO CRACK THE CODED MESSAGE AND THOUGHT, WOULDN'T IT BE GREAT IF SOME POOR SOUL ON THEIR MODEL 1 GOT THERE FIRST!

I HOPE THE MATTHEW SMITH ARTICLE GRABS YOUR ATTENTION. JUST FANCY THAT, ALL THOSE SPECTRUM PROGRAMS BEING WRITTEN ON A TANDY, FIRST, FOR EASE OF DEVELOPMENT!

WELL, THAT'S ABOUT IT FOR THIS ISSUE AND THIS YEAR.
THE NEXT EDITION WILL BE OUT EARLY MARCH 2013, SO IT'S NEVER TOO EARLY TO SEND IN YOUR CONTRIBUTION!
HAVE A GREAT XMAS EVERYONE.



DUSTY

CONSULTANT ELECTRONICS
VIDEO GENIE COMPUTERS
Books — C10 Cassettes — Disks
Your own 'Basic' Genie / TRS80 Programs listed.
Vic 20, ZX81, Genie, TRS80 Software.
59 Mason Road, Erdington, Birmingham. Tel 021-382 7247

THE Video Genie EG3000 Series

WITH *16K user RAM plus extended 12K Microsoft BASIC in ROM *Fully TRS-80 Level II software compatible *Huge range of software already available *Self contained, PSU, UHF modulator, and cassette *Simply plugs into video monitor or UHF TV *Full expansion to disks, and printer *Absolutely complete just fit into mains plug



16K **£299** + VAT

GP80

The most compact 80 column impact graphic-dot printer available - at a very compact price

by seikosha

Graphics, Normal and Double Width Characters can be printed on the same line. Pine Feed Tractor is equipped as standard.
Two Line Feed Commands (1/6 and 1/9 inch)
Print on Plain Paper with two copies
Continuous Self-Inking Ribbon for easy handling
Centronics type Parallel Interface standard
Wide Range of Optional Interface Boards
Self-Test Programme standard



£225 + VAT

Acorn Atom



£150

Unique in concept — the home computer that grows as you do!

Special features include
***FULL SIZED KEYBOARD**
***ASSEMBLER AND BASIC**
***TOP QUALITY MOULDED CASE**
***HIGH RESOLUTION COLOUR GRAPHICS**

Simple to build, simple to operate. A powerful, full facility computer with all the features you would expect.
Just connect the assembled computer to any domestic TV and power source and you are ready to begin.
Full-sized QWERTY keyboard 6502 Microprocessor Rugged injection-moulded case 2K RAM 8K HYPER-ROM 23 integrated circuits and sockets Audio cassette interface UHF TV output Full assembly instructions

£120 + VAT



The Second Generation Personal Computer

Highest performance - lowest price

*48K (8080A)
*16 Colours or shades of Grey
*Multiple High Resolution Graphics Modes (64 x 71, 129 x 159, 255 x 335)
*Character mode (80 x 24)
*Split Screen Modes
*Full ASCII Upper & Lower Character Set
*Unique graphical — Sound Commands for Smooth Music, random frequencies & enveloped sound!
*RS232 I/F

DAI Personal Computer at only + VAT **£595**

SUPERBRAIN

SuperBrain's CP/M operating system boasts an overwhelming amount of available software in BASIC, FORTRAN, COBOL, and APL. Whatever your application . . . General Ledger, Accounts Receivable, Payroll, Inventory or Word Processing. SuperBrain is tops in its class.

320K £1850 700K £2400
1.5Mb £2750

COMMODORE COMPUTERS

PET 8K	£415
PET 16K	£525
PET 32K	£650
PET 8032	£895

DISK DRIVES

4040	£895

PRINTERS

EPSON TX80B (inc. I/F & cable)	£299
EPSON MX80T	£395
ANADEX DP8000	£495
ANADEX DP9500	£895
ANADEX DP9501	£995

VIDEO MONITORS

10" BLACK & WHITE	£85
10" GREEN SCREEN	£95

Aculab

Floppy Tape
The tape that behaves like a disk.

Connects directly to TRS-80 level 2 keyboard. Operating and file handling software in ROM 8 commands add 12 powerful functions to level 2 BASIC. No buttons, switches or volume controls. Full control of all functions from keyboard or program. Maintains directory with up to 32 files on each tape.

for all TRS 80 & Video Genie owners

TRS80 version: **£165**
Video Genie : **£170**

Please add £10 Securicor delivery charge to all computers etc.
Plus 15% VAT on all prices.

Paper Tiger 460

The Paper Tiger 460 sets new standards by incorporating many features previously only available on units costing much more.

Features like a specially developed nine wire 'staggered column' head which overlaps the dots of each matrix character with just one pass of the printhead giving a dense, high quality print image without reducing the units 160 c.p.s. print speed.

It also offers a bi-directional logic seeking device to enhance its print optimisation characteristics and wide range of 'print versatility' features such as mono or proportional spacing, automatic justification, programmable horizontal and vertical tabbing, and 'fine' positioning for word processing applications.

£795 + VAT

Paper Tiger 560

The Paper Tiger 560 is the first printer which bridges the gap between conventional matrix and 'daisy wheel' types offering quality printing at a relatively low price.

Full 'width' 132 column printing at 160 c.p.s., a unique nine wire 'staggered' print head, bi-directional printing, an inbuilt tractor feed and a host of selectable features set it apart from ordinary matrix printers.

Plus for even greater versatility a full dot plot graphics facility, if supplied which includes a 2K buffer.

£995 + VAT

Books & bits

Books — Manuals — Diskettes — ribbons — Paper — chips (2114 x 2 1K) £4 pair. RS232 to Centronics interfaces £40 etc. etc.
A variety of second-hand computer equipment usually available, spares, repairs and service.

MicroStyle

9 St. Peter's Terrace, Lower Bristol Road, Bath, BA2 3BT
Telephone: (0225) 334659. After Hours: (0761) 33283

New Showroom
OPENS 1st JUNE AT 29 BELVEDERE, BATH.
2 minutes from town centre

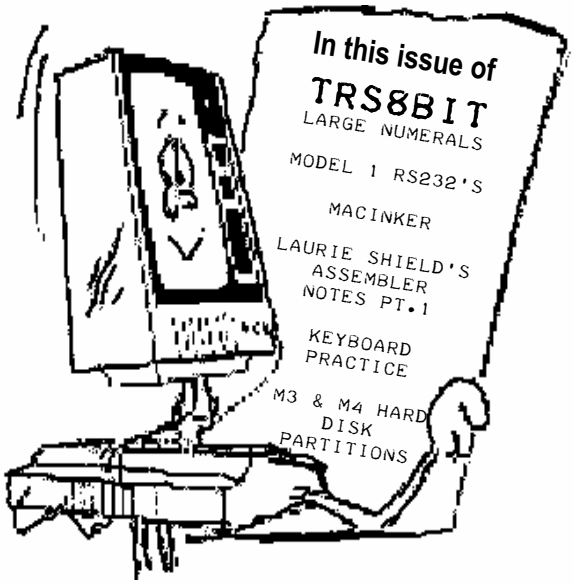
TRS8BIT



HI EVERYONE AND WELCOME TO THIS 'BUMPER' 32 PAGE EDITION, WHICH STARTS OFF OUR 7TH YEAR!

I DON'T LIKE STARTING OFF WITH SAD NEWS, BUT I'M SORRY TO REPORT THE UNTIMELY DEATH OF LANCE WOLSTRUP. HE'S KNOWN TO MANY TANDY FANS IN THE UK AS THE EDITOR IN CHIEF OF THE TRSTIMES MAGAZINE, WHICH WAS PUBLISHED IN THE STATES FROM THE LATE 1980'S TO 1995. HE WAS A BIG SUPPORTER OF ALL THINGS 'TANDY' AND WILL BE SADLY MISSED. I FEEL THERE'S A 'BIG HOLE' LEFT IN THE TANDY SCENE, WHICH WILL BE VERY DIFFICULT, IF NOT IMPOSSIBLE TO FILL.

THERE'S QUITE A BIT TO REPORT ON THE NEWS FRONT. ALTHOUGH EBAY, UK, HAS BEEN QUIET, APART FROM THE SAME-OLD ADVERTS FOR CASSETTE SOFTWARE AT £90 A TIME ETC ETC, I NOTICED A MODEL 100 SELLING FOR JUST UNDER £50.



MATTHEW REED HAS ISSUED VARIOUS UPGRADES TO HIS TRSREAD/WRITE SOFTWARE AND TRSTOOLS.

MIGUEL DUTRA HAS RELEASED V1.6 OF HIS VDK-80 SOFTWARE.

THE MODEL100 CLUB IS RUNNING A PROGRAMMING COMPETITION. ENTRIES MUST BE IN BY 15MAR13 SO BY THE TIME YOU READ THIS, YOU'LL BE NEEDING TO GET YOUR 'SKATES' ON!

NEIL PINNER HAS A TANDY LPVIII PRINTER FOR SALE. EMAIL ME IF YOU WANT FURTHER DETAILS.

'DROID-TRS' SHOULD BE AVAILABLE ON BETA TESTING SHORTLY.

TERRY STEWART HAS A GREAT ARTICLE ON HIS WEBSITE RE SEMI-VIRTUAL DISKETTES (SVD) WHICH 'SIT' BETWEEN A PC AND A SYSTEM-80.
FOR FURTHER DETAILS ON ANY OF THE ABOVE ITEMS, PLEASE CHECK OUT THE INDIVIDUAL WEBSITES. THERE ARE LINKS READY TO USE ON TRS-80.ORG.UK

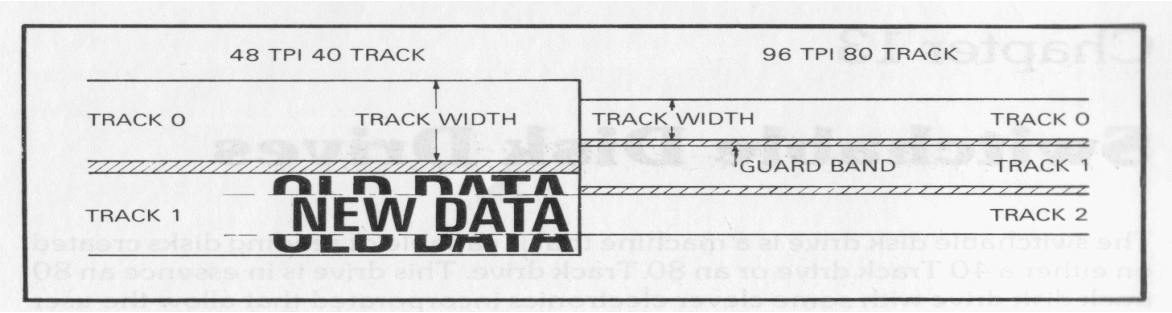
FOR THOSE OF YOU WHO WILL RECALL, IN THE LAST ISSUE, THE EXCITEMENT, ON THE RUN-UP TO XMAS, WHEN I THOUGH I MIGHT BE RUNNING 'REAL' DISKS ON MY M1. WELL, I'M SORRY TO REPORT THAT SANTA GOT IT WRONG, JUST A BIT! THE TWIN DISK DRIVE I RECEIVED WAS FROM A SWITCHABLE 40/80 TRACK, EX BBC BIT OF KIT. MAV DID WARN ME ABOUT THAT POSSIBILITY. I MANAGED TO FIND A BBC DISK

(CONTINUED ON PAGE 2)

DRIVE GUIDE ON EBAY AND THE DIAGRAM BELOW SHOWS THE PROBLEM.
IT SEEMS IT CAN READ AND WRITE A 40 OR 80 TRACK DISK WHICH IT HAS WRITTEN ITSELF, QUITE HAPPILY, BUT IT'S NOT ABLE TO READ 'OLD STYLE' 40 TRACK DISKS.
NOT TO WORRY TOO MUCH THOUGH, I'VE STILL GOT MY VIRTUAL DISKS ON MATTHEW REED'S EMULATOR, IT'S JUST THAT I WOULD HAVE LIKED TO TRY AND READ MY OLD DISKS FROM THE 1980'S. IT'S NO GOOD, I SHALL HAVE TO SEND THEM OFF TO IRA FOR CONVERSION!

THE SECOND IS ON HARD DISK PARTITIONING ON THE M3 & M4'S. THE LATTER HAS STEP BY STEP INSTRUCTION TO HELP AS YOU GO ALONG. IT'S A VERY IMPRESSIVE ARTICLE, WITH A LOT OF TECHNICAL DETAIL AND QUITE MIND BLOWING TO ME!

THEN, THERE'S PART ONE OF LAURIE SHIELDS 'ASSEMBLER NOTES' FROM THE 1980'S. IF YOU WANT TO START OR WISH TO BRUSH-UP YOUR ASSEMBLER PROGRAMMING SKILLS, THIS IS WHERE TO BEGIN. I REMEMBER ATTENDING A COUPLE OF TALKS GIVEN BY LAURIE IN THE 1980'S, HE HAD THE PATIENCE OF □OB AND THAT KNACK, LIKE



AND WHILE I'M TELLING YOU ALL MY WOES, ANOTHER SLIGHT DISAPPOINTMENT WAS WITH THE MACINKER! IT SEEMS THAT THE MAIN RIBBON I WANTED TO RE-INK, FOR MY TANDY LPVII, NEEDS A SPECIFIC RE-INKER AND NOT THE UNIVERSAL ONE. ANYWAY, DEE HAS PROMISED TO TRY AND REPLACE THE RIBBON IN THE CARTRIDGE AS SOON AS HER HAND IS BETTER. WHAT WOULD WE DO WITHOUT THESE WOMEN EH? □UST IN CASE YOU HAVEN'T COME ACROSS THE MACINKER, I'VE REPRINTED, ON PAGE 9 AND 10, ONE OF THEIR BROCHURES FROM THE 1980'S.

AND SO TO DETAILS OF THIS 'BUMPER' EDITION.
IAN MAVRIC HAS SENT A COUPLE OF ARTICLES. THE FIRST ON M1 RS232 CARDS AND MOST GOOD TEACHERS, OF INSPIRING ENTHUSIASM IN EVERYONE AROUND HIM.

THERE'S ALSO A COUPLE OF 'BASIC' ARTICLE ON PRINTING LARGE NUMERALS AND ALSO ONE TO HELP IMPROVE YOUR KEYBOARD SKILLS.

WELL, THAT ABOUT WRAPS UP THIS, BIGGEST YET, ISSUE. I HOPE YOU FIND IT INFORMATIVE AND ENJOYABLE. □UST TO WET YOUR APPETITE FOR THE □UNE 2013 EDITION, MAV HAS PROMISED AN ARTICLE ON THE MODEL 2000. THAT WILL BE A FIRST FOR TRS8BIT. I, FOR ONE, CAN'T WAIT.

TAKE CARE

DUSTY



Large numeral generator

from a program
by E. G. Kemplen

This routine displays a large numeral as a 6 by 7 matrix of asterisks. Written for PET, it should run on other micros quite easily.

The routine sets up two arrays, A$ and A in lines 110 to 190. A$ contains the four elements used to construct the numeral, while A contains ten sets of codes used to select the appropriate element for each row.

This subroutine will work as it is because line 200 contains an instruction to input the value, X, to be displayed.

```
100 PRINT"■"
110 DIMA(70),A$(4),B$(3),S(14)
120 DATA"******"
130 DATA"*     "
140 DATA"     *"
150 DATA"*    *"
160 FORI=1TO4:READA$(I):NEXT
170 DATA1,4,4,4,4,4,1,3,3,3,3,3,3,3,1,3,3,1,2,2,1,1,3,3,1,3,3,1,4,4,4,1,3,3,3
180 DATA1,2,2,1,3,3,1,1,2,2,1,4,4,1,1,3,3,3,3,3,3,1,4,4,1,4,4,1,1,4,4,1,3,3,1
190 FORI=1TO70:READA(I):NEXT
200 INPUT "■";X:PRINT"■"
240 FORI=1TO7
250 PRINT"■■■■■■■";A$(A(X*7+I))
260 NEXT
270 GOTO 200
280 REM
```

WHILE I WAS DELVING INTO AN OLD EDITION OF PERSONAL COMPUTER WORLD, DURING ALL THE CHRISTMAS FESTIVITIES, THIS LITTLE PROGRAM CAUGHT MY EYE. I THOUGH IT MIGHT BE OF INTEREST FOR ME WHEN I'M PLAYING ABOUT WITH LABEL PRINTING. MIND YOU, I'M ALWAYS VERY WARY OF THE USUAL CAVEAT STATING SOMETHING LIKE, "SHOULD CONVERT QUITE EASILY FOR USE ON OTHER MACHINES".
AS THE PROGRAM WAS ORIGINALLY WRITTEN FOR THE COMMODORE PET, I WAS QUITE HAPPY TO 'HAVE A PLAY' AND SEE HOW I MANAGED TO GET ON. I'VE A LITTLE PROGRAM CONVERSION BOOK, WHICH I'VE HAD FOR YEARS, AND I THOUGHT I COULD REFER TO IT, IF AND WHEN NEEDED. FOR INSTANCE, LINE 1ØØ IS THE PET EQUIVALENT OF CLS. LOW AND BEHOLD, IT DIDN'T TAKE TOO LONG AND TURNED OUT TO BE QUITE A NICE LITTLE PROGRAM. THE ONLY COSMETIC CHANGE I THOUGHT MIGHT BE REQUIRED, WAS TO REDUCE THE PRINT MATRIX FROM 6X7 TO 5X7. THIS SEEMED TO GIVE A MUCH MORE BALANCED DISPLAY ON MY MODEL 1.

THEN THE USUAL THOUGHT CROSSED MY MIND! I WONDER IF I WOULD BE ABLE TO MAKE IT INTO A ONE-LINER. THERE CERTAINLY WOULDN'T BE ENOUGH ROOM FOR ALL THOSE DATA STATEMENTS. THOSE PESKY COMMAS TAKE UP FAR TOO MUCH ROOM!. SO IT WAS GOING TO NEED A COMPLETE RE-THINK AND RE-WRITE!

```
10 CLS:DIMA(70)
20 DATA"******"
30 DATA"*     "
40 DATA"     *"
50 DATA"*    *"
60 FORI=1TO4:READA$(I):NEXT
70 DATA1,4,4,4,4,4,1,3,3,3,3,3,3,3,1,3,3,1,2,2,1,1,3,3,1,3,3,1
80 DATA4,4,4,1,3,3,3
90 DATA1,2,2,1,3,3,1,1,2,2,1,4,4,1,1,3,3,3,3,3,3,1,4,4,1,4,4,1
100 DATA1,4,4,1,3,3,1
110 FORI=1TO70:READA(I):NEXT
120 INPUTX
130 FORI=1TO7:PRINTA$(A(X*7+I)):NEXT
140 GOTO120
```

```
? 3
******
     *
     *
******
     *
     *
******
```

I DECIDED, INSTEAD OF BUILDING AN ARRAY OF INFORMATION FOR THE FORMATION OF THE CHARACTERS, I WOULD PLACE IT ALL IN ONE STRING. (B$ IN THE PROGRAM BELOW). I COULD THEN EXTRACT THE INFORMATION USING 'VAL' AND 'MID$' FUNCTIONS. IN ORDER TO RUN THE PROGRAM CONTINUOUSLY, THE 'L' 'FOR-NEXT' LOOP WAS ALSO ADDED. I ALSO TOOK THE OPPORTUNITY TO CHANGE THE MATRIX LAYOUT TO 5X7.

```
10 CLS:DATA"*****","*","    *","*   *":B$="14444413333333133122113313314441333122133112214411333333144144114413
31":FORI=1TO4:READA$(I):NEXT:FORL=0TO1STEP0:INPUTX:FORI=1TO7:PRINTA$(VAL(MID$(B$,X*7+I,1))):NEXT:NEXT
```

```
? 3
*****
     *
     *
*****
     *
     *
*****
```

ONCE AGAIN, IT WAS PRODUCING A NUMBER CLEARLY AND REASONABLY QUICKLY, BUT THERE STILL REMAINED ONE MAIN PROBLEM. I WOULD ALWAYS NEED TO PRINT MORE THAN ONE NUMBER AT A TIME. UP TO SEVEN WOULD BE IDEAL, THOUGH THERE IS ROOM FOR EIGHT!

A REASONABLY SIMPLE TASK IN THE USUAL RUN OF THINGS, BUT I STILL WANTED TO KEEP THE PROGRAM AS A ONE-LINER! SO; I CHANGED THE INPUT STATEMENT TO A STRING, X$, AND CALCULATED THE LENGTH OF THE INPUT-NUMBER STRING. THEN I ADDED AN ADDITIONAL LOOP (FOR NEXT T) TO PRINT EACH NUMBER'S 7TH PART FORMATION ACROSS THE SCREEN. ONCE AGAIN, I EXTRACTED THE INFORMATION USING

'VAL' AND 'MID$' FUNCTIONS. I ALSO NEEDED TO ADD A COUPLE OF SPACES TO THE PRINT STATEMENT TO FILL-OUT THE NUMBERS. I ALSO HAD TO FILL-OUT THE SINGLE "*" TO THE 5 CHARACTER MATRIX SIZE. BECAUSE OF THESE CHANGES, I HAD TO SACRIFICE THE CONTINUOUS LOOP. HOWEVER, SINCE IT'S ONLY A MATTER OF TYPING 'RUN' EACH TIME I WANTED TO PRODUCE A LABEL, I WASN'T TOO WORRIED. I ALSO DECIDED TO CHANGE THE "*" TO A "@" FOR NO OTHER REASON THAN I PREFERRED IT!

AS YOU'VE SEEN, THESE LITTLE PROGRAMS CAN TURN OUT A BIT LIKE 'TOPSIE', BUT THEY ARE SUCH GREAT FUN!

DON'T FORGET, TO CHANGE THE PRINT TO LPRINT WHEN NEEDED.

```
? 258973
@@@@@   @@@@@   @@@@@   @@@@@   @@@@@   @@@@@
    @   @       @   @   @   @       @       @
    @   @       @   @   @   @       @       @
@@@@@   @@@@@   @@@@@   @@@@@       @   @@@@@
@           @   @   @       @       @       @
@           @   @   @       @       @       @
@@@@@   @@@@@   @@@@@   @@@@@       @   @@@@@
READY
>
```

```
10 CLS:DATA"@@@@@","@   ","   @","@   @":B$="14444413333333133
12211331331444133312213311221441133333314414411441331":FORI=1TO4
:READA$(I):NEXT:INPUTX$:C=LEN(X$):FORI=1TO7:FORT=1TOC:PRINTA$(VA
L(MID$(B$,(VAL(MID$(X$,T,1)))*7+I,1)));"   ";:NEXT:PRINT:NEXT
```




IN MAV'S WORKSHOP - SUPPLEMENTAL
by Ian Mavric

CURING MODEL 1 RS232C SERIAL BOARD WOES - PART 1

In my capacity as a TRS-80 repair specialist[1] I see lots of hardware and many variations of it that the more casual TRS-80 user may not be aware of. Take the humble Model 1 RS232 board[2] which fits inside the Expansion Interface in its own little compartment. Most people only think there is one type, but there are actually two, an early one and a later one. Today's discussion will centre around the Baud Rate Generator (BRG) chip and its power supply circuitry.

Three main things go wrong with the RS232 board: the BRG, the UART, and the line drivers. The most common one being the BRG. I was called in to help John Benson and Terry Stewart solve a problem which manifested itself as a serial port with a BRG which was overheating and otherwise not working[3]. The boys noticed that early M1 RS232 boards from 1978-79 had a very pale bluey-green circuit board and a BR2941 BRG. They were comparing with a later board which has a dark-green circuit board and a BR1941 BRG.

I was able to compare both boards side by side, due to having both on hand, and could quickly see three extra components on the early version, R10, R11 and C9. Examination of these parts revealed that they are required to feed -5V to Pin 10 of the BRG (U10/10). The early BR2941 BRG needs -5V on U10/10 but the later BR1941 BRG doesn't. Examination of the data-sheets for the two BRG's shows that this is the only difference. Terry tested this theory by installing a BR1941 in the early board with Pin 10 bent out so it didn't go into the socket and it worked.

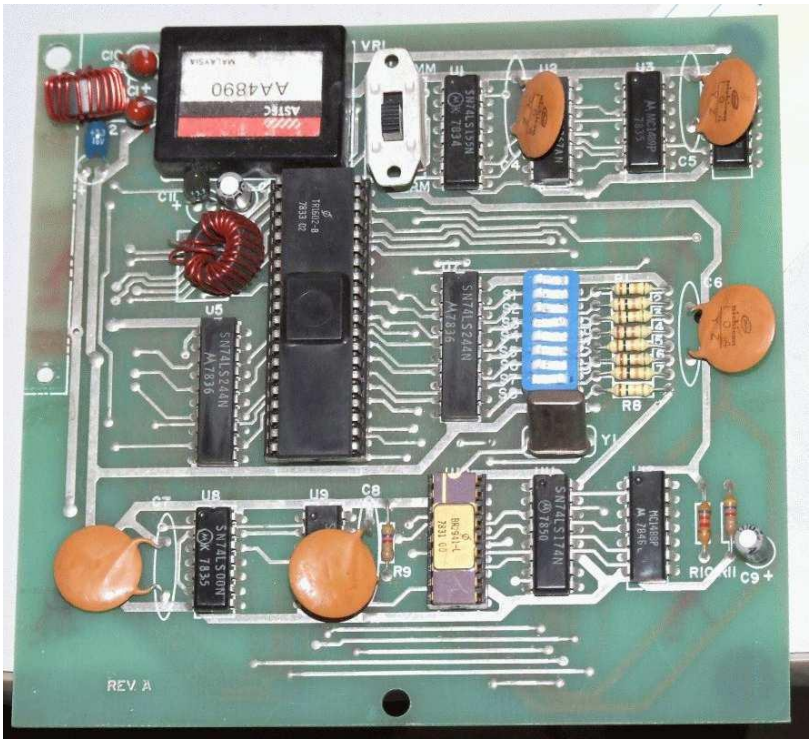
Wanting to set some steadfast rules for replacing the BRG in the future I compared the data sheets for a number of compatible devices including the COM8116, COM5016, AY-5-8116, and the WD1943PD, which is the BRG in the Model 4P[4]. And while the all vary in capabilities, they are all backwards compatible with the BR1941/2941. So I have summarised it in this statement:
"When replacing any BRG in any M1 RS232 board with a modern equivalent, always bend out Pin 10."
I tested this first by installing the BRG out of an unused 4P motherboard I had in my parts-bin and it worked, and then I bought some COM8116 BRG's off eBay and tested one of those which also worked. COM8116 seems to be the most common BRG available for purchase online.

NEXT TIME I will discuss issues pertaining to the Uart, Line drivers and other aspects of the M1 RS232 board.

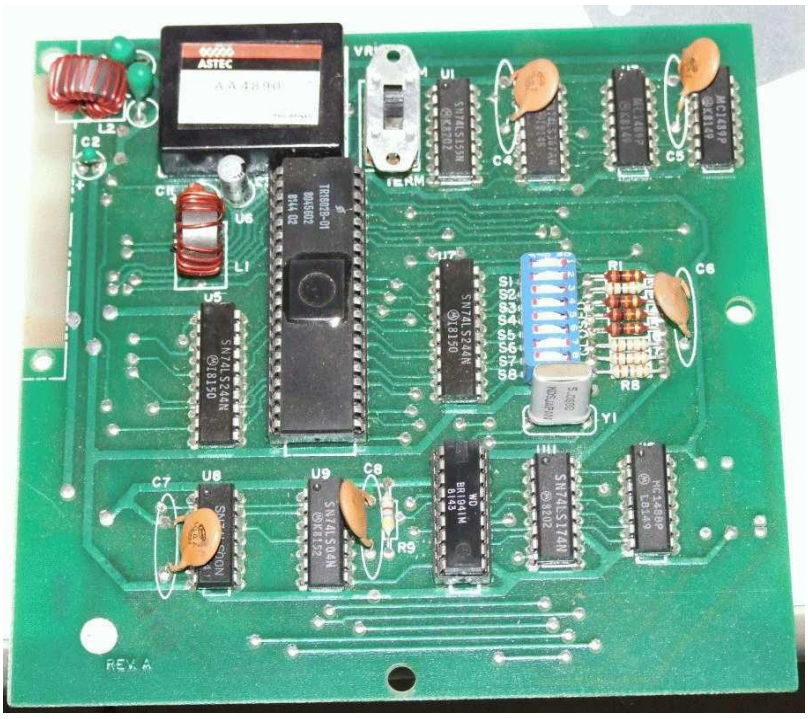
ianm@trs-80.com

REFERENCES:

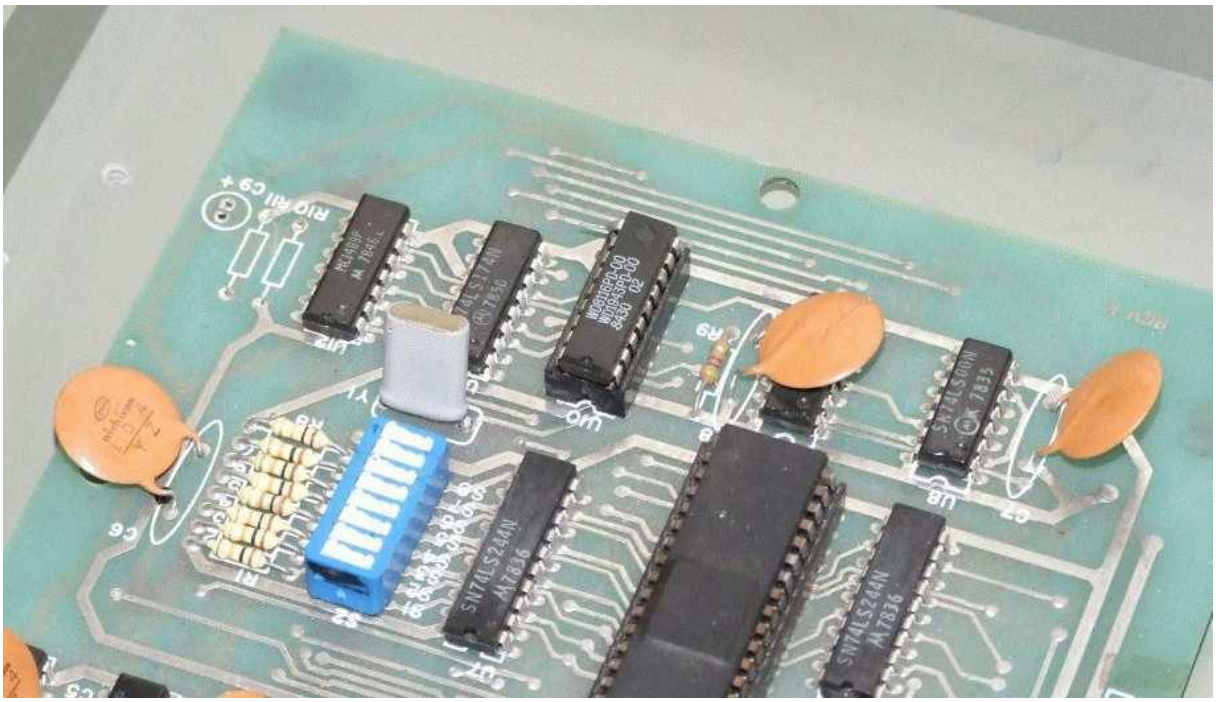
1. http://ianmav.customer.netspace.net.au/trs80/
2. "TRS-80 Catalog" RSC-5, Sept 1981 pp. 13.
3. http://www.classic-computers.org.nz/blog/2011-10-29-trs80-model1-rs232.htm
4. Model 4/4P Technical Reference Manual, Tandy Corp, Cat No. 26-2119, 1985, pp. 142-3 "4.2.16: RS-232-C Circuit".



EARLY RS232 WITH BR2941 AND R10 R11 AND C9



LATE RS232 WITH BR1941 AND NO R10 R11 C9

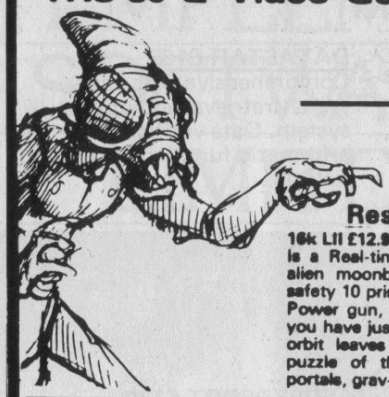


MODEL 4P BRG INSTALLED IN A M1 RS232 BOARD - NOTICE PIN 10 AWAY FROM THE SOCKET.



Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.

TRS-80 & Video Genie Software from ALGRAY



Starquest by Automated Simulations

Rescue at Rigel
16k LII £12.95, 32k TRSDOS £14.95
Is a Real-time Graphic adventure into an alien moonbase to rescue and beam to safety 10 prisoners. You are equipped with Power gun, Blaster and Force shield. But you have just one hour before your ship in orbit leaves and you have to solve the puzzle of the Moonbase with it's tele-portals, grav-shafts and various life-forms.

Star Warrior
16k LII £15.95, 32k TRSDOS £17.95
In this real-time graphic adventure onto the Planet Fornax to destroy the Tyranical Governor. With a choice of Battle Suits, you can fly, jump, walk over swamps, mountains, forests, plains. Battle Heavy Tanks, Maulers, Flitters, Forts. You have a choice of two scenarios, 5 levels of difficulty and competitive scoring system.

ADVENTURES:

Death Dreadnaught — 16k LII £8.95
You are alone (or are you?), on an alien spacecraft with signs of death and carnage all around. With just one lifeboat. Find food air, and fuel, before the ALIEN gets to you and rips your head off to wear as a good luck charm.

Atlantian Odyessy — 16k LII £8.95
Cross Oceans, climb mountains, explore underwater temples in your quest to find the fabled City of Atlantis.

Deadly Dungeon — 16k LII £8.95
Are you fed-up with Mummy's that sneak up and grab your hard won treasures, well now you get the chance to fight back, at the various inhabitants of this 3 level dungeon.

This Month's Bargains:

Space Trek IV.O — 16k LII £4.95
Take your choice of fighting off the TIE fighters on your attempt on the Death Star or try and control the economies of two neighbouring planets in: Population Simulation.

I Ching — 16k LII £4.95
What does the future hold in store? Enter your name for a personal reading or a key-word for a general forecast on a particular event. Extensive use of graphics.

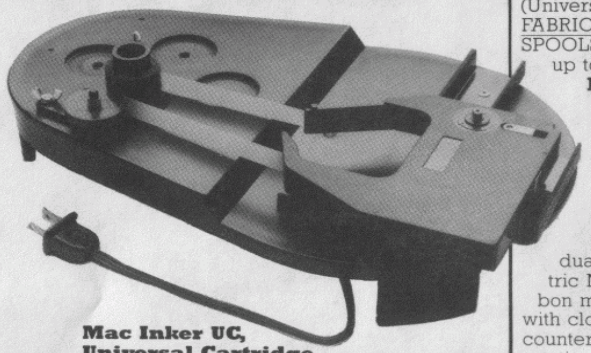
Please make cheques/PO's payable to Algray.

Tel: Barnsley (0226) 83199

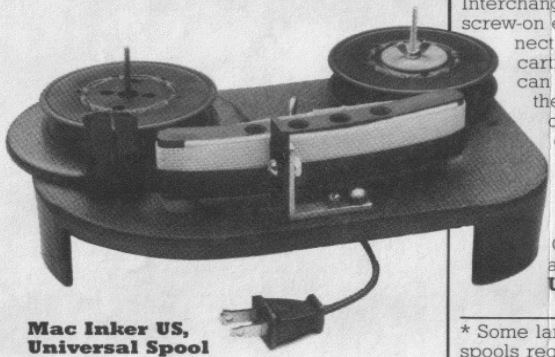
Send to: Algray, Algray House, 33 Bradbury Street, Barnsley, South Yorkshire S70 6AQ.

All prices inclusive of VAT, Post & Packing

The Universals Are Here!



Mac Inker UC, Universal Cartridge



Mac Inker US, Universal Spool

Mac Inker UC

(Universal Cartridge) supports just about all FABRIC RIBBON CARTRIDGES.*

Mac Inker US

(Universal Spool) supports all FABRIC RIBBON SPOOLS (up to 1" in width and up to 4" in diameter).

Mac Inker UC (New US Patent Pending) consists of:

1) A Universal Base ("UNIBASE") and Inking Roller.

2) Interchangeable and add-on Drivers. A dual, Unibase-mounted Electric Motor Drive provides ribbon motion to any cartridge with clockwise or counterclockwise ribbon rotation.

Interchangeable drivers (i.e. screw-on elements that connect the motor to the cartridge adaptor), can be mounted into the cw or ccw drive, depending on the cartridge.

Do you have more than one printer, or do you buy another printer? Just get from COMPUTER FRIENDS another **Mac Inker UC Driver**, at considerably less than the cost of ONE average cartridge. **Mac Inker UC** goes to work for you immediately, dramatically improving print-out quality and practically eliminating ribbon/cartridge replacement costs.

Do you wish to change color? Get another INK KIT (Roller + Bottle of Colored Ink) and use Mac Inker to re-ink your fading color cartridges. Or, get an uninked cartridge from COMPUTER FRIENDS, use Mac Inker to ink the cartridge in the color you choose, then re-ink when needed.

MAC INKER US (Universal Spool) uses the same principle of operation with two design differences:

A) The Inking System ("Ink Transfer") is stationary.

B) A Microswitch disengages the motor automatically when the ribbon has been re-inked and wound-up on the take-up, motor-driven spool.

NOTE Given the feasibility and demand, we have now available a UNIVERSAL SPOOL MAC INKER, that can also support all EPSON 70/80/100* cartridges (see price list).

---

*Epson is a Trademark of Epson Corp.

---

* Some large cartridges or tall spools require their own dedicated Mac Inker. Harmonica type and 'Zip Pack' type cartridges require a dedicated Mac Inker. Please refer to price list.

As of June 85, over 40,000 MAC INKERS are in operation in the US & overseas.

HOW DOES MAC INKER WORK

In three easy steps:

1) Load cartridge (or Spools) onto the Mac Inker table and wind ribbon around inking element.

2) Put a few drops of ink into ink reservoir.

3) Start motor.

As the ribbon passes around the inking element, ink is transferred to the ribbon via a capillary mesh that ensures an even ink distribution. Our dot-matrix, emulsified, lubricant ink complies or exceeds manufacturer's specifications for safe print-head operation. Ink provides excellent performance also with daisy-wheel printers.

Mac Inker(s) with 220V motors available. Dimensions: **Mac Inker UC** (12"Lx8"W x2"H – 30x20x5cm). **Mac Inker US** (9"Lx6"Wx2"H – 23x15x5cm).

MAC INKER GIVES YOU THESE ADVANTAGES

A) Practically eliminates the cost of replacing ribbons. Average cost of re-inking a ribbon is 5 cents. A good quality fabric ribbon will keep printing 40-50-100 times, depending on original length. In our tests, one re-inked Epson* 80 ribbon has exceeded the manufacturer's estimated print-head life. Customer feedback indicates similar and consistent results across the span of printer types.

B) You will produce forever crisp, high quality, legible reports (graphic or alphanumeric).

C) Extended component life. By re-inking the ribbon when print-out quality deteriorates, you can use the head strike setting at a minimum. This reduces the wear both on the ribbon and the print-head.

D) You can add color to your print-outs by inking (and then re-inking) an uninked ribbon in any of our standard colors, (besides black): blue, red, brown, green, purple, yellow.

HOW TO ORDER

A) Call or write to COMPUTER FRIENDS and specify the printer model you have.

B) Select from price list the Mac Inker model #. If customer prefers it, we can still provide (for most cartridges) the first generation Mac Inker(s), dedicated uniquely to each type of cartridge.

C) Send check or money order to address below. Purchase Orders accepted, subject to credit approval. Telephone orders accepted 24 hours/day. VISA/MC accepted.

Computer Friends, Inc.

14250 N.W. Science Park Drive
Portland, Oregon 97229
(503) 626-2291
Telex 4949559 CF
FAX (503)643-5379

Order toll free
1-800-547-3303

Warranted
Made in USA
US Patent Pending

MacInker™

MAC INKER

We generate a lot of paper output here. With all the editing, rewriting, listing and word processing that we do, printer ribbons quickly become a major expense. For a Radio Shack Line Printer VI, the ribbon costs $11.95 plus tax. And we seemed to use one up every time we just printed out a 10,000 name mailing list. No longer are ribbons a major expense.

The Mac Inker has truly done what the manufacturer promised. It has practically eliminated the cost of replacing ribbons. We can re-ink a ribbon for about a nickel, and it works as good as new. One ribbon has been re-inked at least a dozen times now, and it is still holding up. In a few cases, the re-inked ribbon has given better print than when it was new.

The Mac Inker is quite a simple device. It is made of metal, has a small drive motor, an inking head, and it is designed to hold a specific ribbon cartridge.* You just mount your cartridge, guide the ribbon around the ink roller, turn on the motor and leave it alone for about 20 minutes. That's all. The ribbon is now ready to use, after it has had a chance to dry slightly.

We have been using one for six months, re-inked at least 40 ribbons and are still on our first bottle of ink (refills are $3.00). You can also get colored ink (red, blue, green, brown, yellow, purple, orange, gold, silver). We find that our printers work fine and there is no build up of residue or ink on the heads. Be sure to specify exactly what printers you have so that the Mac Inker you get will fit your ribbon cartridge or spool. It is well built, works, saves money and is easy to use. Buy one.

Cameron C. Brown
June 1983 page 103.

*This review covers our very first model, before the advent of the Universal MacInker and, of course, does not describe the many improvements made since then.

NOTES & COMMENTS - TYPICAL QUESTIONS AND ANSWERS

The review on the left, published in BASIC COMPUTING, was Mac Inker 's first (June 1983!!!). Since then, MAC INKER has been reviewed with equal enthusiasm by a good number of major magazines. We hit a peak of fame when the New York Times and the Chicago Sun Times gave the Mac Inker an excellent review, in the computer section. We have a collection of over 700 unsolicited testimonials. We wish to quote from the first written testimonial we ever received (December 82), written by Father Mick Burns, 701 S Walcott, Casper, Wyoming: "...A rare experience indeed exists when a product is as cost effective as yours! I have always been leery of 'testimonials' making exaggerated claims. If one had been presented accurately representing your product I simply would not have read it!.....Some brief calculations quickly showed full payback on the very first day's use of my now treasured MAC INKER! Let me sincerely thank you for such fine product!" Among our recent customers the FAA (Federal Aviation Administration) is using a multicolor red-black MacInker in each of their field offices. Texas Instruments has reported to us, in writing, net savings of $34,000/year using our MacInkers. Schools, Colleges, Universities, Radio-TV Stations, Banks, Army, Navy and Air Force are among our constant and repeated customers. Several customers have reported to us having received 'best idea' awards by their employer, after having proven to their company the savings made possible by MacInker.

Now to some often asked questions.

1) **How many times can a fabric ribbon be re-inked?** It is a function of the length of the ribbon. A 20 yard ribbon (i.e. Epson 80 series) can be re-inked 60-100 times before it shows signs of wear. We run a 2 yards loop in an Epson 80 cartridge and re-inked it until the ribbon had worn out. By extrapolating results, the ribbon had outlasted the manufacturer's expected life of the print-head.

2) **How can a ribbon last so long?** It is a matter of cooling, fabric suppleness and probability. During a complete loop, any one point of the ribbon is not even, statistically, struck once. Even if it were, the fabric has time to recover from the impact (say 1 hour), before the same point passes again in front of the print-head. A ribbon, **regularly re-inked with lubricant, carbonless ink**, will remain supple, will not fray and the ink will help cool the printhead needles where the temperature can reach 480 degree F. Our **ink complies with, or exceeds manufacturer's fabric ribbon ink specifications.** For those interested we have a paper on the physics and the chemistry of ribbons, inks and their interaction, as well as on data from tests conducted at Computer Friends and at the University of Arizona.

3) **How do I recognize a fabric from a carbon ribbon?** Carbon ribbons cannot be re-inked, or rather 're-carbonized'. The cartridge is 'one way only'. When the take up, internal reel is full, the ribbon cannot be re-wound. Fabric ribbons, in general, have an endless loop.

4) **How do I know when the ribbon has made a complete loop?** Here is a tip. Prior to your first re-inking & using correcting fluid ('liquid paper') dab both sides of the ribbon. Start the MAC INKER motor & measure how long it takes for the mark to re-emerge from the other side of the cartridge. This will give the correct re-inking time. From now on use a timer or alarm etc. We can supply a good quality, heavy duty, AUTOMATIC SHUT-OFF, PLUG-IN TIMER (see back of Mac Inker price list).

5) **What happens if the ribbon is overinked?** There is no damage, but the print-out may be too dark or there may be some bleed through, if the paper stock is thin. Recovery is simple, if required. Wrap the ribbon around a paper towel roll & let the ribbon make a loop, releasing the excess ink onto paper. With a minimum of patience (and by reading the instructions) overinking is practically impossible. A properly re-inked ribbon does not appear significantly different from a dry ribbon. After the first re-inking, you will have an immediate feel for when the ribbon is properly inked. One full ribbon pass around the inking element is generally enough.

6) **What does it mean Universal Cartridge and/or Universal Spool MAC INKER?** The Universal Cartridge MAC INKER consists of a Universal Base (UC) and individual Driver Kits for the various Cartridges. They fit into one of the 2 bi-directional motor drives. (Pl. refer to the beginning of the Price LIst). A 'U' after the model # means that a Driver Kit is available for that printer. Note that there are overlaps, that is, the same driver kit supports cartridges of different names and different external characteristics. **For example**, you wish to re-ink Epson 80, Epson 100 and Toshiba 1350 cartridges. You need the Universal Base (UC) with the 271UK (Epson driver) and the 287UK (Toshiba 1350 driver). Mounting or removing the drivers is a matter of seconds. However we normally ship the MAC INKER(s) with a driver already mounted. The Universal Spool MAC INKER base with the 1/2" width S505 adaptor re-inks the majority of all spools. Some spools have large inner diameters. The letters 'A' or 'B' etc. refer to the centering adaptors required to mount the spools. For example the IBM 3262 spool uses the S606A adaptor, where 'S' means spool, '606' indicates the 1" width adaptor and 'A' the centering element.

7) **Is re-inking messy?** With a minimum of care, it is possible not to touch the ribbon, while mounting or removing the cartridge from your MAC INKER. Ink is dispensed from a drip-cap bottle and/or an eye-dropper, for accurate metering. First time users, generally and naturally, tend to rush. Alcohol is a reasonably good cleaner. Soap will wash out the ink after 2 or 3 times. We have found an excellent printing ink cleaner (see price list) in use with the Air Force. We have put it in the price list, as it is the best and gentliest ink cleaner we ever found. It is also an excellent all-round degreaser.

8) **Is any special care needed for the inking elements** between re-inkings? No, the lubricant in the ink will prevent a total dry-out and the dot matrix ink contains no carbon or abrasives. Just shelter the inking elements from excess dust. The Inking elements' life is practically indefinite.

9) **How many inkings will a bottle last for?** It is a function of the length and the width of the ribbon - it also depends on how dry the ribbon was at the beginning. A 2 oz bottle will yield about 25-40 re-inkings on a 20 yards ribbon. If you print a lot, we recommend the pint - there is a saving. All bottles have a metering spout and ink will keep indefinely, if not exposed to strong heat or sunlight.

10) **How to re-ink color cartridges?** We have color cartridges or uninked cartridges for the most popular printers. You can use your MAC INKER to color an uninked cartridge or to re-ink a colored cartridge in its original color. You require a Colored Ink Kit, which includes an inking element and a bottle of ink of the color chosen. With gold and silver color you can re-ink a color cartridge but not ink an uninked ribbon in gold and silver. We have now available Multicolored MAC INKER adaptors for multicolor ribbons (2, 3 and 4 bands). In this case you require the MAC INKER, the appropriate multicolor adaptor and one bottle of ink per each of the colors of your ribbon.

11) **Any concluding thoughts?** The first MAC INKER was sold in May 1982. As we go to print (April 88) we have over 100,000 units in the field. We have constantly improved and upgraded our products, using new designs, techniques and materials (yes!) which have become available in this apparently humble, but useful technology. In 1982, we could never imagine that we would be able to develop a MULTICOLORED RIBBON re-inking technique. We have no association with the Apple Macintosh whatsoever. In fact, we called our product MAC INKER long before the Macintosh, in the hope to be able to write one day, 'OVER 50,000 SOLD'. We are grateful to our Customers, who have made it possible to reach and exceed that goal. We trust that in dealing with us you will find that friendliness has meaning and is not just a commercial platitude.

WARRANTY: 3 months after date of shipment. The only element which can fail is the motor. We have replaced very few motors, during the life of the product.

ASSEMBLER NOTES

Laurie Shields

Introduction

"We have not yet succeeded in answering all your questions ... Indeed; we feel that we have not yet answered any of them. The answers we have found only serve to raise a whole new set of problems. In some ways we feel we are as confused as ever, but we believe that we are confused on a higher plane, about more important things."

Assembler Programming

One school of thought defines assembler programming as the nearest thing to the tortures of the Spanish Inquisition, in its ability to deprive its sufferers from sleep and recognisable human social behaviour; but this opinion cannot truly be relied upon because it totally disregards the self inflicting masochistic aspect of the disease.

Perhaps though we ought to at least consider how the external symptoms are displayed. Usually they take the form of the sufferer ceaselessly keying in at the nearest computer some totally meaningless expressions, that they fondly believe to be 'English type words' and then asking the computer to change the words into equally meaningless numbers. When finished the net effect of their efforts is to achieve the same result as switching the computer off but without the benefit of saving any electricity.

Some of the more expert practioners in the art form have even been known to fill the video picture with one thousand and twenty four '@' characters and generate a buzzing noise as the penultimate climax to their hours of unceasing toil. Ordinary BASIC programmers, who can't get beyond fifty '@' characters before the dreaded 'Out of String Space' appears on their videos, go in awe and trembling before the amazing powers of the assembler.

The following pages on assembler programming and machine code in general on the TRS80 are the result of some considerable pressure from Brian to put down on paper the content of some of the talks and discussions on Zen that have occurred over the past eighteen months. I am deeply indebted to a large number of people especially John Hawthorne who wrote the original Zen and also to Radio Shack for producing the TRS80 and Scripsit as well as countless others who gave me help when I started and didn't know the difference between a Byte and a Bootstrap.

Laurie Shields.
Chesterfield. September 1981.

Chapter 1

To get the best use from these notes you really ought to have a live TRS80 with Zen powered up and working. As there are a number of different versions of Zen with differing degrees of sophistication I shall try to refer only to the original fundamental commands which to a great degree are all upwardly compatible. Zen in its original cassette only form is marketed by Newbear for the TRS80 and other Z80 machines (Sharp, Nascom etc) and in its upgraded enhanced format for the TRS80 Cassette, Aculab or Disc by Laurie Shields Software.

Just to get started and at the same time to use Zen to explore the inside of the TRS80, key the following:

Q5712 <E> and also, by itself X <E>.

Now we have some explaining to do as your video looks like a meaningless jumble of numbers and letters. It actually is displaying part of the Leve12 Rom, a few other bits as well and should be:

```
1650    C5 4E 44 C6 4F 52 D2 45 ENDFORRE    .ND.OR.E
1658    53 45 54 D3 45 54 C3 4C SETSETCL    SET.ET.L
1660    53 C3 4D 44 D2 41 4E 44 SCMDRAND    S.MD.AND
1668    4F 4D CE 45 58 54 C4 41 OMNEXTDA    OM.EXT.A
1670    54 41 C9 4E 50 55 54 C4 TAINPUTD    TA.NPUT.
1678    49 4D D2 45 41 44 CC 45 IMREADLE    IM.EAD.E
1680    54 C7 4F 54 4F D2 55 4E TGOTORUN    T.OTO.UN
1688    C9 46 D2 45 53 54 4F 52 IFRESTOR    .F.ESTOR

 HL   DE   BC   AF   RI   IX   IY   SP   PC
0000 0000 0000 0000 0000
0000 0000 0000 0000      0000 0000 5514 402D
```

If the numbers you have on display for the last two lines are a bit different, DON'T PANIC. It only means that you're on cassette, or floppy tape or are using a different Dos from the one I was using when this was written.
In order to understand any of what Zen is trying to tell us we must first of all make the effort to comprehend the way information, be it program or data, is stored in the computer and the conventions used to present it at the human interface.

Right down at rock bottom most if not all computers store and evaluate numbers in simple binary 0's and 1's. However most of the time we and the computer operate not with these individual Bits but with eight of them at a time called Bytes. Invariably however, just to break the first rule

straight away and purely for our convenience, these bytes are split into half bytes called nibbles, so that we can avoid having to do mental arithmetic to base 256.

By thinking in terms of 4 bits at a time, the powers-at-be have forced us into straining the brain and understanding hexadecimal arithmetic (base 16), which isn't really too bad, and not much worse than feet and inches. Consider the eight bits below with their numeric values when the bit is set (i.e. equal to 1 rather than 0):

```
Bit number:     7   6   5   4    3   2   1   0
Decimal val.    128 64  32  16   8   4   2   1
Hex value.      80  40  20  10  08  04  02  01
```

Now it doesn't take an overdose of grey matter to see that the values of bits 0 to 3 can be represented by the right hand half of two hexadecimal digits and the bits 4 to 7 by the left hand one. Even if all the bits 0 to 3 were set giving a value of 1 + 2 + 4 + 8 = 15 this would not exceed the ability of hexadecimal notation, where 15 is represented by F.

Crib Sheet for the newcomers:

```
Decimal:     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Hexadecimal: 0 1 2 3 4 5 6 7 8 9  A  B  C  D  E  F
```

Even Better Crib Sheet for beginners and experts:

```
            U N I T S              Ascii Value
Tens 0  1  2  3  4  5  6  7  8  9
 00 00 01 02 03 04 05 06 07 08 09 Printer &
 10 0A 0B 0C 0D 0E 0F 10 11 12 13 Display
 20 14 15 16 17 18 19 1A 1B 1C 1D Codes
 30 1E 1F 20 21 22 23 24 25 26 27   1"4$%&,
 40 28 29 2A 2B 2C 2D 2E 2F 30 31   ()-+,-./Ol
 50 32 31 34 35 36 37 38 39 3A 3B 23456789:;
 60 3C 3D 3E 3F 40 41 42 43 44 45 <=>7@ABCDE
 70 46 47 48 49 4A 4E 4C 4D 4E 4F FGHIJRLMNO
 80 S0 51 52 53 54 55 56 57 58 59 PQRSTUVWXY
 90 5A 5B 5C 5D 5E 5F 60 61 62 63 Z      abc
100 64 65 66 67 68 69 6A 6B 6C 6D defghijklm
110 6E 6F 70 71 72 73 74 75 76 77 nopgrstuvw
120 78 79 7A 7B 7C 7D 7E 7F 80 81 xyz
130 82 83 84 85 86 87 88 89 BA BE
140 8C 8D 8E 8F 90 91 92 93 94 95 Graphic
150 96 97 98 99 9A 9B 9C 9D 9E 9F Codes
160 A0 Al A2 A3 A4 A5 A6 A7 AS A9 128 to 191.
170 AA AB AC AD AE AF B0 B1 B2 B3
180 B4 B5 B6 B7 B8 B9 BA BB BC BD
190 BE BF C0 C1 C2 C3 C4 CS C6 C7 Space
```

```
200 CB C9 CA CB CC CD CE CF DO Dl Compression
210 D2 D3 D4 D5 D6 D7 D8 D9 DA DB Codes
220 DC DD DE DF E0 El E2 E3 E4 E5 192 to 255
230 E6 E7 E8 E9 EA EB EC ED EE EF
240 FO Fl F2 F3 F4 F5 F6 F7 FS F9
250 FA FB FC FD FE FF
```

In one eight bit byte we have the ability to store any number from 0 (decimal), OOH (hexadecimal) or 0000 0000 (binary) up to 255 (decimal), OFFH (hexadecimal) or 1111 1111 (binary). We have now gone and confused you by putting a 0 in front of the FFH and someone wants to know why. Well the answer is relatively simple and it doesn't mean that we have squired an extra ninth bit of value zero, it simply follows on from our normal programming convention that if something begins with a letter it is a variable and something beginning with a numeric digit is a number. So to avoid any possible confusion we always (well nearly) put a leading zero in front of any hex number that begins with a letter.

Now this convention is purely one way, is from human to computer. This is because the computer cannot guess as to whether when we key FFH we mean a number, a label or variable or just a plain expletive deletive. When the computer is giving us information the assumption is made that we know what we have asked for and should recognise it when we see it.

So what happened after we keyed Q5712 ? Well the first thing Zen did was to check whether or not we had put an 'H' at the end of our input. Had we done so the Zen would have treated it as hexadecimal; since there wasn't it was treated as an ordinary base ten decimal number and Zen turned the 5712 into hex and rounded the last digit down to the nearest 0 or 8. Decimal 5712 converts exactly to 1650 Hex, in this case no rounding was necessary. The letter Q was interpreted by Zen as a 'Query memory' command so Zen started by printing the value, 1650 and then examined the TRS80's memory starting at 1650 and displaying in hex each byte from 1650 to 1658.

The right hand half of the display is an Ascii representation of the same 8 bytes of memory but shown so that the bytes whose values are greater than 127 and not normally considered as displayable Ascii art. depicted in two formats for maximum information. On the extreme, right is the academically correct display where any bytes greater than 127 are shown as full stops. On the left is a much more

interesting presentation as all the bytes are displayed as recognisable letters. This conversion from unprintable to printable is achieved by Zen subtracting the value 128 from any number greater than 127.

Now this is no accident or defect in the ROM because if we look closely at the letters we see that what is shown is a listing of some of BASIC's instruction codes such as END FOR RESET etc. except that there are no spaces in between. This is a look-up table used by Leve12 BASIC to check on the spelling in your Basic program and if what you key in doesn't match any of the words in the table you get a syntax error. By squashing all the words together Microsoft saved themselves quite a lot of ROM that would been just spaces but they then had to figure out a way of knowing when they had come to the end of one keyword and the start of the next. This they did by adding 126 or 80Hex to the first letter of each word and then, simply by testing if any letter was bigger than 127, they knew where each word started and so to get the correct spelling all they had to was deduct 128 to obtain the correct Ascii value.
This in machine code is the equivalent of starting every word with an enormous 'capital' letter.

Now adding, subtracting and testing for this magical 128 is not very difficult in machine code as it depends solely on whether Bit 7 is set or not, a condition that is very readily detected with the Z80 chip.

So Zen will tell us anything we want to know about the memory in our TRS80, just key the letter Q followed by the address and there it all is. If you want the next 64 Bytes there's no need to give the address again as Zen remembers where the last block finished so just key Q <E>.
Should you require a printout of the Q display then key Shift Q instead of Q and the display will be routed to the lineprinter driver in Zen.

TRS - 80 V/G

LETTER WRITER Compiled word processor **£12.95**. Phone: 082584 603 evenings for details.

SIMPLY SOFT, 109 South Street, East Hoathly, Lewes, Sussex.

Chapter 2

Now what on earth are we to make of the second part of the display:

```
 HL   DE   BC   AF   RI   IX   IY   SP   PC
0000 0000 0000 0000 0000
0000 0000 0000 0000      0000 0000 5514 402D
```

Well lets think of a BASIC program that has the line:

```
100 LET Y = 4 + 3 * 3
```

and imagine that we stopped the program with the Break key just as the Z80 had finished working out that line.

We could then tell the computer to PRINT Y and get the answer 13. Obviously, the variable Y has had it's value stored somewhere, after it was evaluated and, of course that somewhere is in a few bytes of RAM that BASIC has allocated from the memory that was not needed for other things. Now in the middle of processing that line of code the computer had first of all to get the number 4 on one hand, then calculate 3 times 3 (probably using it's toes); add the two numbers together (somebody else's toes); check whether there were any more sums to do and then, and only then, put the result into RAM. Thus releasing the fingers, toes and brain for the next sum.

In the Z80 chip the arithmetical brain is known as the Accumulator or, register, where any numbers in the range 0 to 255 can be added, subtracted, compared or fiddled about with. Since the A register can only hold numbers up to the value 255, consider what will happen if the Z80 tries to evaluate the sum 250 + 20. The result is like adding say 95 and 8 pence. We get 3 pence and an overflow or carry of 1 into the pounds.

```
The result of 250 - 20 in the Z80 is 14 with carry i.e.
             250 + 20 = 270 = too big.
         therefore 270 - 256 = 14 + Carry.
```

the same in hexadecimal:

```
    Carry   Sixteens    Units
                F           A   (15*16 + 10 = 250)
    add         1           4   (01*16 +  4 =  20)
        ======================
                            E   ( A+4 = 10+4 = 14)
                0               (15+1 = 16   = 0 + C)
    1
    ==========================
                0           E   with carry.
```

Since there isn't any room in the A register (remember there's just 8 bits) to store the information about the carry condition we have to have another register alongside where interesting things such as Zero, Carry, Positive or Negative states are brought to the Z80's attention. As this register is effectively waving flags at, or because of the Accumulator, it is known as the Flag or F register.
These two registers are the best of friends and very little activity in the A register is not known about in the F. However there is one very important exception and that is the simple loading of data into the A register. F couldn't care a fig - he's only bothered about exiting events like adds, subtracts and compares.

That's sorted the brain out, where are the fingers and toes. Well theses are the other registers and we have quite a few. Just to make life simple forget completely about RI (that's something to do with refresh and interrupts - all very nasty), forget nearly completely about. IX and IY, and just put to one side for the moment SP (this stands: for stack pointer - very important).

PC stands for program counter and is simply the address of the next byte of machine code that the Z80 would execute. That leaves HL, BC and DE with a repeat underneath alongside another AF Those on the bottom line are there just to confuse you, they are known as the alternative set of registers and represent the fact that the Z80 is very nearly two microprocessors in one with a duplicate set of registers available for use if necessary. Since Microsoft didn't find it necessary to use them to get the TRS80 to work I doubt if we will need them either.

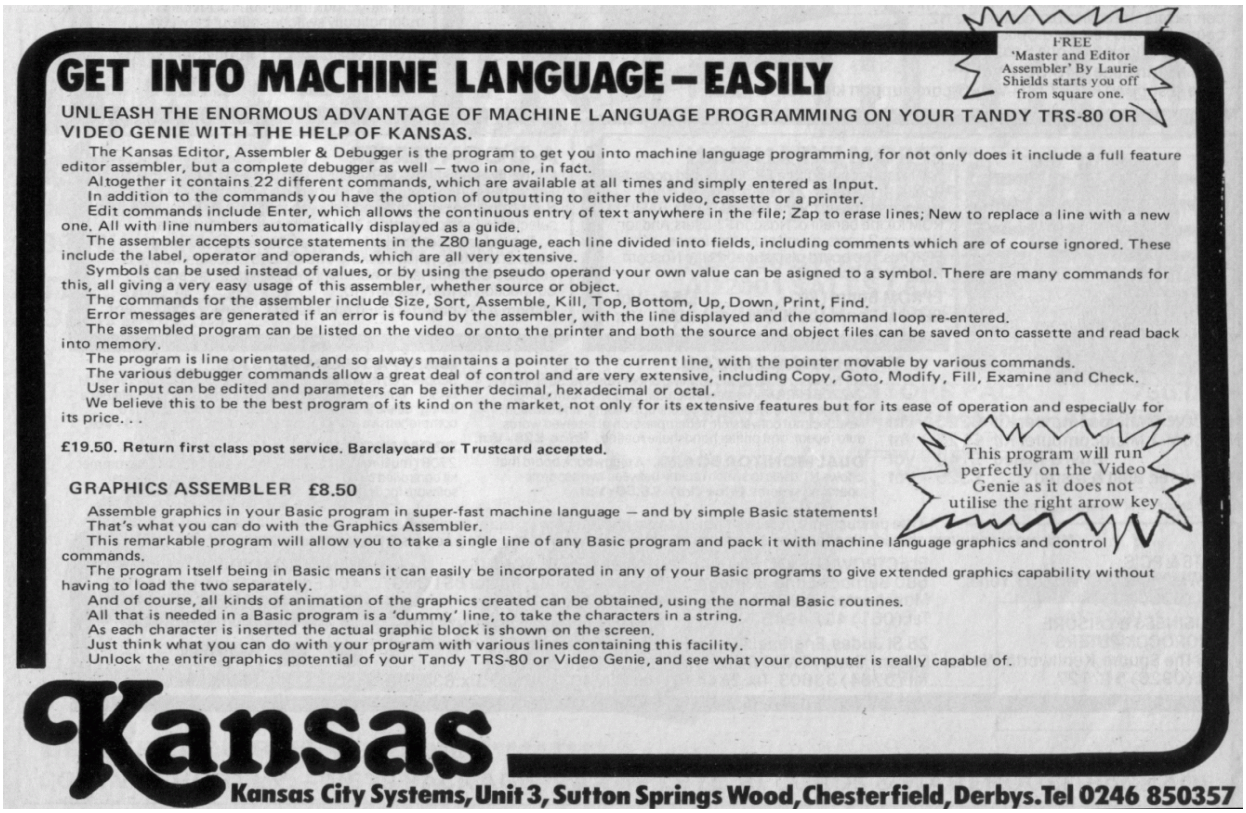
So we are left with HL, BC and DE. These can be used as 6 independent 1 Byte (8 bit) registers H, L, B, C, D and E or as 3 combined 16 bit ones depending on what you want them for. The HL pair (H stands for High and L for Low - tens and units if you like except the tens are two hundred and fifty sixes) is the cleverest of the three and has a limited brain of it's own. Liken it to the fingers, right hand for H and left for L. The others BC and DE are relatively dumb but we couldn't get along without them. B is perhaps the most useful and has a great propensity for counting. Just the thing for stepping through a program and keeping a count of the number of steps.

Unfortunately it's a little bit more complicated than just pressing the break key to stop in the middle of a machine code program and find out just what are the values held in the various registers but it can be done when using the de-

bugging facilities that Zen offers us. It is then that the 'X' display becomes meaningful because it shows us just exactly what the Z80 was up to at the stage of the program when we set the breakpoint to interrupt it. So until we have used the de-bugger to jump into and break out of a program the values displayed by the X command are just as meaningless as telling the computer to PRINT Y would be before we ran the BASIC program that gave Y its value.

Just before we leave the registers we should note that as the register pairs HL, BC, and DE can perform as 16 bit registers, it is possible for them to comprehend a number as big as FFFFH or 65536 and as little as 0. This makes them very useful in pointing to a particular byte in memory for storage or retrieval. HL, remember the pair with very little brain, can also do adds and take-aways in 16 bit arithmetic provided the other number is stored in either BC or DE.
PC (program counter), SP (stack pointer), IX and IY (index registers) are all inseparable 16 bit registers intended for storing memory addresses.

It is best not to try and grasp the whole complexities of the Z80 chip in one go. Instead just get to know each register in turn as you use them in your programs. To start with it is sufficient to know that A holds a small number and can do clever sums with the help of the F, and that HL and BC can hold big numbers. If we need to store a lot of small numbers HL and BC can be treated separately as H, L, B and C.



Chapter 3

Negative Numbers

Negative numbers are a concept that the Z80 would rather do without. Either there's a 1 there, in which case something is there, or there's a 0 and there's nothing there. The preposterous idea that there is less than something there is just too ridiculous.
But since we humans insist on using this hair brained idea then the Z80 tries its best to comply.

Picture the number 255 Dec or OFF Hex in binary notation, (remembering that that leading zero is just to stop the computer thinking you're swearing at it):

OFF Hex = Binary 1111 1111 Beautiful!

Since we always write our minus signs on the left of a number lets turf the leftmost bit over on its side like this: -111 1111. Now we are getting somewhere. That's a negative number if ever I saw one; but what is it? The easy way to find out is to add 1 to it and see what happens. Ready?

```
    - 1 1 1 1 1 1 1
add 0 0 0 0 0 0 0 1
    ---------------- Left blank for you to fill in.

    ----------------
```

Stand by to check your answer:
One plus one equals zero with carry.
One plus zero plus carry equals zero with carry.
One plus zero plus carry equals zero with carry.
One plus zero plus carry equals zero with carry.
That's half of it done.

One plus zero plus carry equals zero with carry.
One plus zero plus carry equals zero with carry.
One plus zero plus carry equals zero with carry.
Remember that the minus sign is really just a Bit = 1 lying on its side so the last one:

One plus zero plus carry equals zero with carry falling off the end (advanced students might guess that it didn't really fall off it went into the Carry flag, more about that later).

Answer: 0 0 0 0 0 0 0 0 = 0 Dec.

So if adding 1 makes the original number 0 then the original number must have been -1. (Get out of that.)

OK I know it didn't look like minus one but I did warn you that the Z80 wasn't happy about it either.

The following is the way the Accumulator would store the number -128 if you wanted it to: 1 0 0 0 0 0 0 0 ie 80 Hex. (Try adding 1 a hundred and twenty eight times.)

Minor Diversion to talk about flags:

As we have a potentially dangerous situation here, since a number in the A register could be treated as a large positive one (between 121 and 255) or a smaller negative one we had better do some flag waving to check on where we are. To help us is the friendly unassuming F register with a very useful flag which is waved whenever one of the instructions that can affect the flags detects that Bit 7 is set at 1, i.e. negative.

We can use this flag in our assembler program decision making by utilising instructions like:
JUMP if POSITIVE to ADDRESS, or
CALL SUBROUTINE if MINUS.

The trio of flags Zero, Sign and Carry form the backbone of virtually all the logic in most machine code programs. There are three other flags:

Parity/Overflow or P/V which some clever people understand. Half Carry or E used in decimal arithmetic I think. Add./ Subtract or N again used in decimal type arithmetic, but as I've never had occasion to use them, their exact function and usefulness hasn't. sunk in yet.

Back to working out negative numbers

Right you've got the picture. Treat Bit 7 (the leftmost) as a sign bit, work out what the others would come to if positive, subtract it from the value of Bit 7 and you've got the negative decimal equivalent. Just to show off the Z80 can do the same sort of thing with 16 Bit (two byte) numbers like OFFFF Hex or 65535 as well. In binary it looks like:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 as before for the sign:
- 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

If we added 1 to it like last time we would get 0 so this must be the way -1 Dec would be stored as a 16 Bit two byte number.

Now sixteen bit numbers are used a lot in the Z80 for memory addresses and one of the things that baffles most BASIC programmers for a while is the problem of PEEKS and POKES above the 16K boundary. Lets examine the numbers in that region; firstly the last byte in 16K and then the next one and finally the last byte in 48K (These sizes of course refer to RAM, we must add another 16K to them for the ROM, keyboard, video etc).

```
Top of 16K - 7FFFHex  = 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Next Byte  - 8000Hex  = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Top of 48K — 0FFFFHex = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

If the Z80 has been told to interpret these as 'human' numbers then the second two are obviously negative. Setting these numbers out with the values of each bit, we get:

```
                          B I T S
Sign    14    13  12   11   10   9   8   7   6  5  4 3 2 1 0
32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1

7FFF =
   0     1    1    1    1    1   1    1   1   1  1  1 1 1 1 1
     = Positive = 32767

8000 =
   1     0    0    0    0    0   0    0   0   0  0  0 0 0 0 0
     = Negative = -( 32768 - 0 ) = -32768

FFFF =
   1     1    1    1    1    1   1    1   1   1  1  1 1 1 1 1
     = Negative = -( 32678 - 32767 ) = -1
```

Don't worry if it doesn't all sink in first time, but when it eventually does penetrate the grey matter you will have a sound grasp of how BASIC handles its Integers, known in the trade as 'Signed 16 Bit Integers' and you will also appreciate why in Basic, the range of values that an integer can hold is limited to -32768 to +32767.

FOR ADVANCED BASIC PROGRAMMERS ONLY

```
10 DEFINT=A - Z
20 Y - 7 + 256 * 8
30 V - VARPTR( Y )
40 PRINT PEEK( V), PEEK( V + 1 )
50 INPUT 'NEW VALUE "; Y
60 GOTO 30
70 END
```

Explanation.

```
10 Make everything Integers
20 Give Y the value 0807Hex and store it in RAM.
30 Find out where Y is stored.
40 Have a peek at what's there.
50 Try your own numbers.
```

Alternatively you could try POKEing some values into V and V+1 and then finding out what Y had become.

What you should observe is that any value of Y is held in RAM such that the remainder from dividing by 256 is stored at the byte in memory pointed to by VARPTR(Y) and the number of 256's in Y is stored at the next byte i.e. VARPTR(Y)+1.

This is **VERY IMPORTANT** and we shall use this fact in PART 2 with USR routines.





The *BEST* in TRS-80s

Call The Right Stuff

Ask for Ian
The number is +61 416 184 893

That's The Right Stuff
And he's in Melbourne







http://ianmav.customer.netspace.net.au/trs80/

KEYBOARD PRACTICE

THE GRAND-KIDS HAVE BOTH (THE BOYS THAT IS), JOINED THE I.T. CLUB AT THEIR SCHOOL AND ARE FINDING THE PC'S KEYBOARD LAYOUT SOMEWHAT CONFUSING. I BRIEFLY TOLD THEM ABOUT THE ORIGINAL REASON FOR THE 'QWERTY' LAYOUT, (SHOWING HOW THE KEYS COULD JAMB ON MY OLD TYPEWRITER) AND I'M SURE THEY WERE BOTH SUITABLY UNIMPRESSED! THEY DO MANAGE REASONABLY WELL, BUT I THOUGHT, IF FINDING THE CORRECT KEY CAN BE FUN, IT WOULD MAKE REMEMBERING THAT MUCH EASIER.

SO, I CAME UP WITH AN IDEA WHICH I THOUGHT MIGHT HELP. I WANTED THE BOYS TO PRESS A SPECIFIC, CHOSEN KEY, WITHOUT LOOKING AT THE KEYBOARD AND BY WAY OF CONFIRMATION THAT IT WAS THE CORRECT KEY, THE SCREEN WOULD BE FILLED WITH 1024 OF THEM!

NOTHING TOO MAJOR, I'M SURE YOU'LL AGREE. THE ONLY PROBLEM I COULD THINK OF WAS THAT BASIC WOULD BE JUST A BIT TOO SLOW AT FILLING THE SCREEN. SO, THERE WOULD BE ONLY ONE THING FOR IT. IT WOULD REQUIRE A FEW LINES OF ASSEMBLER TO SPEED THINGS UP A BIT.

AS ALWAYS, 'ZEN' CAME TO THE RESCUE AND WITH THE 'LDIR' COMMAND THIS WAS ACHIEVED EASILY. (A BIT TOO EASILY AS IT TURNED OUT- SEE THE NEXT PAGE!)

Page 1

```
1                       ORG  7F00H
2                       LOAD 7F00H    ;prog placed in
                                      ;memory here
3 7F00 21003C           LD   HL,15360;start of screen
4 7F03 11013C           LD   DE,15361;start + 1
5 7F06 01FF03           LD   BC,1023  ;no. of times
                                      ;to loop round
6 7F09 EDB0             LDIR          ;fill screen
7 7F0B C9               RET           ;back to BASIC
8                       END
```

THE LITTLE PROGRAM, ABOVE, CONFIRMS THAT.
I THINK THIS WAS MY FIRST, EVER, ASSEMBLY PROGRAM THAT ASSEMBLE FIRST TIME, ERROR FREE!

ALL THAT WAS LEFT TO DO WAS CALCULATE THE DECIMAL VALUES (FROM THE 3RD COLUMN, VIZ. 21 00 3C ETC, WHICH HAD BEEN PLACED IN MEMORY AT 7F00H BY ZEN). I WOULD THEN PLACE THEM IN MY BASIC PROGRAM AS DATA STATEMENTS AND LOAD THEM INTO AN ARRAY.

FOR THIS, I'M TOTALLY AMAZED TO REPORT THAT I USED A MULTI-FUNCTION CALCULATOR FROM 'POUNDLAND'. YEH, BELIEVE

IT OR NOT, THIS BIT OF KIT, (FOR £1) HAS A 10 DIGIT DISPLAY, PROCESSES TRIG. FUNCTIONS, AND OPERATES IN OCTAL, BINARY, HEX AND DECIMAL. THEY WERE ALSO SELLING 8 PACKS OF 2-FINGER PLAIN CHOCOLATE 'KITKATS' TOO, WHICH, NEEDLESS TO SAY, I COULDN'T RESIST!

AS YOU CAN SEE FROM THE SMALL, 7 LINE PROGRAM BELOW, IT WAS SOON UP AND RUNNING. THE SCREEN HAPPILY REFRESHED JUST ABOUT AS QUICK AS I COULD TYPE AND THE BOYS SEEMED TO ENJOY THEMSELVES.

```
10 DEFINTA-Z:J=0
20 DATA 8448,15360,4352,15361,256,1023,-20243,201
30 FORZ=0TO7:READA:U(Z)=A:NEXT
40 A$=INKEY$:IFA$=""THEN40ELSEPRINT@0,A$
50 DEFUSR0=VARPTR(U(0))
60 J=USR0(0)
70 GOTO40
```

AFTER ALL THE EXCITEMENT HAD DIED DOWN, I GOT TO THINKING, WOULD THIS TURN INTO A 'GOOD-OLD' ONE-LINER? AND YES, AS YOU CAN SEE BELOW, WITH A COUPLE OF MINOR CHANGES, (REPLACING LINE 40 WITH A LOOP INSTEAD OF A GOTO), IT DID.

THEN I STARTED TO GET A BIT TOO CLEVER, AS MENTIONED EARLIER. I THOUGHT ABOUT USING A ROM CALL WITHIN THE ASSEMBLER PROGRAM TO REPLACE THE INKEY$ ROUTINE. NEEDLESS TO SAY, I GOT INTO A BIT OF A MESS BUT THANKS TO PETER STONE, I WAS PUT ON THE RIGHT TRACK AGAIN. (AS USUAL, I'D GOT A PUSH AND POP IN THE WRONG PLACE!).

THEN I REALISED I'D NEED TO GET AN EXIT ROUTINE INTO THE PROGRAM AS THE BREAK KEY WAS 'LOCKED OUT' AND A SYSTEM REBOOT WAS THE ONLY WAY TO QUIT THE PROGRAM. NOT GOOD PROGRAMMING PRACTICE EH?

AT THIS POINT, THE DATA STATEMENTS SEEMED TO BE GROWING EXPONENTIALLY, SO I DECIDED TO QUIT WHILE I WAS AHEAD SO TO SPEAK AND STICK TO THE ONE-LINER, AS ORIGINALLY PRODUCED. □UST PRESS THE BREAK KEY WHEN YOU'VE HAS ENOUGH.

```
10 DATA 8448,15360,4352,15361,256,1023,-20243,201:DEFINTA-Z:J=0:
FORZ=0TO7:READA:U(Z)=A:NEXT:FORX=0TO1STEP0:A$=INKEY$:IFA$=""THEN
NEXTELSEPRINT@0,A$:DEFUSR0=VARPTR(U(0)):J=USR0(0):NEXT
```

IN MAV'S WORKSHOP
by Ian Mavric

SETTING UP TRS-80 MODEL III AND 4 HARD DRIVE PARTITIONS

Ok so you've procured your hard drive from eBay, CraigsList, or one of the other myriad of places where they come up for sale from time to time. You've made new cables and following the steps outlined in the last issue[1], have a hard drive which powers up and it's ready to partition for use on your TRS-80 Model III or 4.

DOS AND HARD DISK DRIVER CHOICES

Just about every operating system for the TRS-80 can be made to work with a hard drive, this includes NEWDOS/80, DOSPLUS, TRSDOS 6, CP/M, and even MultiDos. But to my mind only two operating systems work properly with hard drives, having the least number of compatibility issues, and that is LDOS 5.3.1 for the Model III and LS-DOS 6.3.1 for the Model 4. That is not to say the other operating systems mentioned don't do the job, but I have used LDOS and LS-DOS for 20+ years with no issues, and to me it is the de-facto OS for hard disk users. I should also point out that when Tandy first released the TRS-80 5Mb hard drive in 1982, LDOS was Tandy's official OS (called HDOS at the time) so there is a certain lineage which is why it works so well. Just as NEWDOS/80 is probably the best (certainly the fastest) OS for a floppy based Model I or III, LDOS is the best OS for a hard disk TRS-80 even though its floppy operation is slightly slower than NEWDOS/80. Conversely NEWDOS/80 can be set up on a hard drive but its complicated and in my opinion clumsy to use.

The choice of hard disk driver software I have similar views, and the only one which consistently allows me to set up and use standard and modified TRS-80 hard drives is RSHARD5/6 by Misosys[2]. Again I've been using it for more than 20 years and it has never let me down. It's compact and bug free, and very few programs have shown problems running software off hard disk with RSHARD5/6 residing in memory. Radio Shack's drivers are suitable for use with hard drives with the standard 5Mb or 15Mb Tandon MFM drive installed but are somewhat limited beyond the capabilities of those drives. In other words if you modify your hard drive with a non-original MFM hard disk, you will need RSHARD5/6.

PARTITIONING THE HARD DRIVE

Partitioning simply means dividing up the storage space on your hard drive. Within that framework we have a number of limitations to work with and various ways to work around them. RSHARD5/6 allows you to customise partitioning so that just about any MFM hard drive up to about 70Mb can be used with the TRS-80.

Rule 1: LDOS and LS-DOS allow up to 8 disk volumes on a TRS-80 and each floppy drive is 1 volume.
So this means if you have a dual floppy drive Model III then can have up to 6 partitions on the hard drive. If you have four floppys on your Model III you can have up to 4 partitions on the hard drive.

Rule 2: RSHARD5/6 is limited to a maximum of 406 cylinders per partition.

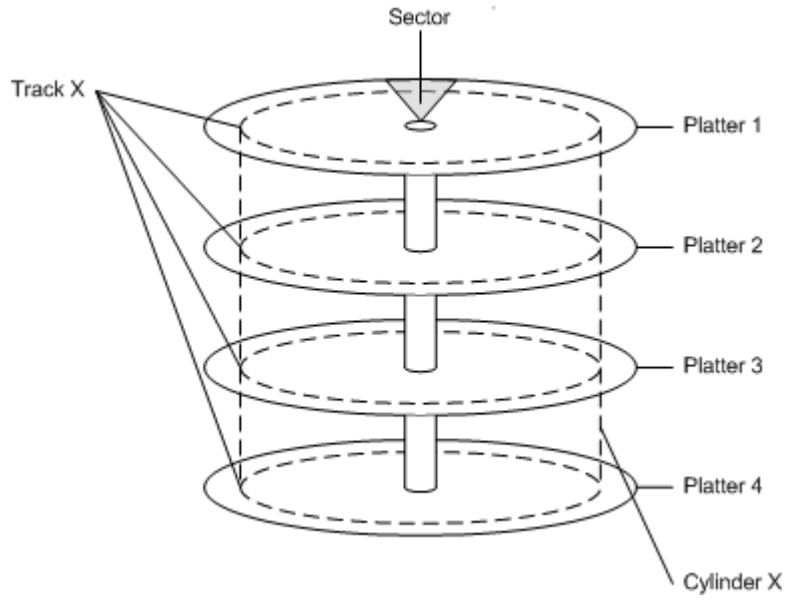
Most smaller hard drives have less than that, eg. Tandon TM602S (5Mb) has 153 cylinders, and Tandon TM503 (15Mb) has 306 cylinders. If you upgrade the MFM hard drive to, say, a Seagate ST-225 (20Mb) or ST-251 (42Mb) which have 615 and 820 cylinders respectively, you can see that the only way to make use of all the capacity of the hard drive is to partition it.

Rule 3: LDOS and LS-DOS limit the number of files per logical drive to 256.
Clearly even on a small 5Mb drive, if set up in a single volume, all 256 file slots will be filled before the all of the space is allocated. By making as many partitions as possible we maximise the space used on the hard drive, we can also divide the partitions into categories… say Partition 1 (drive :0) has the DOS, Partition 2 (drive :1) has Superscripsit and Visicalc, Partition 3 (drive :2) has games, Partition 4 (drive :3) has EDAS, PRO-DUCE etc., and drives :4, :5, :6, and :7 are floppy drives. While one can set up a hard drive with as little 1 partition, it's in our interest to have as many as possible.

USE THESE RULES TO DETERMINE HOW TO PARTITION YOUR HARD DRIVE

SMALLER DRIVES – PARTITION BY HEAD – example 1.
On smaller drives the simplest way to set up partitions is to allocate a head (and therefore all the tracks under that head) to that partition. So on a 5Mb TM602S, which has 4 heads, the 4 partitions each have one head reading 153 tracks. It's easy to remember your partitioning because a hard drive has a fixed number of tracks and heads. If you lose or ruin your HD boot floppy it's easy to deduce that on a drive which has 4 heads

and 153 tracks per head, each partition will be 1 head with 153 tracks. This is in fact how Radio Shack set up their hard drives from the factory and it works well for hard drives in the 5Mb, 10Mb, 12Mb and 15Mb range.
Notice I didn't say cylinders – cylinders represent all the tracks which carry the same number on each drive platter… see diagram:



LARGER DRIVES – PARTITION BY CYLINDER – example 2.
On a drive with more than 406 cylinders, even if you wanted to partition it by head, you are restricted by Rule 2, so unless you wish to give up a large proportion of hard drive space, partitioning by cylinder is the only way to achieve this. Say you have a ST-251 40Mb which has 820 tracks and 6 heads, if you specify 406 cylinders and one partition per head, yes you end up with 6 partitions, but lose 20Mb of disk capacity. The way to make full use of the disk capacity is to specify fewer cylinders and spread them across all 6 heads. So when RSHARD asks for how many cylinders for the partition, instead of replying 820 we would say 136 and when RSHARD asks how many heads for the partition instead of replying 1 we would say 6. That way you send up with your partitions spanning cylinders 0-135 for partition 1, cylinders 136-271 for partition 2 and so on. Once you decide on how your partitioning will be you need to record it, because if you lose or ruin the HD Boot disk it can be a real nightmare to remember where all the partitions start and finish.

THEN MANIPULATE YOUR HARD DRIVES

The driver is invoked via the command: SYSTEM (DRIVE=x,DISABLE,DRIVER="RSHARDn")<Enter> Where x is the

volume number and n is the version (5 for Model III LDOS 5.3.1 or 6 for Model 4 LS-DOS 6.3.1). See examples 1 and 2. Formatting the hard drive is accomplished by using the command: RSFORMn :x<Enter> Again n is the version and x is the volume number.
Once the first partition is formatted you can copy the DOS to partition 1 and make it the system partition thusly: BACKUP :0 :x (NEW,S,I,Q=N)<Enter>, and change the system drive so the hard drive is now drive zero: SYSTEM (SYSTEM=1) <Enter> .

WHAT HAPPENED TO MY FLOPPY DRIVES?

As you built your hard drive partitions and they started to take over volume (drive) numbers :0, :1, :2, and :3 the floppy drives were disabled as the numbers were given to hard drive volumes. This is normal. After you have set up all your hard drive partitions you need to bring your floppy drives back online with new numbers, and this is done using the command: SYSTEM (DRIVE=x,DRIVER="MOD3") or SYSTEM (DRIVE=x,DRIVER="FLOPPY"). The former being for Model III LDOS and the latter being for Model 4 LS-DOS.

COMPLETING THE SETUP

Those familiar with LDOS / LS-DOS will be familiar with the SYSGEN command - it saves your current configuration in a CONFIG/SYS file, so that next time you boot that floppy the previous configuration is restored. This is even more important for hard drives because as well as saving the hard and floppy drive configuration, it also transfers control of drive zero to the hard drive partition 1 upon boot. In Model III LDOS the command is: SYSTEM (SYSGEN, DRIVE=x)<Enter>, where x is the floppy drive you normally boot from (drive 0 on a floppy system and probably drive 4 or 6 on a hard disk system). The equivalent command on Model 4 LS-DOS is: SYSGEN (DRIVE=x)<Enter>.

UAL BOOT BOTH LDOS AND LS-DOS ON A SINGLE HARD DRIVE

As you play around with setting up a hard drive, it soon becomes apparent that you can enable and disable hard drive partitions in any order you like. An interesting side effect of this is that it doesn't take long to realise that you install two operating systems and have two boot disks, one for LDOS and one for LS-DOS. So you aren't then restricted to having your hard drive set up for Model III or 4 mode, you can boot Model 4 mode LS-DOS for serious work like Multiplan, Superscripsit and Profile 4 Plus, and boot LDOS to run your favourite Model III mode games. Here's how its done:
Let's say we have a standard Model 4D and a 5meg hard drive with 4 partitions.

* Starting in Model 4 mode we format all 4 partitions and install LS-DOS in partition 1.
* Disable partition 2 (that's where LDOS will go) so we have 3 partitions and 2 floppy drives, and SYSGEN your boot floppy - label it Model 4 HD Boot.
* Boot LDOS and bring online all 4 partitions and install LDOS on partition 2.
* Disable partition 1 (which is where LS-DOS is and not needed for Model III mode) so again we have 3 partitions and 2 floppy drives, and SYSTEM (SYSGEN,DRIVE=x) your boot floppy - label it Model 3 HD Boot.

What you have now is made the opposing DOS invisible to each operating system, so both can co-exist on the hard drive but not over-write or interfere with each other. A lot of commands in each DOS have the same names so if you were to accidentally issue FORMAT from LDOS while in LS-DOS mode the system will error out. Two of the hard drive partitions remain visible to both DOSes, and this works because both LDOS and LS-DOS share the same file structure. Check my video on YouTube showing a dual-boot hard drive[3].

EXAMPLE 1: Partitioning by HEAD

1. Power on Hard disk then the TRS-80 Model 4.

2. Boot Model 4 with LS-DOS 6.3.1 disk which has RSHARD6/DCT and RSFORM6/CMD on it.

3. Configure the Hard disk by entering the SYSTEM command:

```
SYSTEM (DRIVE=1,DISABLE,DRIVER="RSHARD6")
   Prompt as follows:
Enter drive address <1-2> and F/R [1;F] -----------> <enter>
Enter the STEP RATE <10us-7.5ms> [10] -------------> <enter>
Enter the physical TRACKS per surface [306] -------> 153
Enter the total number of HEADS [4] ---------------> 4
Enter partition's number of heads <1-4> -----------> 1
Enter partition's number of cylinders [153] -------> <enter>
```

4. Format the Hard disk using the TRSFORM6 command:

```
RSFORM6 :1
   Prompt as follows:
Disk pack name: HARDA
Password: PASSWORD
Manual Lockout: N
```

5. Repeat the process for remaining partitions:

```
SYSTEM (DRIVE=2,DISABLE,DRIVER="RSHARD6")
SYSTEM (DRIVE=3,DISABLE,DRIVER="RSHARD6")
SYSTEM (DRIVE=4,DISABLE,DRIVER="RSHARD6")
```

(answering 1 head and 153 cylinders each time)

6. Format the partitions you just created:

```
RSFORM6 :2
   Prompt as follows:
Disk pack name: HARDB
Password: PASSWORD
```

```
Manual Lockout: N

RSFORM6 :3
   Prompt as follows:
Disk pack name: HARDC
Password: PASSWORD
Manual Lockout: N

RSFORM6 :4
   Prompt as follows:
Disk pack name: HARDD
Password: PASSWORD
Manual Lockout: N

7. Relocate the Second Floppy Drive to :6 using the SYSTEM command:
SYSTEM (DRIVE=6,DRIVER="FLOPPY")
   Prompt as follows:
I/O ADDRESS: 2

8. Perform a disk backup to the Hard disk to copy the files over:
BACKUP :0 :1 (NEW,S,I,Q=N)

9. Identify the Hard Disk as the system disk
SYSTEM (SYSTEM=5)

10. Generate the configuration file on the boot floppy for the hard disk:
SYSGEN (DRIVE=5)
```

EXAMPLE 2: Partitioning by CYLINDER

```
1. Power on Hard disk then the TRS-80 Model 4.

2. Boot Model 4 with LS-DOS 6.3.1 disk which has RSHARD6/DCT and RSFORM6/
CMD on it.

3. Configure the hard disk by entering the SYSTEM command:
SYSTEM (DRIVE=1,DISABLE,DRIVER="RSHARD6")
   Prompt as follows:
Enter drive address <1-2> and F/R [1;F] -----------> <enter>
Enter the STEP RATE <10us-7.5ms> [10] -------------> <enter>
Enter the physical TRACKS per surface [306] -------> 820
Enter the total number of HEADS [4] ---------------> 6
Enter partition's number of heads <1-4> -----------> 6
Enter partition's number of cylinders [153] -------> 136

4. Format the Hard disk using the TRSFORM6 command:
RSFORM6 :1
   Prompt as follows:
Disk pack name: HARDA
Password: PASSWORD
Manual Lockout: N

5. Repeat the process for remaining partitions:
SYSTEM (DRIVE=2,DISABLE,DRIVER="RSHARD6")
SYSTEM (DRIVE=3,DISABLE,DRIVER="RSHARD6")
SYSTEM (DRIVE=4,DISABLE,DRIVER="RSHARD6")
SYSTEM (DRIVE=5,DISABLE,DRIVER="RSHARD6")
SYSTEM (DRIVE=6,DISABLE,DRIVER="RSHARD6")

 (each time specifying 6 heads and 136 cylinders)
```

6. Format the partitions you just created:

```
RSFORM6 :2
   Prompt as follows:
Disk pack name: HARDB
Password: PASSWORD
Manual Lockout: N

RSFORM6 :3
   Prompt as follows:
Disk pack name: HARDC
Password: PASSWORD
Manual Lockout: N

RSFORM6 :4
   Prompt as follows:
Disk pack name: HARDD
Password: PASSWORD
Manual Lockout: N

RSFORM6 :5
   Prompt as follows:
Disk pack name: HARDE
Password: PASSWORD
Manual Lockout: N

RSFORM6 :6
   Prompt as follows:
Disk pack name: HARDF
Password: PASSWORD
Manual Lockout: N
```

7. Relocate the Second Floppy Drive to :7 using the SYSTEM command:

```
SYSTEM (DRIVE=7,DRIVER="FLOPPY")
   Prompt as follows:
I/O ADDRESS: 2
```

8. Perform a disk backup to the Hard disk to copy the files over:

```
BACKUP :0 :1 (NEW,S,I,Q=N)
```

9. Identify the Hard Disk as the system disk

```
SYSTEM (SYSTEM=6)
```

10. Generate the configuration file on the boot floppy for the hard disk:

```
SYSGEN (DRIVE=6)
```

NEXT TIME I will discuss how to fix or upgrade a hard drive by replacing the original MFM with a different one and what to do with those 3 wires.

ianm@trs-80.com

REFERENCES:

1. " EXPANDED DISCUSSION ON TRS-80 MODEL III AND 4 HARD DRIVES", TRS8Bit December 2012 pp. 8-14.
2. http://tim-mann.org/trs80/doc/rshard.pdf
3. http://www.youtube.com/watch?v=0_JbYGis0a8

TRS8BIT



FLAMING JUNE ARRIVES TOGETHER WITH THE LATEST EDITION OF TRS8BIT! BOTH, I HOPE, PUTTING A SMILE ON YOUR FACES. WELCOME, TO WHAT IS TURNING OUT TO BE A MOST EXCITING YEAR!

FIRST I'D LIKE TO MENTION, WHAT I CONSIDER TO BE THE REALLY EXCITING NEWS, OF 2 BRILLIANT PIECES OF HARDWARE JUST DEVELOPED WHICH ARE AVAILABLE FROM MAV'S WEBSITE.
THE FIRST IS A NEW DOUBLE POWER SUPPLY, WHICH MAV AND DAVID COOPER HAVE DEVELOPED, FOR THE MODEL 1. NO MORE PANICKING WHEN THE 'OLD' POWER BRICK DIES. IT'S MADE TO SIT NEATLY IN THE EXPANSION INTERFACE USING THE SPACE TAKEN BY THE 2 OLD POWER SUPPLIES.
THE SECOND IS A 'VIRTUAL' HARD DISK FOR THE MODEL 3 OR THE MODEL 4. A NEAT PIECE OF ELECTRONIC ENGINEERING PLUGS INTO YOUR M3 OR M4 VIA THE 50WAY CONNECTOR AND COMBINED WITH AN SD CARD ACTS JUST AS AN ORIGINAL HARD DISK! THE 4 GUYS WHO DESERVE AN HONOURABLE MENTION FOR DEVELOPING THIS FANTASTIC BIT OF KIT ARE FRED VECOVEN, ANDREW QUINN, DAVID COOPER AND IAN MAVRIC (MAV).
FULL INFORMATION AND ORDERING DETAILS OF BOTH THESE BITS OF KIT ARE ON MAV'S WEB SITE.

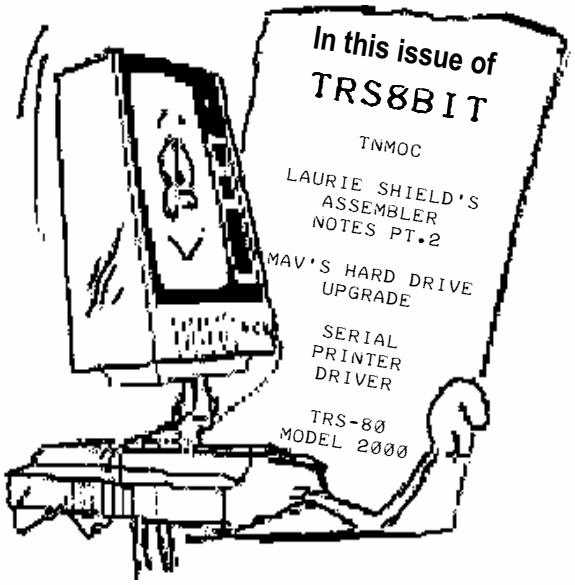
MATTHEW REED HAS RELEASED VARIOUS UPGRADES. HIS PC Z80 ASSEMBLER IS NOW ON V1.2. TRSTOOLS IS AT V1.09 AND TRSREAD/WRITE IS NOW AT 4.33

TWO NEW EMULATORS HAVE APPEARED. ONE FOR THE PC BY JIM EADIE AND ONE FOR THE ANDROID OPERATING SYSTEM BY ARNO PUDER.

I NOTICED THE OTHER DAY THAT THE CLUB100 ARE OFFERING A CONVERSION SERVICE FOR ANYONE WHO CAN NO LONGER READ THEIR TPDD AND TPDD2 3 ½ DISKS AND WOULD LIKE TO USE THEM ON THEIR NADSBOX. CHECK OUT THE CLUB100 WEBSITE FOR FURTHER DETAILS.

THE UK'S EBAY HAS SEEN A BIT MORE LIFE DURING THE LAST 3 MONTHS. A MODEL 3 WITH TWIN DISK FETCHED £63 AND A MODEL 1 WITH AN EXPANSION INTERFACE (NO DISKS OR MONITOR) FETCHED £73. THERE HAS BEEN A COUPLE OF MODEL 100'S FETCHING AROUND THE £50 MARK AND A MODEL 102 WHICH ONLY MADE JUST OVER £40!

JOHN PYE HAS HAD A MODEL 1 ADVERTISED ON THE WEBSITE FOR NEARLY A MONTH AND SO FAR THERE'S BEEN NO INTEREST SHOWN!



FOLLOWING ON FROM THE MARCH EDITION AND THE SAGA OF MY SOMEWHAT UNIMPRESSIVE ATTEMPT, TO UPGRADE MY MODEL 1 SYSTEM TO DISKS, I'M PLEASED TO REPORT THAT MY NEXT COURSE OF ACTION CHEERED ME UP NO-END!
I DECIDED TO SEND ALL MY OLD, PERSONAL, M1 DISKS, WHICH HAVE BEEN STORED AWAY FOR OVER 30 YEARS, TOGETHER WITH A FEW OTHERS, DONATED TO THE SITE, TO IRA FOR CONVERSION AND THEN, TAKE ADVANTAGE OF THEM VIA MATTHEW REED'S EMULATOR.
THERE WAS ONE SPECIFIC PROGRAM THAT I REALLY DID WANT TO HAVE AVAILABLE AGAIN, (WHICH FEATURES AS AN ARTICLE ON PAGE 27). SO I DULY PACKED THEM UP AND POPPED DOWN TO THE POST OFFICE. WHAT A SHOCK! POSTAGE WOULD BE OVER £120, NO GUARANTEED DELIVERY BUT SHOULD ARRIVE WITHIN 28 DAYS "IF ALL GOES WELL", NO TRACKING AND NO INSURANCE! (HOW MAY I HELP YOU SIR, EH!!)
WELL, BACK TO THE DRAWING BOARD OR RATHER GOOGLE, WHERE I CAME ACROSS THE 'PARCEL2GO.COM' WEBSITE. THE PACKAGE COULD BE COLLECTED FROM MY HOME, DELIVERY SHOULD TAKE 3 WORKING DAYS, FULLY TRACKED AND INSURED FOR FREE UP TO A VALUE OF £25. THE COST, JUST IN CASE YOU WONDERED, WAS JUST UNDER £34.
SO, I DULY PAID USING A CARD AND ON THE AGREED DATE, A FRIDAY, THE MAN AND VAN ARRIVED, PICKED UP THE PACKAGE AND OFF HE WENT INTO THE SUNSET SO TO SPEAK! NOW HERE'S THE REMARKABLE BIT, MONDAY EVENING (UK TIME), I HAD AN EMAIL FROM IRA TO SAY THAT THE PACKAGE HAD ARRIVED AND TUESDAY EVENING (UK TIME) I WAS BROWSING AN EMAIL FROM IRA CONTAINING A 'ZIPPED' FILE OF MY 70-ODD 5 ¼ DISKS.
WHAT BRILLIANT SERVICE ALL ROUND EH! ONCE AGAIN, THANKS IRA, FOR SUCH DEDICATED SERVICE WHICH WAS TRULY WELL ABOVE AND BEYOND NORMAL EXPECTATION.
SO THE MORAL OF THIS LITTLE TALE IS; IF YOU'VE A STASH OF OLD 5 ¼ DISK WITH NOTHING TO READ THEM ON, OR WHICH YOU WOULD LIKE TO USE WITH AN EMULATOR, SEND THEM OFF TO IRA FOR CONVERSION BEFORE IT'S TOO LATE AND THEY BECOME UNREADABLE.

I HOPE YOU'VE NOTICED I'VE RECENTLY PLACED A 'CLUSTRMAP' ON THE WEBSITE. I NOTICED THE FIRST ONE ON MIGUEL'S SITE AND THOUGH IT WAS A BRILLIANT IDEA. I THOUGHT IT WOULD BE INTERESTING TO KNOW WHAT, IF ANY, WORLD-WIDE INTEREST THERE IS IN THE U.K. SITE. IT TOOK ME TOTALLY BY SURPRISE! I THINK JUST ABOUT EVERY CONTINENT IS REPRESENTED. A BIG WARM WELCOME TO YOU ALL, I'M SO DELIGHTED THAT THE 'OLD TANDYS' STILL ATTRACT SO MUCH INTEREST.

SO; DOWN TO THIS ISSUE.
THERE'S A QUICK NOTE ABOUT THE 'NATIONAL MUSEUM OF COMPUTING' WHICH I'M HOPING TO VISIT SHORTLY.
LAURIE SHIELD'S ASSEMBLER NOTES MOVES ON TO PART TWO.
MAV COMPLETES HIS TRILOGY ON THE TRS-80 HARD DRIVES.
THERE'S A SERIAL DRIVER FOR ALL US 4K OR 16K USERS.
MAV WONDERS IF THE MODEL 2000 WAS THE BEST TRS-80 EVER AND FINALLY, WOULD YOUR BELIEVE IT?

TAKE CARE
EVERYONE

DUSTY



HI DUSTY,

I THOUGHT YOU MIGHT BE INTERESTED TO HEAR ABOUT THE 'NATIONAL MUSEUM OF COMPUTING' WHICH IS IN THE SAME GROUNDS AS BLETCHLEY PARK.

I WAS LUCKY ENOUGH TO GO ON A GUIDED TOUR OF BOTH BLETCHLEY PARK AND THE MUSEUM QUITE RECENTLY AND FOUND THE COMPUTER DISPLAYS IN THE MUSEUM MOST FASCINATING.

THE ONLY DISAPPOINTMENT WAS THE FACT THAT I COULDN'T SEE A 'TANDY' ANYWHERE. NEITHER WAS THERE ANY MENTION OF TRS8BIT TOO!

I DON'T THINK THEY ARE AWARE OF ANYONE USING EARLY MICRO COMPUTERS OTHER THAN DISPLAYING THEM AS 'OLD TECHNOLOGY'

PERHAPS IF YOU EMAILED THEM THE LATEST COPY OF TRS8BIT, IT WOULD PUT OUR NAME ON THE MAP, SO TO SPEAK.

ONE THING WHICH DID CATCH MY EYE, WAS AN ICL 2960 SERIES COMPUTER WITH EDSK60 DISK DRIVES. 60 MEG. AND AS BIG AS A DOUBLE FRIDGE/FREEZER! (REQUIRING MUCH MORE ELECTRICAL POWER THOUGH). THEY TOLD ME THEY HOPE TO GET IT RUNNING IN THE NOT TOO DISTANT FUTURE.

I'VE ENCLOSED A PHOTO OF THEIR ICL 2900 SETUP, JUST TO REMIND YOU OF THOSE 'HAPPY TIMES' WE HAD RUNNING 'RJE' AND 'GEORGE 3'.

BEST WISHES.

E.T. FONEHUME.

THANKS FOR THE EMAIL E.T. THE PHOTO TAKES ME BACK A WHILE! I'VE MANY HAPPY MEMORIES OF THAT TIME.
I CAN STILL REMEMBER THE WONDERFUL MEALS IN THE STAFF CANTEEN AND YOU HAVING TO LOAD ALL THOSE 'GEORGE CHAPTERS' WITH THEIR CHECK DIGITS, EVERY TIME AN UPGRADE WAS REQUIRED!

I'VE PUT A LINK TO THE TNMOC ON THE WEBSITE.



THE NATIONAL MUSEUM OF COMPUTING - WWW.TNMOC.ORG

THE NEW POWER SUPPLY FROM:
THE RIGHT STUFF



LOST ONE OR BOTH OF YOUR MODEL I POWER SUPPLIES? ARE YOURS WORN OUT OR JUST PLAIN BUSTED? DID YOU JUST BUY OR INHERIT A MODEL I AND IT DIDN'T COME WITH THEM?

BUY THIS ONE - IT POWERS YOUR MODEL I SYSTEM!

* TAKES ONLY 1 POWER POINT *
* LIVES IN THE EXPANSION INTERFACE *
* AND POWERS BOTH THE KEYBOARD AND E.I. *

ONLY $A145.00 + SHIPPING

AVAILABLE FROM:
HTTP://IANMAV.CUSTOMER.NETSPACE.NET.AU/TRS80/

ASSEMBLER NOTES Part 2
Laurie Shields

Chapter 4

Assembler Programming

Like all good stories we must have a beginning, an end and something in the middle. It's the middle part that's rather awkward so let’s dispose of the easy bits first.

First Simple Rule:

All assembly language programs must finish with the simple statement END. This tells the assembler (more about him later) that it has come to the end and nothing else is of any consequence and can be ignored. Without an END to your program you will get an error message EOF. Of course if you are using the dreaded E....m you will know that just as it was being put on the shelf for sale somebody spotted there was a bit missing and it had to be added on after the END so the END wasn't the end after all and it set the scene for all the other incomplete bits of software from 2.0 to the M....3 Dos.

Second Simple Rule

All assembly language programs need a beginning which tells the assembler (he's here again) the position, (location or address) in memory for which the machine code program is written. This beginning or origin statement takes the form:
ORG value
Whence 'value' can be any decimal, hexadecimal number in the range 0 0FFFFH. (It could also be a label with an associated numeric value known to the assembler).

This is the big difference in thinking between Basic and machine code.

Basic programs are completely position independent and to a large extent machine or even type of computer independent. Machine code is written not just for a particular central processor but also very likely for a particular configuration of computer and further more to occupy and work in only one part of that computers memory.
This often raises the question "Why bother ?" but we won't go into that just yet.

The value used for the ORG of your program depends on the

purpose of the program. If you were re-writing the whole of the Level 2 Rom you would start with an ORG 0, not caring whether it was Hexadecimal or ordinary numbers and then grow old gracefully. Most of the time however we want to create a program to work in a section of unused memory and a convenient place is the upper part of the 16K Ram as this is not used by the Aculab, cassette or any of the Disc operating systems. All our programs therefore will be based on an ORG of 7800Hex but of course you could change it if you wanted to.

Our assembly language source code (notice the crafty way these new words creep in) now looks like this:

```
ORG 7800H
?
?
?
END
```

and as some folk might say: "It certainly looks impressive but what does it do ?".
There's a definitive answer to that question, that goes back to time immemorial or at least to Babbage, and it is: "Well it doesn't do anything yet ... but wait till I get the bugs out."

Third Simple Rule

Assembly language programming isn't a bit like BASIC. Don't cry about it - there's nothing anybody can do - and remember you couldn't understand ON ERROR GOTO and PRINT USING once.

Sooner or later we are going to want to change a Hex number into decimal so lets do it with 7800H:

```
7000H = 7         4096 =    28672
 800H =   8        256 =     2048
  00H =      0      16 =        0
   0H =         0    1 =        0

7 8 0 0 Hex =   30720 Decimal
```

Now that wasn't too hard was it ? I leave you to work out for yourself how to do get back from decimal to hex.
There's a tradition build up with the TRS80 that the first machine code program example should be whiting out the screen in no-time-at-all, so just to be that bit different we will fill the video with a chequer-board pattern. In order to achieve this laudable objective we need to know how the video is connected to the computer ... AND ... I don't mean that silly little plug on the end of a bit of wire.

Fourth simple rule

Most apparently straight forward bits of English don't mean, what they used to mean when talking about computers.
What we mean is the logical connection between the Z80 and the video character generator. Fortunately the Level 2 manual comes to our aid and informs us that the video is treated as if it were a chunk of ordinary RAM of 1024 bytes starting at 3000 Hex (or 15360 Dec) and finishing at 3FFF Hex (16383 Dec).

So our object is to write in assembler the nearest equivalent of the following BASIC program:

```
10 FOR X = 15360 TO 16383: PORE X,161: NEXT
```

Now just hold on there a little! That's far too complicated for an assembler programmer to understand. Can't we simplify it a little first ?
Well we could try:

```
10 FOR X = 15360 TO 16383
20 PORE X ,161
30 NEXT X
40 END
```

That's getting better but I'm afraid all this FOR / NEXT business is just a little too .. how shall we say .. intricate .. can't we get there in singles.
Third time lucky ?

```
10 REM Simple BASIC
20 LET HL = 15360
30 LET BC = 1024
40 POKE HL,161
50 HL = HL + 1
60 BC = BC - 1
70 IF BC <> 0 THEN GOTO 40
80 END
```

For those who didn't skip the earlier chapters, the letters used for the variables might seem familiar; they are of course the same as those used for two of the Z80 registers.

Before we can go much further there is one important concept. that we must master and that is the complete irrelevance of line numbers in assembler programming. Whereas in BASIC we have the ability to refer to line numbers in the program to change the flow of control by GOTO's and GOSUB's, in assembler we must identify the particular line with a label and then refer to that label.
Labels or symbols can be called anything you like, they can be as long as you like provided they begin with a letter and aren't one of the assembler's reserved words like END or LOAD etc.

In our BASIC program that we are going to copy the program keeps jumping back round in a loop until BC is exhausted so let us use the label LOOP. To convey to the assembler that a particular collection of letters is to be referred to later we must add a colon immediately after the label as a means of declaring it to the assembler.

First GOLDEN rule of assembler programming.

You will never learn ANYTHING by reading until you have actually keyed it into the machine yourself.

So stop reading, put the cat out, switch on the computer, get your copy of Zen, and System load, @Run or Zen/cmd now ! Make sure you have a clean cassette, formatted floppy tape or disc with some room on it as we shall save the source file for future use. From here on in <E> means the Enter (TRS80) or New Line (Video Genie) key.

To switch Zen into the text entry mode key: E <E> (Enter the Editor). Zen responds in a similar fashion to the AUTO mode in BASIC by printing the number 1 for the first line and waits for you to key in anything you like, so key in the following except for the line numbers, which Zen does for you.

```
1 ORG 7800H <E>
2 LD HL, 15360 <E>      (let HL = 15360)
3 LD BC, 1024 <E>       (let BC = 1024)
4 LOOP: LD (HL),161 <E> (poke HL,161)
5 INC HL <E>             (HL = HL+1)
6 DEC BC <E>             (BC = BC-1)
7 LD A, H <E>             (if
8 CP 0 <E>                  BC
9 JP NZ,LOOP <E>               <
10 LD A,C <E>                    >
11 CP 0 <E>                        0
12 JP NZ,LOOP <E>                    GOTO)
13 END <E>
14 . <E>
```

The full stop tells Zen that we have come to the end and there's no more text input. Zen ignores the full stop and switches control back to the command level.

Let's step through the program nice and gently to see what we've done, forgetting all about ORGS and ENDS for the time being.
2 The LD instruction loads the HL register pair with the value 15360 which is the address of the top left hand corner of the video. The LD part of the instruction is known as the OPCODE and the 15360 part is the OPERAND.

3 Likewise BC becomes 1024, which is our Byte Count.

4 This is the tricky line. LOOP: is a label LOOP because it has a ':' after it. LD we know means load but what does (HL) imply. Well its beautifully simple (it needs to be) once you get the hang of it. HL by itself means the register pair HL. (HL) means the byte in memory that HL is looking at, so for the first time round HL=15360 and the load instruction means that particular byte, next time round it will be 15361 etc. 161 is simply the value of the graphics character that is to be loaded into the byte pointed to by HL.

5 & 6 Add one (INCrement) to HL and subtract one (DECrement) from BC. Since the Flag registers generally go on strike when we are doing 16 bit arithmetic the Z flag will not be triggered when BC becomes zero, so the only way to find out if the whole of the 1024 in BC has gone is to check B and C separately and if either of them are not zero then carry on looping.

7 Load the clever A reqister with the value held by B.

8 ComPare means for the Z80 'test for the difference between the A register and' in this case 0. If in testing the Z80 finds no difference then the Z flaq will be set.

9 Is it, because if not there's work still to be done.

10 Check the value in C the same way.
11 Compare.
12 Keep up the good work.
13 Everybody happy ?

Fifth simple rule

Condition testing causes most of the headaches in assembler.

At the end of our program we would have filled the video completely and the Z80 world then execute the next machine code instruction which, if we do nothing about it would be just whatever the next byte in Ram just happened to be. The fact that we have put an END in our code does not mean that the computer will respond with >READY the way it would in Basic, when control is passed back to the keyboard.

In machine code the only way to stop the Z80 chip is to switch the computer off, this being rather a drastic measure. We can however hand control back to Zen provided we arrange it before we go blasting the video with graphics. How to do it you will find out later.

Meanwhile the video is displaying Z>, which is Zen's way of saying 'Master I am here to do your will - What is your command ?".
We must now understand that Zen has got a singularly one track mind when it comes to manipulating source code. If we want Zen to display, print or alter any line we have to tell Zen in simple language whether to go to the Top line, or Up so many, or Down so many lines. To review our text which conveniently, is less than one video page enter the following:

T <E> (This tells Zen to point to the Top line)
<E> (Enter by itself tells Zen to clear the Video)
P13 <E> (Print thirteen lines)

At this point lets assume that we have an error in line 4, say there's a semicolon ';' instead of a colon ':' How do we fix it ? First get Zen's mind concentrating on line 4 by keying U9 <E> (Up nine lines) and line four is displayed. Now key N <E> (for New) and depending on what version you have, either line four will be zapped and you can key it in afresh or it will be displayed with the cursor at the end so you can backspace to the error and retype the line from the error onwards. Our program should now look like this:

```
1 ORG 7800H
2 LD HL,15360
3 LD BC,1024
4 LOOP:LD(HL),161
5 INC HL
6 DEC BC
7 LD A,B
8 CP 0
9 JP NZ,LOOP
10 LD A,C
11 CP 0
12 JP NZ,LOOP
13 END
```

To turn these Mnemonics into actual machine code bytes we activate the assembler in Zen by keying A <E>. Disc users will be prompted by Zen for Source File> to which the answer is simply <E>, unless they want to assemble direct from disc and then they wouldn't be reading this chapter.
Then Zen asks OPTION.
Since you might want to just check for errors, or display the assembled code on the video, or write it to cassette as a System tape etc, this is the stage at which you tell Zen where to send the output. Since we want it on the Video, key V <E> and you will get the complete program assembled as shown below except for the headings, or an error message such as OPERAND or HUH ? where Zen can't decipher your typos.

```
Line Memory Object Label Opcode Operand
  # Address Code
  1                        ORG    7800H
  2 7800 21003C             LD     HL,15360
  3 7803 010004             LD     BC ,1024
  4 7806 36A1      LOOP:    LD    (HL),161
  5 7808 23                 INC    HL
  6 7809 OB                 DEC    BC
  7 780A 78                 LD     A,B
  8 780B FE00               CP     0
  9 7800 C20678             JP     NZ,LOOP
 10 7810 79                 LD     A,C
 11 7811 FE00               CP     0
 12 7813 C20678             JP     NZ,LOOP
 13                         END
```

Note that Zen has understood the decimal numbers but in generating the machine code program these have all been changed into hexadecimal.

Now somebody at the back is bound to ask whether it really takes as much effort to find out if something has been done (lines 7-12) as it does to do it (lines 2-6). In all honesty the answer must be 'not quite' but we don't want to run before we know which direction we should be walking.

Lets prove that the program actually works before we go any further. To do that with E.... m or M....80 or very nearly any other assembler the technique is to assemble to cassette or disc, go back to the operating system, (system) load the object code, power up Debug or Supermon and jump to your program start address. Now that is just too much like HARD WORK so we will get Zen to do it all for us.

First we need to tell Zen that the object code bytes should be loaded into memory as they are evaluated during assembly. To do this we need to insert one line of code into our source file between line 1 and 2. Here's how: Goto line 2 (T <E> for Top, then D1 <E> for Down 1) Zen should be displaying line 2. Key E <E> to enter the editor and we get the prompt 2. Key LOAD 7800H <E> followed by the full stop . <E>. Assemble the program again, using if you have a printer the E option, so that you can frame the printout for posterity as the your first working machine code program. The result of the assembly should look exactly as before except the line 2 to 13 now have the numbers 3 to 14 and there's the new line 2 'LOAD 7800H'. To check that the code is in memory use Zen's Query command

Q7800H <E>. You should get:

```
7800    21 00 3C 01 00 04 36 Al  !.<...6.    !.<...6!
7808    23 0B 78 FE 00 C2 06 78  4.x....x   i.x..Q.x
7810    79 FE 00 C2 06 78 xx xx  y.... X..  y..Q.x..
```

The xx figures aren't important, except that if we don't intercept the Z80 after its done our program then they are what will happen next. The Ascii representation on the right will be different as Scripsit can't print the graphics that you get on your video.

To execute the program we want to instruct Zen to Jump or Goto the address 7800 Hex and let the program do its thing and when address 7816 Hex is reached to return to Zen. There's a minor complication here as very early versions of Zen used the single command letter G whence you were then prompted for the address, later versions included the address with the G (Goto) like this G7800H <E> and finally the versions of Zen with the Global replace facility use the command format J7800H <E> (Jump). Since you can all refer to your manuals I don't expect that this will be too problematical.

Depending on your version, key one of the following: J7800H <E>, or G7800H <E>, or G <E> followed by 7800H <E>.
Zen will then ask for a Breakpoint, key 7816H <E>, and faster than the eye can blink your video will be filled with spots.

Unfortunately if you didn't buy your Zen from me, the screen will, even more quickly, be cleared so that you might not catch it; but you know the answer to that.
Before we dash off and celebrate lets write the source code to tape or disc. Prepare your magnetic bulk storage device for recording and make sure the cassette, floppy tape or disc has some space on it and key W <E>. Zen will then ask you for a name or filespec as appropriate and then record your source file. Cassette and floppy tape users have the V command to verify that the recording is a good one and does not generate any checksum errors.

Sixth simple rule.

Assembler is so fast that five minutes of programming will take at least five hours to get right. This is known as the advantage in speed of Assembler over Basic.



Genie III...

your third and greatest wish!

The new Genie III surpasses even its predecessors by harnessing all the benefits of modern, professional microcomputing in one unit.

Versatility.

Software availability is exceptionally good – in fact the Genie III has more software available than any of its competitors. And even if you don't want to use a ready made business package, with the Genie III you can write your own! The Genie III is fully operational in either CP/M, or level II basic. It has a built-in screen, 64K RAM, and dual disk drive with 730K storage capacity, all incorporated in its compact and attractive casing.

Speed.

The new Genie III is an obedient slave in the office, quickly and quietly taking care of all your routine work, from printed invoices, mailing lists and letters to stock control.

Economy and simplicity.

It all costs less than a new office junior, yet it's simple enough to operate, and comes complete with a teaching programme to show you how.

Reliability.

Your Genie III is reliable too, manufactured to the most stringent standards with top quality components. And even in the unlikely event that your Genie should ever temporarily fail to please you are assured of very fast, competent service because the Genie III is only available from our specially selected SUPERDEALERS. Dealers whose technical expertise and experience in the fields of installation, software availability and repair is proven.

Software.

Finally, it's compatibility with CP/M or Level II BASIC means there is more Software available for Genie III than any other comparable system!

Please send me FREE, 16 page colour brochure on the Genie Computer System. I enclose 25p postage.

Name _____

Address _____

_____

Telephone _____ CT·82

LOWE electronics

Chesterfield Road,
Matlock,
Derbyshire DE4 5LE.
Telephone: 0629 4995.
Telex: 377482 Lowlec G.

Chapter 5

Now that it works lets change it

But first we will review what we have learnt.

1. The registers A, B, C, H, L, etc. or the register pairs HL, BC, etc. can be loaded with a value using the LD instruction (LD BC,1024).
2. The contents of the registers can be incremented or decremented (INC HL).

3. Data can be transferred between the registers, again using the LD Instruction (LD A,B)

4. The contents of the A register can be compared with some other value (CP 8), and if they are the same the Zero flag is set (1), otherwise it is reset (0).

5. Instead of executing the next instruction the program can jump to a different location that has been identified with a label.

The next few pages are going to be the hardest to grasp for some time, so make sure you've got some peace and quiet before you start; and until you've grasped OR, XOR and AND you'll never make the grade.

These three very powerful instructions do Bit by Bit checks and comparisons between the A register and some other data. The other data can be virtually anything from a fixed numeric value, the contents of any register (including the A itself) or the contents of a byte in memory pointed to by either the HL or one of the index registers (IX or IY)

Let us assume that the A register contains 49 decimal and we will see what happens if we carry out OR, XOR and AND with 26 decimal. Now 49 Dec = 31 Hex and 26 Dec = 1A Hex
As the instructions are going to operate at Bit level we had better convert to binary 0's and 1's.

```
49 Dec = 31 Hex =   0 0 1 1 0 0 0 1
26 Dec = 1A Hex =   0 0 0 1 1 0 1 0
```

With all three instructions we ignore the 0's and fix our minds on the 1's.

The OR instruction means take the 1's from either the first value OR the second OR both and put them in the A register.

```
So we get   0 0 1 1 0 0 0 1 = 31H
        OR  0 0 0 1 1 0 1 0 = 1AH
            ---------------
            0 0 1 1 1 0 1 1 which is 3B Hex.
```

Its something like the 'lowest common multiple'. In that the result is bigger than either of the two starting numbers and it has got 'set' bits common to both.

The XOR instruction, abbreviated from EXCLUSIVE OR, means take the 1's from either the first value OR the second value but EXCLUDE those where both values have a 1, and put the result in the A register.

```
So we get   0 0 1 1 0 0 0 1 = 31H
        XOR 0 0 0 1 1 0 1 0 = 1AH
            ---------------
            0 0 1 0 1 0 1 1 which is 2B Hex
                  ^
                  this bit was excluded.
```

This operation can be considered most like a two-way switch. Any bits that are set in the second number will ensure that the result for those particular bits will be the dead opposite of whatever they were with the first number and visa-versa. The most commonly seen example is blinking 'star' on the cassette load.

The AND instruction is simpler and means take the 1's where they occur in both the first valve AND the second value with the result into the A register.

```
So we get   0 0 1 1 0 0 0 1  = 31H
        AND 0 0 0 1 1 0 1 0  = 1AH
            ---------------
            0 0 0 1 0 0 0 0   which is 10H
                  ^
                  this is the only bit set in 31H AND 1AH
```

This operation is very akin to a filter. Any bits set in one of the numbers that don't match those set in the 'filter' are not set in the result.

Now part of the importance of these operations lies in the fact that the Zero flag in the F register is triggered to set or reset depending on whether the resultant value in the A register is zero or not. Also as none of the three instructions can generate a carry condition the C flag is automatically reset irrespective of the final value.

Back to our program. We left it with a lot of coding just to check if the BC register pair had reduced to zero. Consider now what would be the result of doing an OR operation between the B and C registers. If either of the registers had just one bit not equal to zero the final result would reflect this condition and it would not be zero. So we now have a means of checking BC in one go .... or have we? Well we nearly have; remember one of the values for the OR command needs to be in the A register first, so it looks like this:

```
7 LD A,B    (load the contents of the B register into A)
8 OR C      (test if either of them is not zero)
9 JP NZ,LOOP
10 END
```

You have now been introduced to the true assembler's abiding passion - trying to save the odd byte here and there. In general the main objective of Basic programmers is to get their creations working without any error messages. Until the OM appears very few ever worry about how much memory is being used.

With true machine code it is very different. Whilst the program is being assembled the hard facts are there on the screen and every byte that is used is open to criticism. There are of course the people who use macros, relocatable object modules and linking loaders who neither know nor care about such niceties, but we will have none of their ways of working here.

The designers of the Z8O haven't finished with us yet though. Being true men of spirit, they have provided yet another means of saving one byte out of our program. The instruction JP NZ,LOOP takes up a hulking three whole bytes just to tell the Z80 to jump back six or so steps. We have available a two byte instruction that will do just that. It uses the OPCODE JR (Jump Relative) and is used in exactly the same way. Since in two bytes we cannot include all the details for an instruction to jump and the address of any of the 65536 addressable memory locations the use of this command is limited to jumps of plus 127 or minus 128 bytes.

Now we have a plan so power up Zen and read in the old source file by keying R <E> and answering the prompts for a filename. Since there are only a few lines to change at the end of the program the simplest way is to zap the lines we don't need and key in the new ones. Get Zen pointing to line 8 CP 0, which is the first of the ones we don't want and key Z5 <E> (zap five lines). This should leave you with just
END for line 8. Press E <E> to enter the editor and key in the following lines:

```
8 OR C <E>
9 JR NZ,LOOP <E>
10 . <E>
```

Again, note the use of the fullstop to terminate text entry.
We have now got it into the most compact form so just to make life difficult we are going to incorporate this extremely useful routine (?) into a USR function from Basic.

There are, as you might have guessed just a couple of problems.
 i) How do we link our program to Basic ?
ii) How do we ensure our program will not upset Basic ?
Lets tackle the second one first. When Basic hands control over to a USR routine it always ensures that there is nothing important in the AF or HL registers, so we can do what we like with those. As for the BC and other registers we do not know whether what they contain is of any consequence or not so we had better save the contents of BC before we use it and restore whatever was there when we are finished.

Fortunately the Z80 has a very useful technique of saving and restoring the contents of a register pair. It is known as the 'stack' and works a bit like a spring loaded money-box allowing you to push values into storage and pop them back again afterwards. The stack is also used by the Z80 itself for remembering return addresses from subroutines. Most machine code programmers at one stage or another get their pushes and pops crossed with their calls and returns generating something that resembles more of an infestation than a bug, but more of that later. Sufficient to say that with our program all we need to know and do is PUSH BC at the start and POP BC at the end.

But what happens at the end ? How do we get back to Basic ? Well the way Basic processes a USR statement is to call it like a subroutine. A subroutine in machine code works just like one in Basic and needs at least one happy return, abbreviated to RET.

Make the necessary changes to your source file so that after assembling it looks like this:

```
1                       ORG     7800H
2                       LOAD    7800H
3 7800                  C5      PUSH BC
4 7801 21003C           LD      HL,15360
5 7804 010004           LD      BC,1024
6 7807 36A1   LOOP:     LD      (HL),161
7 7809 23               INC     HL
8 780A 0B               DEC     BC
9 7808 78               LD      A,B
10 780C B1              OR      C
11 780D 20F8            JR      NZ,LOOP
12 780F C1              POP     BC
13 7810 C9              RET
14                      END
```

If you haven't got a printer write down the object code bytes: C5, 21, 00, 3C etc. as we are now going to change the ORG to another value, say 8700H and re-assemble. Comparing the object code generated for the two addresses should show no difference.
This is **VERY IMPORTANT**. It means that our program, unlike most m/c ones, will function correctly anywhere in memory. We are not tied down to a particular block of Ram that might be needed for something else.

Now as for linking it with Basic we have a number of different techniques available but all of them require us to know the individual values of our code in decimal rather than hex. If you are using E.... m, M....80 or one of the other very powerful assemblers then you will have no doubt large reference tables in the manuals to help with the conversion process. That's not how we do it with Zen however, simply get back to Basic and the TRS80 will do it for you.

To get to Basic from cassette or Aculab do a Jump or Goto memory address 114, from disc Zen return to Dos first and then Basic. Having got to >Ready prompt, key the following to extract from memory the decimal values of the program that Zen loaded into memory on the last assembling operation:

```
AD = 30720 :FOR X = 0 TO 16 :PRINT PEEK(AD + X); :NEXT
```
and you should get:

```
197 33 0 60 1 0 4 54 161 35 11 120 177 32 248 193 201
```

This is obviously some very important data so lets turn it into Data statements:

```
60 DATA 197,33,0,60,1,0,4,54,161
70 DATA 35,11,120,177,32,248,193,201
```

If you haven't forgotten we discovered earlier that Basic stores integers in memory with each one being allocated two bytes. Well since we have seventeen bytes to put somewhere lets reserve some memory for 9 integers by setting up an array US%(8) - US for USR, % makes it integers and subscripts 0 to 8 give nine values i.e. eighteen bytes of Ram, so:

```
10 DIM US%(8)
```

OK we've got 18 bytes - where is it? Well the problem is that it is likely to move about depending on whether or not any other simple variables come asking Basic for storage space. But wherever it is, Basic will always give us its position via VARPTR, and simply asking for VARPTR(US%(0)) will point to the first byte of the 18.

To put the decimal values back into memory we simply execute, with Pokes, something similar to the Peeks that got them out.

```
10 DIM US%(8)
20 FOR X = 0 TO 16 30 READ C
40 POKE (VARPTR( US%(0) + X), C
50 NEXT X
60 DATA 197,33,0,60,1,0,4,54,161
70 DATA 35,11,120,177,32,248,193,201
80 rest of program
```

There's a 1 in 5000 chance that this routine would fail should the result of (VARPTR(US%(0) +X) be greater than 32767 and what we ought to do is test it first before executing the Poke, but I leave that to you.

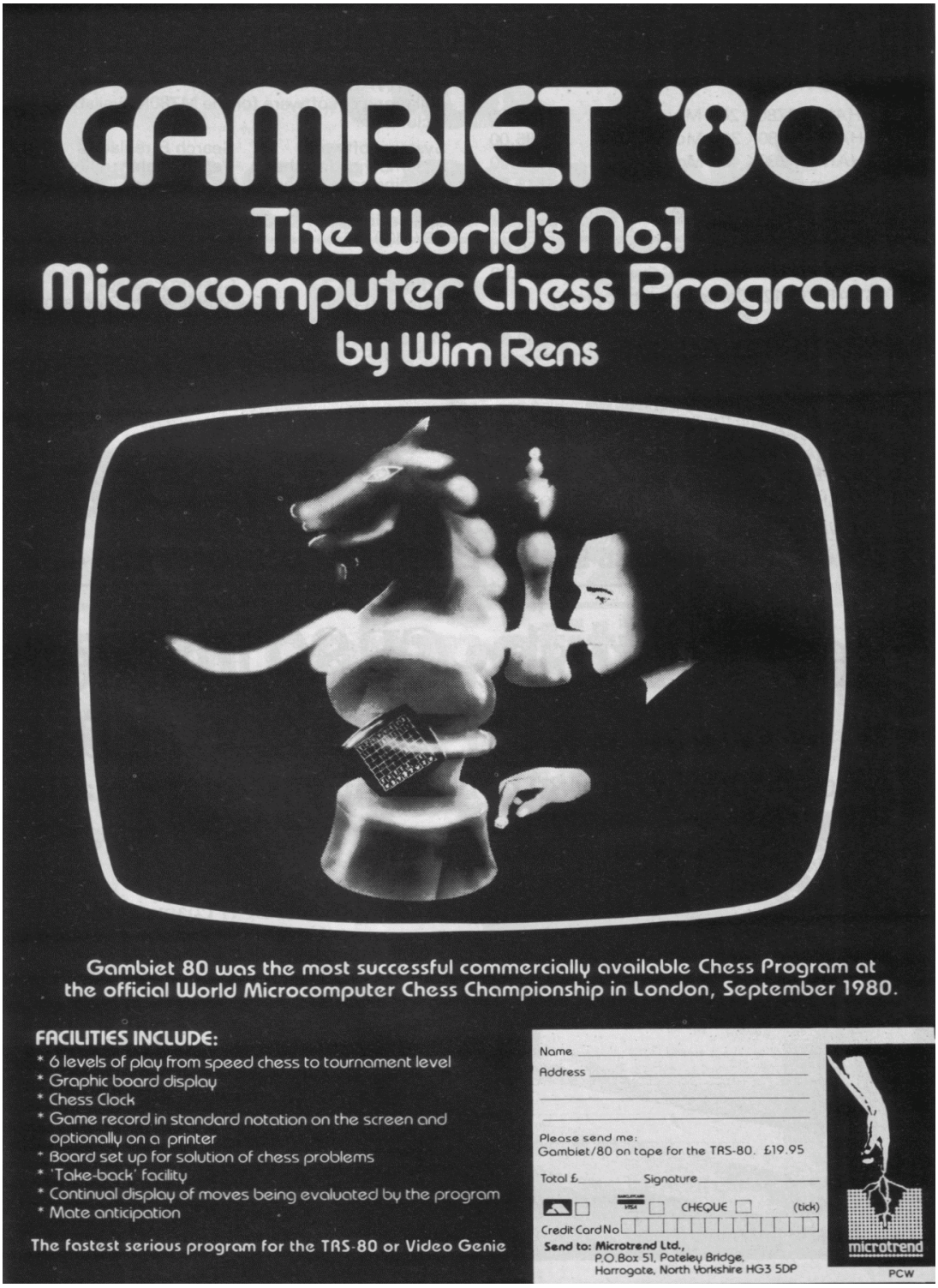
When line 80 is reached the whole of our machine code program is safely in memory and we now set up a subroutine at say line 1000 which can be called at any stage in the main Basic program. Disc users have it easy in setting up USR routines and their subroutine would be:

```
1000 DEFUSR = VARPTR( US%(O))
1010 X = USR(0):RETURN
```

Level 2 programmers have a bit more work to do, which involves, as you might have guessed, dividing the VARPTR value by 256 and Poking the result and the remainder in the USR vectors given in the manual:

```
1000 X = INT( VARPTR( US%(0))/256)
1002 PORE 16527, X
1004 POKE 16526e VARPTR( US%(0)) - 256 - X
1010 X = USR(O):RETURN
```

It is necessary for us to set up the USR address every time we need to access this routine due to Basic's memory allocation procedures mentioned earlier.



IN MAV'S WORKSHOP
by Ian Mavric
HOW TO UPGRADE A TRS-80 HARD DRIVE

To conclude this trilogy of articles on the TRS-80 hard drive, I'm going to talk about how to change the MFM hard drive in your TRS-80 5Meg (26-1130) or 15Meg (26-4155W) unit to something else. A large amount of the information in this article comes from the late Roy T. Beck who wrote a ground-breaking series of hard drive articles in Computer News 80 in the early 1990s[1]. These articles are very hard to find and so I am presenting elements of them here.

WHY CHANGE THE MFM HARD DRIVE?

The MFM hard drive has been installed in the TRS-80 5 or 15Meg disk system for around 30 years. The Tandon TM602S (5meg) and TM503 (15meg) drives were robust and reliable for the time, but Tandon designed them with a 5 year lifespan, which is a duty cycle of around 11,000 hours MTBF (mean time between failures)[2]. In case you don't have a calculator handy that works out to about 458 days continuous use. Since most people used the drive about 8 hours a day, 5 days a week ((8*5)*260 = 10,400) this is how Tandon arrives at an approximate 5 year lifespan (5 years being 260 weeks). Beyond that the drive is living on borrowed time. The drives were super expensive when new and hence usually had a hard life, often used for more than 8 hours a day and in certain cases (like a database for a BBS) run continuously. Long story short, they are very often worn out by the time I get them and often need replacement due to stiction[3] (google it), platter surface breakdown (bad blocks), or just plain worn out. The solution will be to replace it.

THOSE THREE WIRES

Pretty soon after pulling apart a TRS-80 hard drive you notice three wires connected to various points on the hard drive logic board. If you have worked on old XT and 286 PCs which have MFM drives you know you don't have these three wires to worry about. Tandy complicates things because once the three wires have been removed from the MFM logic board the drive will refuse to format, crashing out with a Write Protect error. So what is going on here. Once I explain what these wires do it becomes much clearer.

ORANGE WIRE

The orange wire connects up with the protect switch on the front panel of the hard drive, when disconnected from the hard drive the controller board believes its getting no signal (equivalent of the write protect button pressed in) so won't permit any data to be written to the hard drive. This wire is important and can't be ignored[4, and] Roy found if you solder it to a place on the HDC then you can restore its functionality. On the 5Meg 26-1130 and 15Meg 26-4155W (early "Radio Shack") drives with the big 8x300 HDC you need to solder that wire to R27 pin 6, which is the pin closest to the 8x300 CPU (see Picture 1). On the later small WD1010 HDC you need to solder the orange wire to the leg of R54 closest to J10 (see Picture 2). While you are at it you need to bend J1 pin 5 (8x300 HDC) or J6 pin 5 (WD1010 HDC) over so it no longer goes into the 20-way data cable to the hard drive. This prevents +5VDC being sent to the hard drive and shorting it out.

WHITE AND YELLOW WIRES

These two wires go to the Active light on the front panel so the light flashes when the head on the hard drive steps in or out. They can actually be taped off and not connected if you don't want the Activity light but to my mind that's a lazy solution. A much better practice is to solder them to the right places to make the light work again. On the 5Meg 26-1130 and 15Meg 26-4155W (early "Radio Shack") drives with the big 8x300 HDC you need to solder the white wire to J6 solder-pad 6 and the yellow wire to J6 solder-pad 26 (see Picture 3). On the later small WD1010 HDC you need to solder the white wire to the leg of R38 closest to U28 and the yellow wire to the leg of R23 closest to U23 (see Picture 4).

WHAT TO EXPECT

Once these modifications have been made, the drive should still power up and go through its test sequence normally. On the 8x300 the Active light stays off until the self-checking is done, then the light stays on, unless the drive is stepping, in which case the light flashes off momentarily. On the WD1010 the Active light is on when powered up and when all self checking is complete the light goes out - only flashing momentarily while the drive is stepping.

You can now set aside the Tandon MFM and replace it with any MFM hard drive you can find on eBay, Craigslist etc. My

personal preference is for Seagate ST-225 (20Meg) and ST-251-1 (42Meg) hard drives, as well as NEC D3142, D5136H, Miniscribe 8438, and Microscience HH-1050. The Redhill web site[5] has many good MFM hard drive recommendations as well. Before you start setting it up you need to know the specs for the replacement hard drive as they are not always indicated on the drive itself - in fact drives which are labelled with how many heads and cylinders they have are quite rare. Fortunately google is your friend and you can find the specs on just about any MFM hard drive in a few moments. It is handy as well to check if you are installing a voice-coil MFM hard drive - it will always have an odd number... as one platter surface is lost for head positioning data. Case in point is the Microscience HH-1050 which has 3 platters/6 heads but only 5 heads are used, and 1024 cylinders. You can tell RSHARDx (see my article from last issue) the drive has 6 heads but when you format it every track will fail because the drive itself will not allow data to be written to the 6th head. It's read only, with positioning information for the other 5 heads.

A NICE SIDE-EFFECT

A somewhat unexpected side-effect of upgrading your MFM hard drive to a newer one is typically in an increase in speed, sometimes as much as 30%. Here's why: during the years 1981-1989 the technology of MFM drives increased rapidly from small 5Mb and 10Mb hard drives up to 40, 70 and eventually 115Mb hard drives. The MFM interface is inherently fast and the drive always spins at the same speed: 3600 rpm, so the higher capacity the drive, the more densely packed the information on the platter, so more is transferred per unit time. Also the stepping speed of the heads increased as stepper motors became faster and more accurate, and eventually gave way to voice-coil actuation. The Tandon TM602S and 503 are examples of early hard drive technology circa 1982, but installing a larger drive from the late 1980s will almost always result in a noticeable increase in speed.

CONCLUSION

To my mind, hard drives really bring out the best in a TRS-80 system, when working well they are a delight to use, fast and reliable. The average file size being so small on a TRS-80 most users would have a hard time filling a 15Mb hard drive, let alone something larger like 40Mb. If you have any more questions about how to repair or keep your hard drive operational, or want one of your own, feel free to drop me a line.

NEXT TIME: Project to build a Model 1 Power Supply

Please email me at - ianm@trs-80.com

REFERENCES:

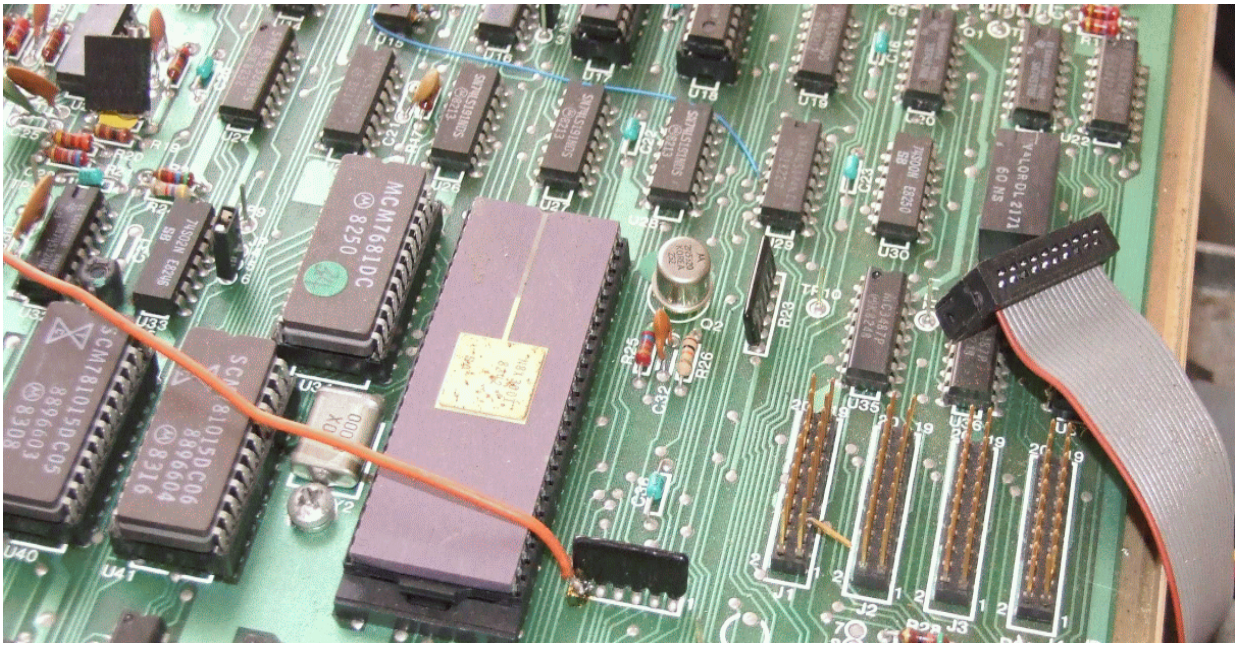
1. "My Adventures with Hard Drives - Part 8", Computer News 80, Vol 4. No. 8, pp.19-23.
2. http://nemesis.lonestar.org/computers/tandy/hardware/storage/mfm.html
3. http://en.wikipedia.org/wiki/Stiction
4. "Radio Shack Service Manual: 5 Meg disk system 26-1130", pp. 27-29.
5. http://redhill.net.au/d/c1.php



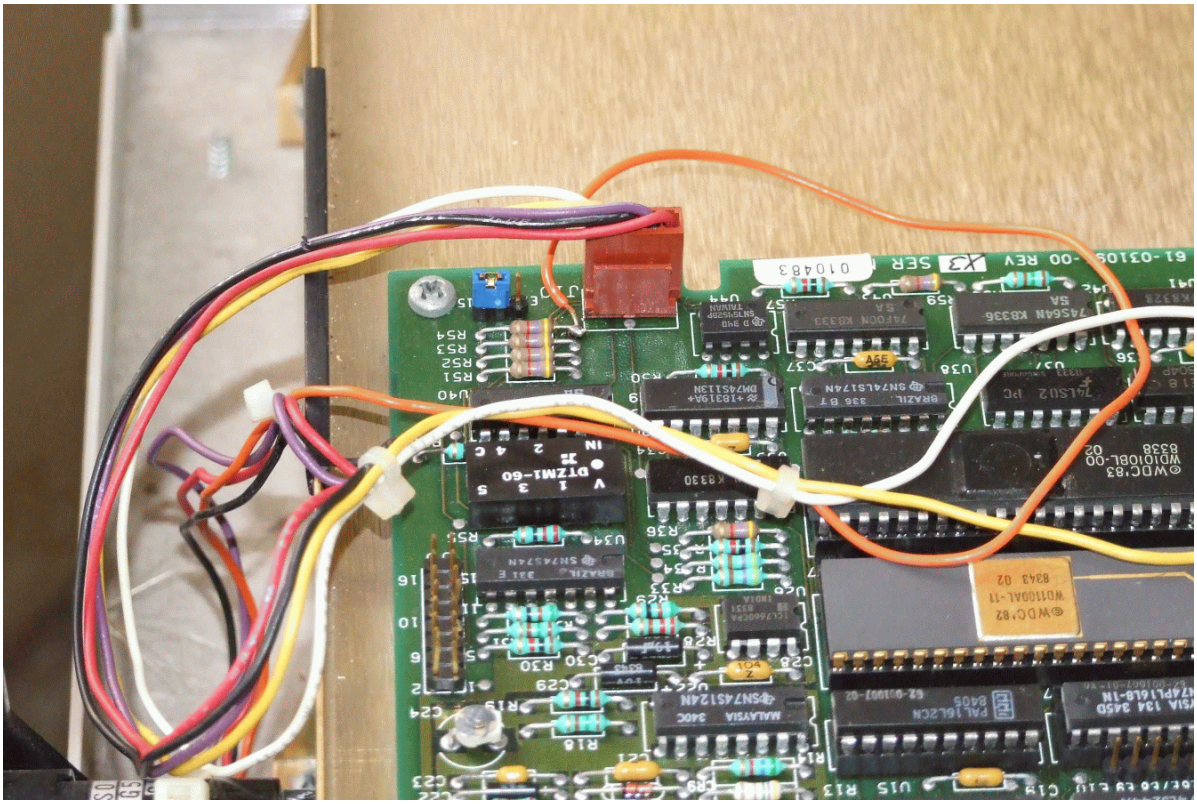
Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.

--== o0o ==--

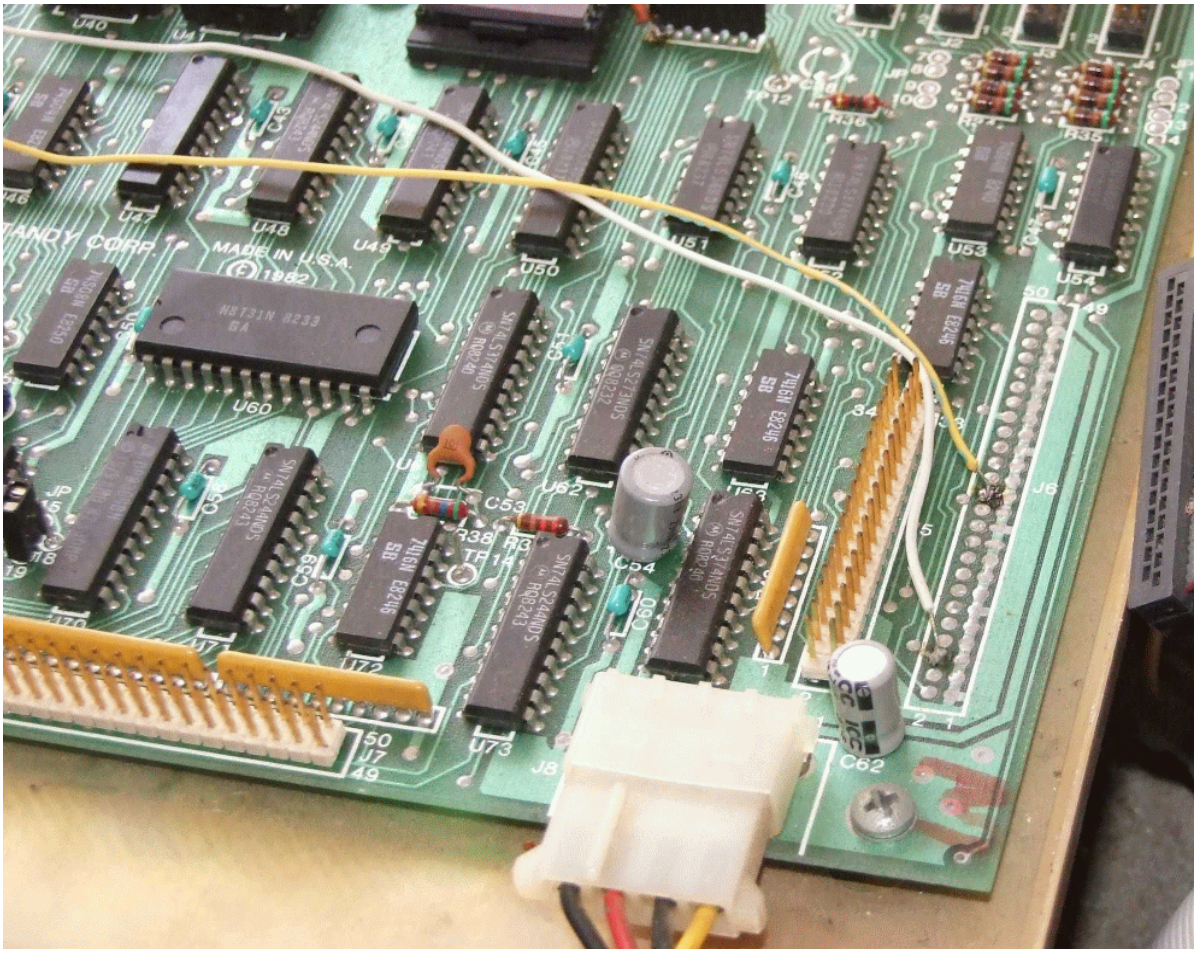




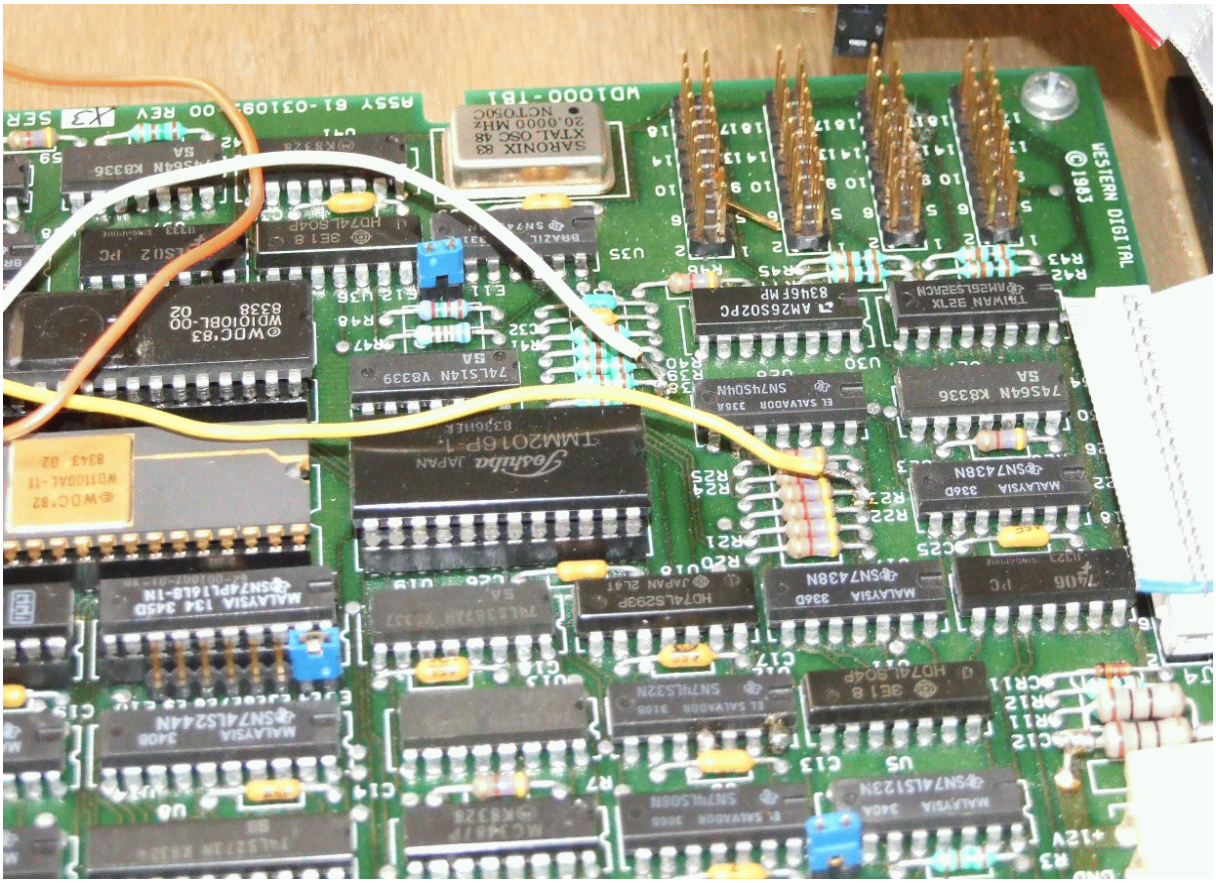
Orange wire in the 8x300 HDC



Orange wire on the WD1010 HDC



White and Yellow wire on the 8x300 HDC



White and Yellow wire on the WD1010 HDC



The *BEST* in TRS-80s

Call The Right Stuff

Ask for Ian
The number is +61 416 184 893

That's The Right Stuff
And he's in Melbourne







http://ianmav.customer.netspace.net.au/trs80/

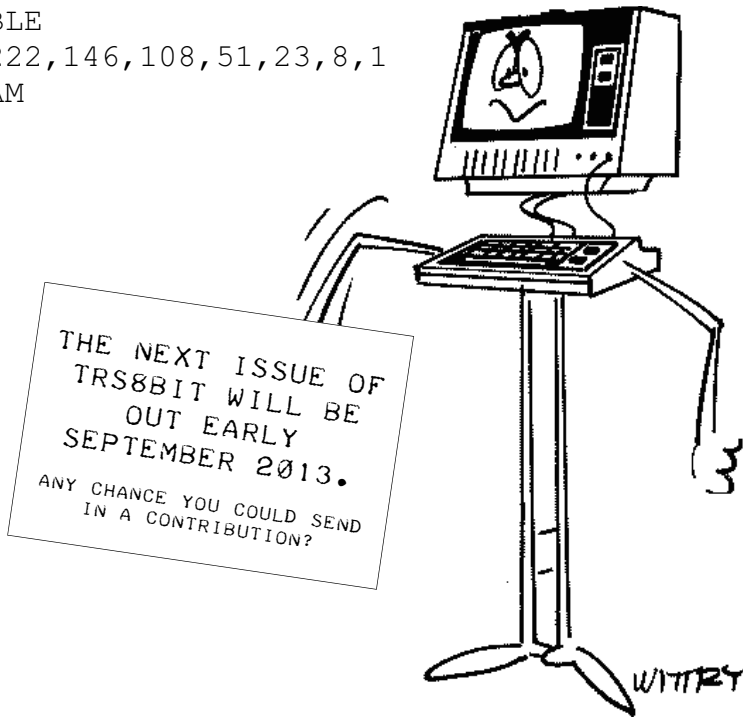
SERIAL PRINTER DRIVER

HERE'S A SUPER LITTLE PROGRAM, WHICH HAS A LOT OF SENTIMENTAL VALUE FOR ME! IT WAS ONE OF THE PROGRAMS WHICH IRA RESCUED FOR ME FROM MY OLD FLOPPY DISKS. IT WAS GIVEN TO ME, MANY YEAR AGO, BY FREDDIE NICHOLS, OF OPTRONICS, IN TWICKENHAM, WHEN I WAS ATTENDING AN OPEN-DAY WITH NATGUG. I REMEMBER AT THE TIME, IT WAS A GOD-SEND AS I HAD JUST BUILT THE CASSETTE PORT SERIAL INTERFACE WHICH LEON HELLER DESIGNED (SEE TRS8BIT VOL 1, ISSUE 2) AND HAVING ONLY A 4K M1 LEVEL 2 SYSTEM, THERE WASN'T MUCH SOFTWARE AVAILABLE, AS YOU CAN NO DOUBT IMAGINE!
THE PROGRAM WAS ALWAYS CASSETTE BASED AS I DIDN'T HAVE A DISK SYSTEM, BUT I CLEARLY REMEMBER COPYING IT TO DISK AT A LATER DATE, "JUST IN CASE". I DON'T THINK I ENVISAGED THE 'CASE' BEING 30 YEARS HENCE THOUGH!

IF YOU WOULD LIKE A COPY OF IT (TO SAVE YOU TYPING IT IN), PLEASE EMAIL ME. I'LL SORT OUT A VIRTUAL DISK OR CASSETTE FILE FOR YOU. IF YOU NEED A 'REAL' CASSETTE PROGRAM, ONCE AGAIN, PLEASE EMAIL ME.

```
10 CLS
15 PRINT"MEMORY SIZES ARE :"
16 PRINT
20 PRINT" 4K - 20360"
30 PRINT" 16K - 32650"
40 PRINT" 32K - 49030"
50 PRINT" 48K - 65420"
60 PRINT
70 PRINT"BAUD RATES ARE  110      450      2400"
80 PRINT"                150      600      4800"
90 PRINT"                300      1200     9600"
100 PRINT"PENCIL BAUD RATES (BX)"
110 PRINT"               110  (B1)   150  (B2)   300  (B3)"
120 PRINT"               450  (B4)   600  (B5)  1200  (B6)"
1000 'BASIC PROGRAM FOR INITIALIZING TRS232 PRINTER INTERFACE
1020 'MADE UNDER LICENCE
1040 'BY ---  OPTRONICS  --- TWICKENHAM
1060 'SET MEMORY SIZE (LEAVE AT LEAST 110 BYTES)
1080 POKE 16553,255 :INPUT "ENTER MEMORY SIZE";MS :MS=MS+1
1100 HB=INT(MS/256):LB=MS-256*HB
1120 POKE 16422,LB: POKE 16423,HB
1140 IF MS>32767 THEN MS=MS-65536
1160 '
1180 'POKE PROGRAM INTO MEMORY
1200 FOR I=0 TO 105 :READ D
1220 POKE MS+I,D :NEXT
1240 '
1260 'SET BAUD RATE
1280 INPUT "BAUD RATE";B :BR=1
1290 IF B=110 THEN 1500
```

```
1300 IF B=150 OR B=300 OR B=450 OR B=600 THEN 1380
1320 IF B=1200 OR B=2400 OR B=4800 OR B=9600 THEN 1440
1340 PRINT "INVALID SELECTION" :PRINT :GOTO 1280
1380 BR=BR+1 :B=B-150 :IF B<>0 THEN 1380
1420 GOTO 1500
1440 BR=BR+1 :B=B/2 :IF B<>600 THEN 1440
1480 BR=BR+4
1500 FOR I=1 TO BR :READ D :NEXT
1540 DH=INT(D/256) :DL=D-256*DH
1560 '
1580 'POKE BAUD RATE CODE
1600 POKE MS+24,DL :POKE MS+25,DH
1620 POKE MS+66,DL :POKE MS+67,DH
1640 POKE MS+76,DL :POKE MS+77,DH
1660 '
1680 INPUT "ADD LF AFTER CR (Y/N)";Q$
1700 IF Q$="Y" THEN 1800
1720 IF Q$<>"N" THEN 1680
1740 FOR I=1 TO 4 :POKE MS+48+I,0 :NEXT
1780 '
1800 INPUT "NUMBER OF NULLS (0-127)";N
1820 IF N<0 OR N>127 THEN 1800
1840 POKE MS+42,INT(N+1) :END
1880 '
1900 'MACHINE LANGUAGE PRINT PROGRAM
1920 DATA 243,121,254,13,40,3,254,32,216,245,229,197,6,9,55
1940 DATA 245,245,33,1,252,205,33,2,33,222,0,43,124,181,32
1960 DATA 251,241,31,245,48,19,33,0,252,24,19,14,2,175,13,40
1980 DATA 2,24,219,62,10,24,215,24,47,198,0,33,1,252,205,33
2000 DATA 2,0,0,33,222,0,43,124,181,32,251,16,212,17,222,0
2020 DATA 203,74,40,11,33,0,252,205,33,2,27,122,179,32,251
2040 DATA 241,241,254,13,40,198,183,40,197,193,225,241,201
2060 '
2080 'BAUD RATE TABLE
2100 DATA 615,450,222,146,108,51,23,8,1
2120 'END OF PROGRAM
```



IN MAV'S RETROSPECTIVE WORKSHOP
THE TANDY TRS-80 MODEL 2000
Was it the finest TRS-80 ever?
by
Ian Mavric

By 1982 the microcomputer world was in a state of change, not just for Tandy but for all computer manufacturers. Prior to 1981 there was just two main markets for micros, namely that of powerful home and small business computers and small cheap home computers. In the former category you have computers like the Apple II, Commodore Pet 4032 and TRS-80 Model III, and the latter category was contested by computers like the Commodore VIC-20, TRS-80 Color Computer, and Sinclairs. Just about this time IBM decided to join the market with its IBM PC aimed squarely at the market enjoyed by the Apple II, Pet and TRS-80 Model I/II/III, and its computer was a run-away success. It was anything but cheap and so had little effect on sales of cheap home computers, though it did have an effect on most other business orientated micros. Suddenly CP/M, monochrome screens, 64K ram and single sided disk drives looked very old fashioned. IBM was onto a winner and software companies flocked to the machine. Compatible machines were produced and very quickly a new standard for microcomputing emerged which is still with us today, the PC.

Most manufacturers realised to survive they needed to make PC compatible machines and out of that came companies like Compaq and Dell. But going back to '82, even Tandy realised that it's new powerhouse products for 1983 (the Model 4, 4P, 12 and 16) were going to be eclipsed by the IBM PC and may have a limited lifespan beyond about 1984. You have to hand it to Tandy for sticking to their beliefs and we have some of the finest Z80-based computers ever made in the Model 4 and 4P, but faced with dwindling sales projections they decided to bite the bullet and produce a PC of their own, but not just any PC, this would be the super-est PC of them all. No detail would be overlooked and every aspect of the computer needed to be an improvement on the IBM PC/XT (which was the version on the market in January 1984) but sell for around the same price as the IBM. In November 1983 Tandy proudly announce the newest, greatest TRS-80 yet, the Model 2000. It would outdo IBM from every angle, but had one major flaw, it wasn't properly compatible; it was only somewhat compatible.

The 2000 graced the cover of 80micro in January 1984 and in fact the computer was well received by the computing press

at the time, who were growing bored of plain compatibles which did nothing in particular better than the IBM PC/XT. The 8MHz 80186 CPU had the 2000 running rings around the IBM, the 720K disk drives held twice in information the IBM drives did, and the display was sharper and if you went the whole nine yards and added hi-resolution colour graphics and a CM-1 monitor you got 16 colour 640x400 graphics (not related to VGA which was still 4 more years away) which made the IBM's CGA 4-colour 320x200 graphics look positively lame. Priced competitively with the IBM PC/XT and much cheaper than the soon-to-be released IBM PC/AT it looked like the 2000 was going to be THE machine to have, but that software compatibility problem came back to bite Tandy on the ass. Testing most software for the IBM PC/XT, some 90% didn't work, but in fact if the software companies took the time to patch their programs to run on the 2000, you often got the best result. A good example of this was 1983's killer app Lotus-1-2-3, which didn't run on the 2000 in its standard form. It took Lotus programmers only 6 hours to patch it work on the 2000, after which they proudly declared the 2000 to be the finest machine to run 1-2-3 on. Even Bill Gates sung the praises of the 2000, appearing in an advert for Tandy, saying how his latest and greatest invention, Windows, worked best on a powerful computer with lots of memory and sophisticated graphics, and that the 2000 was ideal. The only problem was that by the time the computer had been out for 4 months (March 1984) it was clear that not many people were buying it.

It turns out that the minimum requirement to be successful in the PC compatibles world was indeed to be compatible and for most of the software it wasn't. If software companies could be bothered to make their software compatible with the 2000 then it worked well exceptionally well (look at Microsoft's Flight Simulator 1.0 and Windows 1.0, Ashton-Tate's dBASE II, and Autodesk's AutoCAD), but along with not many people buying the 2000, software companies lost interest in patching their programs for the 2000. In April 1984 Tandy, recognising the problem, had an emergency meeting with its computer design teams and laid down the plans for a new project "August", which was to be a properly PC compatible computer that was much cheaper, and be ready to announce in four months and on the shelf in six. We know it as the original Tandy 1000.

Those who have used the Tandy 2000 often have favourable memories of it and indeed I was mightily impressed by it when I got mine. I didn't see it as a replacement for the Model 4, that wasn't until later on when the Tandy 1000 came along and it was clear from the pricing that the 1000 was

going to impact on Model 4 sales. The 2000 was always an expensive beast. A basic dual drive Tandy 2000 with 128K could be had for a "modest" $A4450 but you could easily spend over $A10,000 by the time you got the fully expanded Tandy 2000HD, upgraded it to 768K, hi-resolution graphics, colour monitor, etc. It never seemed to be a serious threat to the $A2299 Model 4 – the jobs both computers excelled at were vastly different.

In my study of the history of Radio Shack and its computing products and their place in microcomputing, I found some interesting titbits of information about the TRS-80 Model 5, obviously planned as a replacement for the Model 4. Looking at feedback from TRS-80 Model III and 4 users, they compiled a wish list of specifications for the new machine and reading through them they look strangely familiar:

Increased memory past 128K, preferably 256K or more
A 16-bit CPU of around 6-8MHz
Hi Resolution graphics as standard
Colour and sound, better than the Apple IIe and IBM PC/XT
Vastly increased disk storage capacity
Internal hard disk drive
More function keys
Backwards compatibility with the Model III/4

Reading this list, except for the last item, reads like the specs list for the Tandy 2000. With projects like August on the drawing board, no wonder future development of Z800 (which Zilog was lazy to release anyway) based advanced TRS-80s was shelved. Examined in this context the Tandy 2000 really does appear to be the zenith of TRS-80 development, and although a failure in the marketplace, it's a success from a performance and reliability aspect, as well as setting very high final standards for ergonomics and user eye, neck and hand fatigue (these becoming big issues in the mid 1980s).

So next time you see a sad looking Tandy 2000 languishing on eBay, CraigsList or a thrift store, give thought to what was at the time one of the most powerful and expensive TRS-80s, one of the last real signs that Radio Shack was trying to be a force to be reckoned with, producing innovative new products. Instead of just languishing in the small-but-comfortable monochrome Z80 world, or the "me too!" plain PC compatible zone, tried to really make a really stunning, classy TRS-80.

ianm@trs-80.com

ULTRA PERFORMANCE MS-DOS SYSTEM



Introducing the Tandy TRS-80 Model 2000 Personal Computer

2-Disk Tandy 2000 **4450.00**

Hard Disk Tandy 2000 HD **6950.00**

Non-interface monitor, color graphics and Digi-Mouse not included

- **Dramatic Speed and Increased Storage Capacity Let You Perform Bigger Tasks in Less Time**
- **User-Expandable for Your Future Needs**
- **Choose From the Hottest Names in Software**
- **Stylish, Ergonomic Design Gives You Maximum Versatility. Productivity and Comfort**

Setting New Standards for Personal Computers. The Tandy 2000 is a truly remarkable computer that offers you much more than other 16-bit MS™-DOS based computers More speed. More disk storage. More expansion. Advanced graphics. And the superior quality and support that keeps Tandy Corporation at the forefront of the industry.

Large Library of Proven Software. Rapidly becoming an industry standard, MS-DOS lets us bring you the most popular and sophisticated programs on today's market. Programs like dBase II™, Microsoft-Multiplan™, PFS® :File, MultiMate™ and many more. Others are under development, including the state-of-the-art MS-Windows operating environment from Microsoft® .

Extremely High Speed. A "next generation" true 16-bit CPU makes the Tandy 2000 dramatically faster than other MS-DOS computers. That means all the Tandy 2000 software runs faster, so you get the job done quicker.

Vast Storage Capacity. Tandy 2000's two floppy disk drives let you store more documents and crunch through bigger tasks with ease. For even greater storage, the Tandy 2000 HD features a built-in 10 million character hard disk drive.

Flexible, Ergonomic Design. Tandy 2000 easily adapts to you and your environment. In fact, you can actually "fine-tune" the modular system to your personal needs. Using our space-saving options, you have the smallest possible desk "foot-print" — only 21.6 x 31.1cm. And the low-profile, 90-key keyboard with 12 function keys and a 10-key numeric datapad assures fast, comfortable typing.

Easy Expansion Capabilities. Add additional memory, stunning high-resolution monochrome and color graphics, our Digi™-Mouse and more. And you can install most of these options yourself.

Compare the Tandy 2000 to Other MS-DOS Computers. In actual benchmark comparisons, Tandy 2000's true 16-bit microprocessor performed almost three times faster than the IBM® PC. Each Tandy 2000 disk drive offers over four times the storage of an IBM PC drive and twice that of the TI® Professional. Our color graphics are twice the resolution of COMPAQ™ or IBM, and offer twice as many colors. And all four of our user-accessible slots are available for future growth. Putting it simply, the Tandy TRS-80 Model 2000 delivers extraordinary quality with higher performance than other popular computers, including the IBM PC.

128K 2-Disk Tandy 2000. 26-5103 . . . . . 4450.00

256K 1-Disk Tandy 2000 With 10-Megabyte Hard Disk. 26-5104 . . . . . 6950.00

Compare the Tandy TRS-80 Model 2000 to the IBM Personal Computer

Price Comparison†	Tandy Model 2000	IBM Personal Computer
Base Unit	$4450.00	$3003
2nd Drive	Included	$669 (320K)
Monochrome Monitor	$349.00	$588
Display/Printer Adapter	Included	$392
128K RAM	Included	$156
RS-232	Included	$158
MS-DOS 2.0	Included	$115
Total Cost*	$4799.00	$5081
Feature Description		
Internal Memory	128K Standard	64K Standard
Disk Capacity Per Drive	720K	160 or (320K optional)
Clock Speed	8 MHz	4.7 MHz
True 16-Bit Microprocessor	Yes (80186) 16 bit/16 bit data path	No(8088) 16 bit/8 bit data path
User-Available Expansion Slots*	4	2
Graphics Options		
Color Resolution	640 x 400	320 x 200
Number of Colors	8	4
Monochrome Resolution	640 x 400	640 x 200
Graphics on Monochrome Monitor	Yes	No

*Comparable IBM configuration with monochrome adapter and display, communications adapter, two 320K disk drives and 128K RAM. †Manufacturer's pricing as of 2/6/84

"Designing a revolutionary concept in software demanded a computer with extraordinary performance. The Tandy 2000 delivered." —Bill Gates
Chairman of the Board, Microsoft

Bill Gates has been at the leading edge of personal computing from the very beginning. His company is a leading producer of microcomputer software.

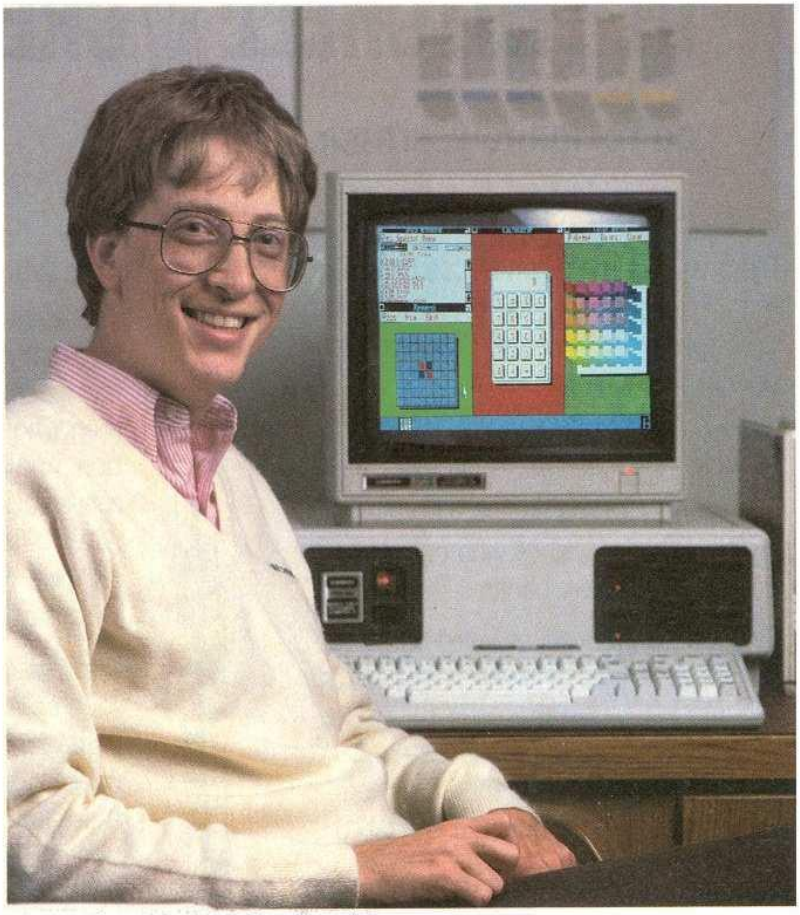
"Our newest software product, MS-Windows, is an integrated windowing environment. It will let personal computer users combine individual programs into a powerful, integrated system.

"When we set out to design MS-Windows in color, we knew that the Tandy 2000 computer would let us turn an extraordinary product into a work of art. The graphics are sharp and crisp, and gave us a degree of creativity like nothing before.

"Our engineers were quite impressed with the processing speed of the Tandy 2000's 80186 microprocessor, too. And while the finished product will utilize the 2000's Digi-Mouse, the well-laid out keyboard has helped us speed through the design stage.

"We're proud of our work. So when we want to show someone how great MS-Windows really is, we give them a demonstration. On the Tandy 2000."

Isn't it time you enjoyed peak performance from a personal computer? Go ahead, watch how much faster today's most sophisticated programs run on the high-technology Tandy 2000.



You can choose from the hottest programs around, thanks to our exclusive Express Order Software service.

Tandy 2000 systems start at $2999, and can be leased for only $105 per month. Come in today and see what you've been missing.

TANDY ADVANCED TECHNOLOGY SERIES
Radio Shack ...AND AFFORDABLE TOO

Engineered for Excellence!

We've introduced the latest in technology for over 60 years. The Tandy 2000 offers twice the speed, graphics resolution and disk storage of other MS-DOS systems.

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Send for our New 1985 computer catalog.

Mail To: Radio Shack, Dept., 85-A-564
300 One Tandy Center, Fort Worth, Texas 76102

NAME _____
COMPANY _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____
TELEPHONE _____

*Plus applicable use/sales tax. Prices apply at Radio Shack Computer Centers and participating stores and dealers. MS is a trademark of Microsoft Corporation.

BELIEVE IT OR NOT!

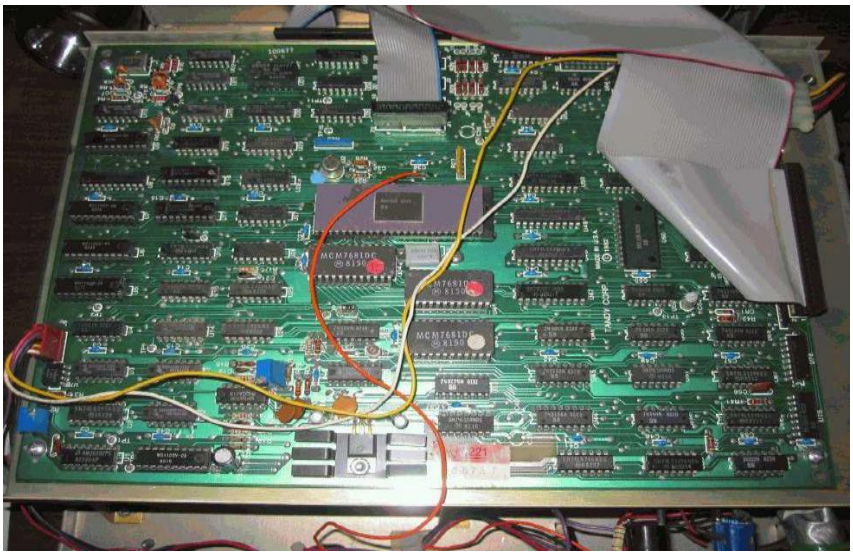
BOB BOYD BOUGHT A HARD DRIVE OFF EBAY AND IT DIDN'T WORK, SO HE HAD SOMEONE THERE PERFORM THE 'ROY T BECK' 3-WIRES MODIFICATION (WHICH IS EXPLAINED IN THIS ISSUE, SEE PAGE 21) BUT THEY DID IT WRONG, SO BOB SENT ME THE HARD DRIVE UNIT FROM PHILADELPHIA, USA TO MELBOURNE, AUSTRALIA BUT THE WD1010 HDC BOARD HAD BEEN SHORTED OUT AND WAS NOT FIXABLE.

BOB THEN LOCATED ANOTHER HARD DRIVE ON EBAY BUT WHEN HE GOT IT HOME FOUND A COUPLE OF HORNET'S NESTS INSIDE IT AND IT WAS THE EARLIER 8X300 HDC ANYWAY. HE SENT IT TO ME TO SEE WHAT I COULD DO WITH IT AND AFTER SOME HOURS CLEANING THE BOARD AND DRYING IT IN THE HOT AUSTRALIAN SUN, I CAUTIOUSLY PLUGGED IT IN AND IT WORKED! IT NEEDED A FEW FINE ADJUSTMENTS BUT IT WORKED WELL AND WITHIN A WEEK WAS ON ITS WAY BACK TO PHILADELPHIA, WHERE ITS IN DAILY USE NOW.

BEFORE

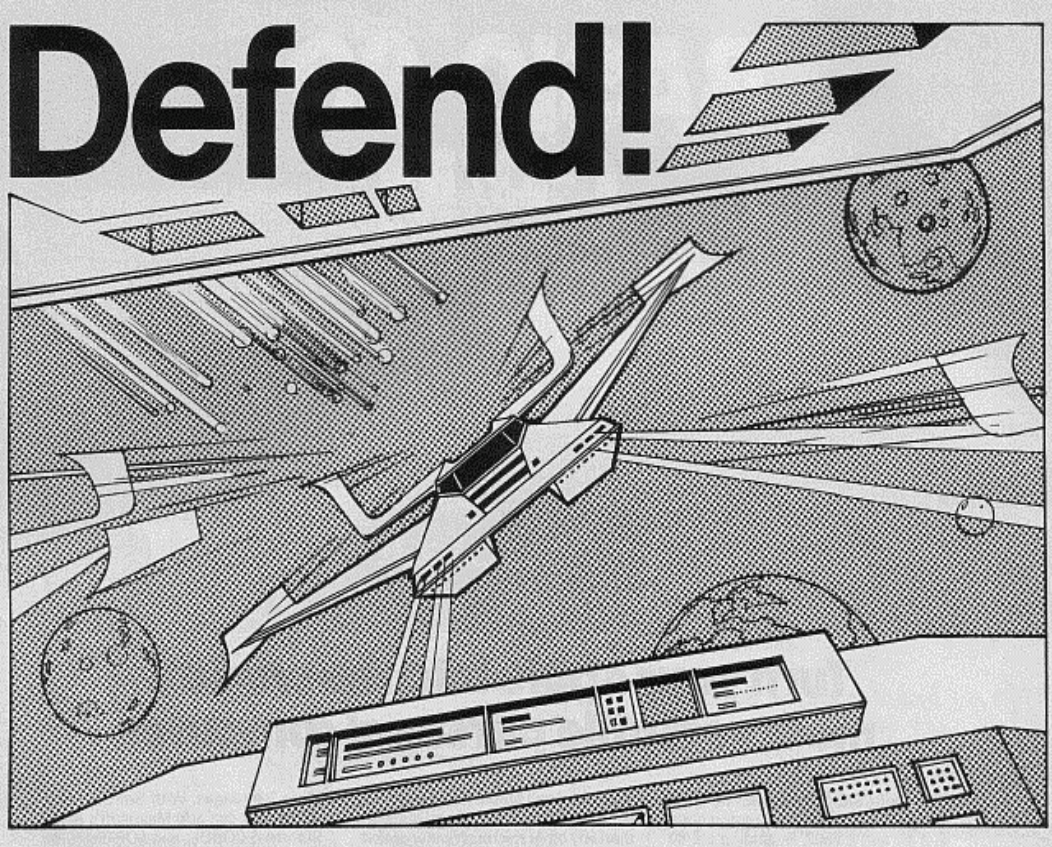


AFTER



INNOVATIVE
TRS 80-GENIE SOFTWARE
from the professionals

Defend!



First there was Invaders, then came Asteroids, and now DEFEND!!!

Carrying on in the same tradition, Defend is a fast arcade type action game, complete with sound effects. Enemy spaceships come at you fast and furiously. If you succeed in shooting them down before they get your ships, you must still get yourself through a meteor shower (but at least they don't shoot at you) and finally, if you emerge unscathed, you must navigate a tunnel in order to get yourself completely out of danger. An enthralling game with excellent graphics, personalisation of highest scores and points bonuses. One of its best features is the "crisp" and immediate control the player has over the manoeuvreability of his ship which includes diagonal movement. Machine language, of course, for speed. A matter of taste, but we think it beats Invaders and Asteroids. Suitable for TRS-80 Models I and III and all Genie models.

Tape (16K) ..... £13.00 + V.A.T. = **£14.95**
Disk ..... £16.00 + V.A.T. = **£18.40**

MOLIMERX LTD
A J HARDING (MOLIMERX)

1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

TEL: [0424] 220391/223636 **TELEX 86736 SOTEX G**

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 [**refundable**] plus £1 postage.

TRS8BIT

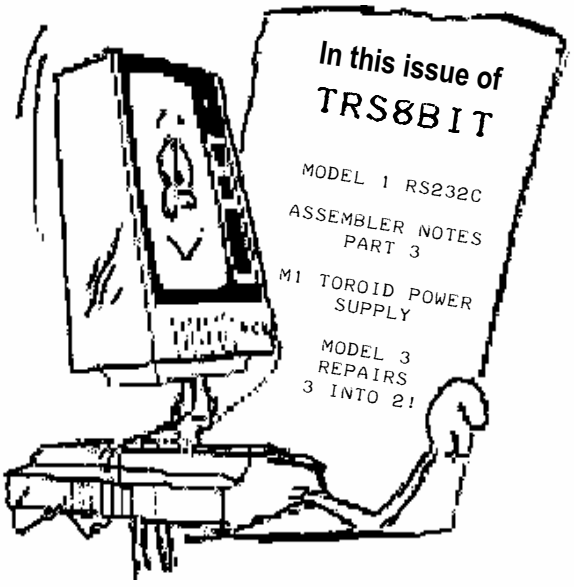


WELCOME, ONE AND ALL TO THE SEPTEMBER 2013 EDITION!

THERE'S SO MANY EXCITING THINGS TO REPORT, I'LL GET STRAIGHT DOWN TO THE 'NITTY GRITTY'.

MAV AND COLLEAGUES HAVE BEEN WORKING HARD TO PUSH THE DEVELOPMENT OF 'FREHD' FORWARD TO SUCH A POINT, THAT, AS AT PUBLICATION, THE M1, 3 AND . VERSIONS ARE UP AND RUNNING. THEY ARE AVAILABLE FROM MAV'S WEBSITE, IN EITHER KIT FORM OR COMPLETE AND READY TO USE. I UNDERSTAND THAT A 'MOD' IS REQUIRED FOR THE M1 VERSION BUT THE OTHERS JUST 'PLUG-IN AND GO'. THE M.P VERSION IS EVEN SELF-BOOTING!
MAV ALSO CAN SUPPLY ANY NECESSARY ADDITIONAL CABLES TOO.

TO ME, THIS IS AN OUTSTANDING ACHIEVEMENT AND I'M PLEASED TO HEAR THAT THERE SEEMS TO BE ENOUGH INTEREST SHOWN TO JUSTIFY ALL THEIR HARD WORK AND EFFORTS. I BELIEVE THAT THIS HAS TO BE GOOD NEWS FOR EVERYONE INVOLVED WITH TRS-80'S.



MATTHEW REED HAS RELEASED VARIOUS UPGRADES TO MOST OF HIS SOFTWARE. V1.28 FOR THE EMULATOR, V1.11 FOR TRSTOOLS AND V..3. FOR TRS-READ/WRITE. FULL DETAILS ARE AVAILABLE FROM HIS WEBSITE.

THE MODEL 1 HAS JUST HAD IT'S 36TH BIRTHDAY. I HOPE THEY HAD LOTS OF CAKE AT THE PARTY!

I'VE COME ACROSS A FIRM, BASED IN THE UK, WHO HAVE OFFERED TO HELP WITH TANDY REPAIRS. 'CANDU-ICT.COM' ARE BASED IN LINCOLN. I'VE PLACED CONTACT DETAILS ON THE HARDWARE SUPPORT PAGE OF OUR WEBSITE.

FOR THE NEXT EDITION, MAV HAS GENEROUSLY OFFERED TO RUN A COLUMN ENTITLED "ASK MAV", WHERE HE'LL ENDEAVOUR TO ANSWER YOUR HISTORY AND HARDWARE COMPATIBILITY QUESTIONS. PLEASE CONTACT HIM DIRECTLY, VIA HIS WEBSITE.

ON THE TRS-80 FORUM, THERE HAS BEEN VARIOUS POSTINGS REGARDING A .8K UPGRADE WITHIN THE KEYBOARD USING .16. CHIPS. I'VE COME ACROSS VARIOUS HARDWARE 'ADD-ONS' WHICH INCREASED MEMORY USING .116 CHIPS OVER THE YEARS, BUT I'D NEVER SEEN ANY INFORMATION REGARDING THE USE OF .16. CHIPS. I'LL TRY AND OBTAIN DETAILS FOR THE DECEMBER EDITION.

DEE AND I MANAGED TO VISIT BOTH BLETCHLEY PARK AND THE NATIONAL MUSEUM OF COMPUTING DURING JULY. BP, AS YOU'LL PROBABLY GUESS, WAS FASCINATING AND WE THOROUGHLY ENJOYED THE GUIDED TOUR. THE NMC WAS JUST UP OUR STREET TOO, EVEN THOUGH THERE WAS VERY LITTLE INFORMATION ABOUT TRS-80'S. THEY HAVE A VAST LIBRARY OF REFERENCE MATERIAL, BOTH BOOKS AND UK MAGAZINES WHICH IS AVAILABLE FOR MEMBERS TO USE FREE OF CHARGE. I ONLY WISHED WE LIVED A LITTLE NEARER!

I'M PLEASED TO REPORT THAT, ONCE AGAIN, THERE HAS BEEN OVER 120 DOWNLOADS OF THE LAST EDITION OF THE NEWSLETTER. THANK YOU ALL FOR YOUR CONTINUED SUPPORT.

SO, DOWN TO THIS ISSUE. MAV CONCLUDES HIS SERIES OF ARTICLES ON THE PROBLEMS WHICH CAN BE ENCOUNTERED WITH THE M1 RS232C INTERFACE. THIS ARTICLE WAS MISSED OUT FROM THE LAST ISSUE. THIS WAS ENTIRELY MY FAULT FOR WHICH I OFFER MY APOLOGIES AND HOPE IT HASN'T SPOILT YOUR ENJOYMENT OF THE SERIES TOO MUCH.

JUST FOR ALL OF US STRUGGLING TRAINEE ASSEMBLER PROGRAMMERS, LAURIE SHIELDS HAS KINDLY UPDATED AND REVISED PART 3 OF HIS 'ASSEMBLER NOTES' SERIES. LAURIE TAKES US THROUGH THE 'CALL' SUBROUTINE AND PLACING OUTPUT TO THE VIDEO.

MAV GIVES US DETAILED INSTRUCTION ON HOW TO BUILD THE TOROID MODEL 1 POWER SUPPLY.

AND FINALLY, MAV, IN HIS USUAL EXPERTLY DETAILED MANNER, TALKS US THROUGH VARIOUS PROBLEMS, WITH THEIR SOLUTIONS, HE ENCOUNTERED WHEN REPAIRING TRS-80'S.

WELL, THAT'S JUST ABOUT IT FOR NOW. I'D JUST LIKE TO REMIND YOU ALL THAT THE NEXT EDITION WILL BE THE XMAS ONE, SO IF THERE'S ANYTHING IN PARTICULAR YOU'D LIKE TO SEE, PLEASE EMAIL ME AND I'LL TRY MY BEST TO HELP.

TAKE CARE,

DUSTY





HERE'S A PHOTO MAV EMAILED TO ME, OF A FREHD, HOUSED IN A STANDARD 5 1/." DISK DRIVE CASE.

IN MAV'S WORKSHOP

by Ian Mavric

CURING MODEL 1 RS232C SERIAL BOARD WOES - PART 2

Last time[1], I talked about the Baud Rate Generator as installed in the TRS-80 Model I RS232 board, and how to replace it if it goes bad on you. Today's discussion will centre around the UART[2] (Universal Asynchronous Receiver/ Transmitter) chip and its line drivers.

SIMPLE TESTS SAVE TIME AND HELP NARROW DOWN THE PROBLEM

Every so often I find that the RS232 is completely unresponsive to testing. The way I test an RS232 is to short together pins 2 and 3 (Rx and Tx) and run a comms program on the TRS-80 and see if what is typed appears on the screen. It's called a loopback test. If I get no response on the loopback test I connect the RS232 to another known-good TRS-80 via a null-modem cable, both computers running comms programs, and see if there is one way communication. This should be your first step before starting to suspect problem parts. If there is no communication there could be problems with the BRG, UART or line drivers, but if you get 1-way communication, you've already eliminated the BRG as the culprit.

UART OR LINE DRIVER?

This then points to a problem with the UART or line drivers. The UART is easy to spot, its that big socketed 40pin chip and in a Model I RS232 its a TR1602 which design and implementation dates back to the 1970s. It works well at all common baud rates (300, 600, 1200 and 2400) but like a lot of early devices doesn't like to get too hot and is pretty unforgiving if shorted out. Radio Shack in its infinite wisdom puts it an un-vented compartment under a hot monitor so its life in the TRS-80 application is arduous. Then the RS232 manual[3] shows how to build custom cables for serial applications like connections to serial printers and modems, which if you build wrong or short can easily blow the UART. There is not much to testing the UART, but if you know your BRG is good then 90% of the time the problem will lie with the UART. They can be purchased on eBay for about $15 but be sure to request a warranty of some sort, as many have date codes going back to the mid-70s and are "pulls" meaning they are 2nd hand and removed from used office equipment - not in itself a bad thing if the parts are OK.

Despite being electronically fragile, the UART is easy to

replace because its in a socket so if you have spares on hand it can be changed in around 10 seconds. The line drivers however are soldered to the board.

WHY ARE LINE DRIVERS PRESENT?

A line driver[4] is an amplifier used to improve the strength of an analogue or digital signal at its source by driving the input to the transmission line with a higher than normal signal level. This increases the quality of a transmission over a long run of cable, and is especially important for RS232 cables more than a few metres long. Basically they give the signals going out of the RS232 port a bit of extra oomph so without that function there may be no result in the lookback test even though the BRG and UART are functioning correctly. On the Model I RS232 board they are U3 and U4 (which are MC1489 chips) and U12 (which is a MC1488 chip). Since they are soldered in, if you suspect one is causing a problem, unsolder and replace all three - they are easy to find and inexpensive.

COMPATIBLE DEVICES

What do you do if you can't find a TR1602 for your Model I RS232 board? Luckily the backwards compatible nature of some newer more readily available UARTs can help us out. The TR1865 (which is found in the 4P and gate-array 4 and 4D)[5] just drops in and works a treat. Another device you may see is by SMC called the COM8017 and COM8018, these work fine with the Model I, and also can be used to substitute a failed UART in all Model III, 4, 4P and 4D TRS-80s. The only thing you actually need to remember about UART chips on TRS-80s is that the TR1602 will only work on the Model I. I've tried it on a Model 4, it doesn't.

OTHER AREAS FOR CONCERN ON THE MODEL I RS232

There are a couple of other things which can go wrong with the Model I RS232 and I'll cover them quickly. Sometimes Y1, the 5.0688MHz crystal can cause problems, if it doesn't work everything will appear dead because the BRG uses it for timing the baud rates. Also L1 and L2, circular induction loops can cause a short when they are bent over - this isn't really a fault, but the exposed legs of the inductors, when bent over, lean against the exposed circuit traces on the component side of the board, so check them carefully and straighten them out. This problem comes from rough handling of the board when its not installed in the EI. The TERM/COMM switch causes

confusion sometimes, as it switches the polarity of pins 2 and 3 on the RS232. In the TERM position the RS232 acts as per standard specifications[6], in the COMM position it acts like a null modem adaptor[7] has been added to the cable. Finally, the solder coated edge connector can oxidise like all connectors on the Model I and needs occasional cleaning with a pink pearl eraser.

If you have any questions concerning the serial ports on 8-bit TRS-80s, feel free to drop me a line.

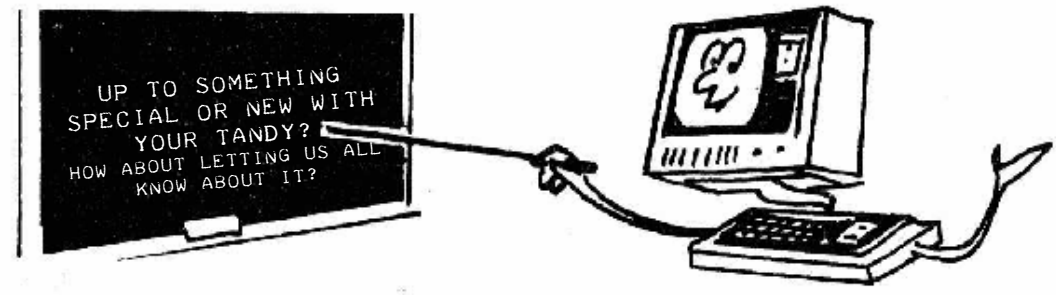
ianm@trs-80.com

REFERENCES:

1. TRS8Bit March 2013, pp. 6-8
2. http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter
3. Radio Shack, 26-1145 Model I TRS-80 RS-232-C Interface manual, 1978, p.31
4. http://en.wikipedia.org/wiki/Line_driver
5. Model 4/4P Technical Reference Manual, Tandy Corp, Cat No. 26-2119, 1985, pp. 142-3 "4.2.16: RS-232-C Circuit".
6. RS-232 Made Easy: Connecting Computers, Printers, Terminals and Modems, Martin D. Seyer, 1984, pp.77-88, 171-173.
7. http://en.wikipedia.org/wiki/Null_modem



Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.



THE NEW POWER SUPPLY FROM:
THE RIGHT STUFF



LOST ONE OR BOTH OF YOUR MODEL I POWER SUPPLIES? ARE YOURS WORN OUT OR JUST PLAIN BUSTED? DID YOU JUST BUY OR INHERIT A MODEL I AND IT DIDN'T COME WITH THEM?

BUY THIS ONE - IT POWERS YOUR MODEL I SYSTEM!

* TAKES ONLY 1 POWER POINT *
* LIVES IN THE EXPANSION INTERFACE *
* AND POWERS BOTH THE KEYBOARD AND E□I□ *

ONLY $A1□5□ØØ + SHIPPING

AVAILABLE FROM:
HTTP://IANMAV□CUSTOMER□NETSPACE□
NET□AU/TRS8Ø/

ASSEMBLER NOTES Part 3
Laurie Shields

Chapter 6

Using the ROM Routines

So far we've learnt how to load registers and memory, count, jump and perform certain logical operations like ADD and XOR etc.
Now for the machine language equivalent of Basic's GOSUB. In machine code these are still called subroutines but instead of accessing them with a GOSUB we use a CALL instruction. As with the jump commands we must provide a reference to the memory address either absolutely in hexadecimal or by means of a label such as CALL SUBR1 or CALL OUTPUT. Just like in Basic when the subroutine has done its job it needs to instruct the Z80 to return to the command immediately following the call and this is achieved with a RET instruction.

Now the Level 2 ROM is jam-packed with hundreds if not thousands of subroutines, most of which are inextricably interlinked and which form the Basic Interpreter. Some of these routines however are available for general use and the first ones we shall meet are to do with data input from the keyboard and data output to the video and printer.

The easiest to get to know is the routine for printing a character on the video at the current cursor position and moving the cursor on to the next. It starts at memory address 0033H and as the leading zeros aren't necessary it is nearly always referred to as simply 33H.

This routine will save us a lot of work as it not only finds the current cursor location, puts our character there and increments the cursor but it also looks after any scrolling of the video that should be needed if we do any printing on the bottom line.
To use 33H we simply load the A register with the character we want to display and then CALL 33H.
On encountering our CALL instruction the Z80 pushes the address of the next instruction onto the stack so that it knows where to come back to, and then it jumps to 33H and starts following the machine code instructions there.

Let’s test this routine to prove that it really works. Use R

to kill off any source file you have in memory and then enter the editor and key in the following code:

```
1           ; TESTING ROM 33H
2           ORG 7800H
3           LOAD 7800H
4 7800 3E2A     LD A,'*'
5 7802 CD3300   CALL 33H
6 7805 00       NOP
7           END
```

The semicolon in line 1 is the Assembler's equivalent to REM in Basic, and the NOP pseudo-op generates a 00 byte, which is a No Operation code as far as the Z80 is concerned. We've put it there just to give us an address for the breakpoint when we test the routine.

Assemble the program and check, with Q7800H, that it has loaded into memory. Now before we jump or goto 7800H to test it, use X to examine the registers first and pay particular attention to the values of AF
and DE. Now jump to 7800H and set a breakpoint at 7805H. Key X again to examine the registers and the video should look like this:

```
Z>X
 HL   DE   BC   AF   RI  IX   IY   SP   PC  Flags
0000 0000 0000 0000 0000                   -------
0000 0000 0000 0000 0000 0000 5627 402D    -------

Z>J7800H
Brkpnt>7805H
Z>X

 HL   DE   BC   AF   RI  IX   IY   SO   PC   Flags
0000 401D.0000 2A20 7200                    --1----
0000 0000 0000 0000 0000 0000 5627 7805     -------
```

Well apart from the fact that only the most recent disc version of Zen has the Flag register decoding, what are we to make of this ?
Firstly, the routine worked and if we look hard enough we can see that the "*" got printed at the cursor position following the last input before Zen jumped to our program. Which, since our last keystroke was an <Enter> or carriage return, was at the start of the line after the 'Brkpnt>7805H".
Secondly the AF, DE, PC and R registers came back with different values to those they initially had on leaving Zen. The A register was of course altered by our program but what about the others ?

We should have expected the F register to alter as that register responds to most sorts of activity, except loading memory and registers, and there must have been a fair bit of it going on with checking on cursor position, displaying the character, incrementing the cursor etc. Likewise the Refresh register and Program Counter have simply behaved in the normal way they would be expected to, but what is 401D doing in the DE register pair ?

Well it's a long story but here goes: Just imagine that you were designing the TRS-80 and were writing the Level 2 ROM. You are faced with the problem of implementing a video display routine using just upper-case characters but you know that sooner or later the computer will be fitted with a lower-case character generator about which you have no information. It is impossible to write a display routine that will handle a chip about which you know nothing and once the Rom is finalised there's no way of altering it.

The solution is to have dedicated address in Rom which will point to your fixed upper-case only Rom video driver program. So that when we want to display a character we call a fixed address in the computer, such 33H, this in turn looks to an address in Ram, say 401DH, to find out where the video driver routine is really located. Normally of course the answer will be the address of your routine in Rom but if some time hence a computer has been fitted with lower-case or there's some other reason for not using the Rom routine then all that is necessary is to put the details of the new display program into 401DH and everything will still work perfectly.

There are actually 8 bytes starting at 401DH dedicated to the video display routine. This block of memory is called 'Device Control Block' usually abbreviated to DCB. It contains some identifying information, the address of the actual subroutine to handle the character output, the current cursor location and a few other bits and pieces.
There are two other DCB's in that area doing much the same thing, at 4015H there is the keyboard DCB and at 4035H the lineprinter DCB.

Keyboard debounce was a problem on the early TRS-80's and a number of programs were written that intercepted the keyboard scan by picking up the address from the DCB and simply executing a delay count before carrying on with the Rom routine. Likewise if your printer needs some special treatment then all you have to do is to put the address of your printer driver module into the DCB at 4026H and all printing from Basic or machine code programs will go to your routine. (all that is, that do their printing properly by calling the ROM).



The *BEST* in TRS-80s

Call The Right Stuff

Ask for Ian
The number is +61 416 184 893

That's The Right Stuff
And he's in Melbourne







http://ianmav.customer.netspace.net.au/trs80/

So what happens when we call 33H. Well the Rom loads the DE registers with the address of the Video DCB (401DH) and jumps to a sort of DCB handler routine that looks at the DCB, checks on whether it is input or output, most of the other registers are saved on the stack, the actual driver program address is loaded into HL and then we jump to where HL is pointing. On return all the saved registers are restored and we are back to our program - except that DE is different.

This highlights one of the problems of using somebody else's subroutines, in that occasionally they use some registers, changing the values in them that we had put there and then not restoring them on the return to our program. We must therefore be very certain, before we use any ROM routines, which registers they use and whether or not they are restored on return. Now, if we refer to Tandy's documentation they tell us that the IY register may be altered as well, so we must be prepared to look after both DE and IY if we want to use 33H.

Since we will probably use it over and over again during a program we may as well set up a general purpose subroutine labeled, say, OUTPUT that we can simply CALL for output. It will look like this,

```
OUTPUT:PUSH DE
     PUSH IY
      CALL 33H
      POP IY
      POP DE
      RETURN
```

By pushing the contents of DE and IY onto the stack, we can allow the ROM to do what it wills with them as after it has finished we simply pop them back off the stack, being careful not to get them crossed over, and thus we are in total control.

There are two other routines concerned with data input/output that are used in a very similar fashion. The first is located at 3BH and outputs to the lineprinter. The difference between this and the one for the video is that 33H is bomb-proof and if we call it then as sure as eggs is eggs the character in the A register will appear instantly on the video even if the whole screen needs to be scrolled, if we call 3BH and a printer isn't connected then the computer will hang up for ever and a day waiting for one to be attached. And what's more even if the printer is there

and working some of them don't get round to actually doing any printing until a carriage return is sent to signal the end of the line.

The second routine is, of course, to do with keyboard input and is located at 2BH. Keyboard input is somewhat different from data output in that should a key not be being pressed when we call this routine then there will not be any data. This keyboard scan is therefore identical in approach to the INKEY$ function available in Basic.

To use it we must first save the DE and IY registers, then call 2BH, restore the registers and test the contents of the A register to see if it contains any data. It is then up to us to decide if we want to hang around repeating the operation until a key is depressed or to carry on with something else and come back to the keyboard again later.

A typical scan the keyboard / display on the video routine with our usual ORG 7800H and LOAD 7800H at the start might look like this:

```
1               ORG 7800H
2               LOAD 7800H
3 7800 CDOA78   GETCRAR:CALL KEYSCAN
4 7803 B7           OR A
5 7804 28FA         JR Z,GETCHAR
6 7806 CD1478       CALL OUTPUT
7 7809 00           NOP  ; Just to give a breakpoint
8                      ; Rest of Program
9 780A D5       KEYSCAN:PUSH DE
10 780B FDE5        PUSH IY
11 780D CD2B00      CALL 2BH
12 7810 FDE1        POP IY
13 7812 Dl          POP DE
14 7813 C9          RET
15 7814 D5       OUTPUT:PUSH DE
16 7815 FDE5        PUSH IY
17 7817 CD3300      CALL 33H
18 781A FDEl        POP IY
19 781C D1          POP DE
20 781D C9          RET
21              END
```

Test this routine as we did the others by Jumping or Going to 7800H and setting the breakpoint at 7809H. What should happen is that the TRS-80 will appear to go into suspended animation until a key is depressed whence the character corresponding to the keystroke will be displayed and then control returns to Zen from the breakpoint.Let’s look at

the program in more detail.

Line 3 : This is a call to our subroutine KEYSCAN. Note we have not saved any registers so should there have been anything important in A or F then we have lost it.

Lines 9 to 14 : This is the subroutine KEYSCAN. First the DE and IY registers are pushed onto the stack and the ROM keyboard scan is called. If a key is pressed whilst 2BH is executing the character will
be in the A register, otherwise it will contain a zero. The state of the flags in the flag register are not meaningful. The IY and DE registers are popped off the stack and then the routine returns.

Line 4 : We must now check on whether the A register contains a zero which means no key depression, or non-zero. To do this is it necessary to trigger the Flag register to the contents of the Accumulator. The quickest and easiest method is simply to OR the A register with itself. If A is zero then the Z (zero) flag will be set, otherwise it will be NZ.

Line 5 : If the A register is zero then jump back and do it all again, otherwise carry on. This call, test and jump back loop, is typical of a keyboard input routine.

Line 6 : The A register now has the character for the video so call the display routine.

Line 7 : Just there to give a breakpoint address.

Lines 15 to 20: The video display routine. Saving DE & IY and calling 33H, then restoring IY & DE before returning.

Examining the registers before and after we test this program should show that the only ones that do not return their initial values are the A and F ones, where the A will contain the keystroke character and the F could be anything.

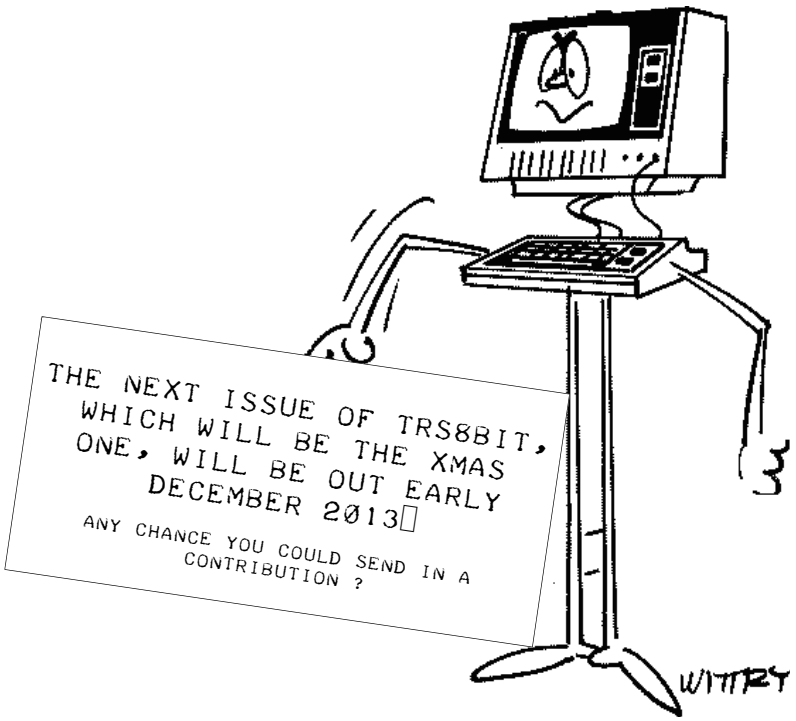
The problem with this routine is that it is limited to just one keystroke and more often that not we will want to input something more meaningful such as a name or a quantity. The first thing we need to do is to keep on accepting characters until the <Enter> key is pressed.
Insert the extra lines to the program so the beginning looks like this:

```
1              ORG 7800H
2              LOAD 7800H
3 7800 CD0E78   GETCHAR:CALL KEYSCAN
4 7803 B7          OR A
```

```
5 7804 28FA         JR Z,GETCHAR
6 7806 CD1878       CALL OUTPUT
7 7809 FE0D         CP 13
8 780B 20P3         JR NZ,GETCHAR
9 780D 00           NOP
10                : Rest of Main Program etc.
```

Now test it as before but this time the program will carry on accepting characters from the keyboard until the <Enter> key is pressed. We have made some progress but if you think about it not all that much, firstly you will find that by repeated pressing of the back-arrow key you can wipe out everything on the video and secondly, even if we key in something useful, our program hasn't remembered it.

All the characters were only 'known' by the computer for a brief instant between the key being pressed and their appearing on the video. To be really useful our program must store each key depression somewhere so that they can all be handed onto the next part of the program. It must also keep a count of the characters so that backspacing beyond the first one is not allowed. Later on there will be further complications about such things as unprintable characters or those keys which mean special things such as <Clear>, but all of that will wait for the next session.



LOOKING FOR FAST, INEXPENSIVE, UNLIMITED MASS STORAGE FOR YOUR TRS-80 MODEL I/III/4/4P/4D?

The amazing

"FreHD"



-Emulates a TRS-80 hard drive, but faster than any hard drive!
-Works with your favourite DOS (LS-DOS, LDOS, CP/M, Newdos/80 2.5)
-Uses SD card for storage medium
-Bonus free Real Time Clock function!
-Designed in Belgium and proudly built and shipped from Australia
-Kit form or fully assembled

Order yours today:
http://ianmav.customer.netspace.net.au/trs80/emulator

Chapter 7

Putting it on the Video

Now that we are all at home calling the ROM lets develop a few more complete routines to display and input messages. Display first as that's the easy one.

The simplest technique involves a straight forward message ending with a carriage return or CR as it is usually abbreviated. Assembler has a command similar to the LET in Basic for allocating a value and takes the form EQU. In future we will always refer to ASCII 13 as CR and use the EQU pseudo op at the beginning of our programs to initialise it.

OK we want to display a message, say, "PRESS ENTER" in our machine code masterpiece. Obviously somewhere in memory we will have to have these 11 characters stored and we will need to refer to them so we will use the label MESSAGE to have the start address of the message, i.e. the address in Ram of the letter "P". Since our program might also want to display other messages of differing lengths we must include some means of identifying the end of the message and for that purpose we will use the CR.

Review the problem:

1, Put the characters "PRESS ENTER", followed by CR into memory.
2, Start displaying the characters on the video one by one.
3, Keep checking on whether or not a CR has been displayed.
4, If it’s a CR then STOP displaying and return.

It should be obvious by now that what we want is to set up a register pair to point to the message and go round a loop getting the character pointed at, displaying it, checking for a CR and if not incrementing the pointer and looping back. The program looks like this:

```
1              ORG 7800H
2              LOAD 7000H
3          CR: EQU 13
4 7800 210778      LD HL,MESSAGE
5 7803 CD1378      CALL VIDEOUT
6 7806 00          NOP ; For Breakpoint
7 7807 50524553 MESSAGE:DB  "PRESS ENTER",CR
7 780B 5320454E
7 780F 5445520D
8 7813 7E      VIDEOUT:LD A,(HL)
```

```
 9 7814 23           INC HL
10 7815 CD1D78       CALL OUTPUT
11 7818 FE0D         CP CR
12 781A 20F7         JR  NZ,VIDEOUT
13 781C C9           RET
14 781D D5        OUTPUT:PUSH DE
15 781E FDE5         PUSH IY
16 7820 CD3300       CALL 33H
17 7823 FDE1         POP IY
18 7825 D1           POP DE
19 7826 C9           RET
20              END
```

By now you should be so expert at assembling these programs, that jumping to them with the de-bugger and getting back from the breakpoint will be taken for granted, but note the value returned in the HL register of 7813H. This is the next byte in memory after the last character displayed. I suppose that it is worth mentioning that there is really no need to be pushing and popping DE a IY as neither are being used in our programs for anything important, but it should become second nature whenever calling 2BH, 33R or 38B to save these two registers otherwise one day you WILL be caught out.

The fun starts, I suppose, with line 4 where the HL register is given the label Message to take with it on line 5 where VIDEOUT is called. VIDEOUT doesn't get involved with the complications of actually outputting to the video, that's left to OUTPUT. There's enough to be getting on with just picking up the characters pointed to by HL and checking, after OUTPUT has done with them, on whether or not to get some more.

Line 7 is also of interest as we have used Zen's all purpose DB (Define Bytes) pseudo-op. This supercedes the DEFM (DEFine Message) and the DEFB (DEFine Byte) of the more primitive assemblers, and allows the mixture of different operands as long as they are separated by commas. Also notice that for each character in the DB statement there is a corresponding hexadecimal value generated in the object code listing.

VIDEOUT, starting at line 8, is the important routine and has a few points of note. Firstly the HL register arrives with the address of the start of the string of characters for output. The first character is loaded into the A register and immediately HL is incremented. This is a fairly universal convention of maintaining a pointer, be it stored either in memory or in a register, as always pointing to the next character. Once the current character has been loaded from, or to memory or wherever and reference to it is no longer required, the pointer is 'bumped' to the next one. The output

subroutine is then called and whatever was in the A register is now on the video.

We know from earlier programs that the output routine will return with the value still in the A register so we check on whether it was a CR and if not, loop back for the next character. If it was a CR then VIDEOUT finishes with a return.

There is a weakness with this program in that all our strings for output must finish with a CR otherwise VIDEOUT doesn't know when to stop. To make our routine more useful it would be nice if we could detect the end of a string in mid-line so to speak. Since our old friend zero is an easily detected non printable character we can use it to flag the end of a string where we do not want a CR. We must make a small addition to our VIDEOUT program, first: immediately after loading A with each character execute an OR A command and if it is zero then return. Now we meet one of the many aspects of the Z80 that makes life easy for us. We want to return if the zero flag is set, i.e. the equivalent of: IF Z = 1 THEN RETURN. In assembler it's as simple as RET Z.

Thus we have two exits from VIDEOUT, the first is to leave immediately if the next character it gets is zero and the second exit is following the output of a CR.

The program, with a few added messages now becomes:

```
1               ORG 7800H
2               LOAD 7800H
3           CR: EQU 13
4 7800 211378       LD HL,MESS1
5 7803 CD3378       CALL VIDEOUT
6 7806 211F78       LD HL,MESS2
7 7809 CD3378       CALL VIDEOUT
8 780C 212878       LD HL,MESS3
9 780F CD3378       CALL VIDEOUT
10 7812 00          NOP ; For Breakpoint
11 7813 50524553 MESS1: DB  "PRESS ENTER",CR
11 7817 5320454E
11 781B 5445520D
12 781F 54455354 MESS2: DB  "TESTING»» ",0
12 7823 494E473E
12 7827 3E3E3E00
13 782B 3C3C3C2E MESS3: DB  "<<.>>>",CR
13 782F 3E3E3E0D
14 7833 7E     VIDEOUT: LD A,(HL)
15 7834 23          INC HL
16 7835 B7          OR  A
```

```
17 7836 C8          RET Z
18 7837 CD3F78      CALL OUTPUT
19 783A FE0D        CP  CR
20 783C 20F5        JR  NZ,VIDEOUT
21 783E C9          RET
22 783F D5      OUTPUT: PUSH DE
23 7840 FDE5        PUSH IY
24 7842 CD3300      CALL 33H
25 7845 FDE1        POP IY
26 7847 D1          POP DE
27 7848 C9          RET
28              END
```

Testing this version in the usual way will give the following on the VDU. Note that Mess3 follows immediately after Mess2:

```
Z>J7800H
Brkpt>7812H
PRESS ENTER
TESTING>>>><,<.>>>> >>>
Z>
```

There are of course other ways of achieving the same objective. If you should examine a machine code program generated by a compiler or program generator then you will find that usually each message is preceded by a single byte value giving the number of characters for output. This quantity is loaded into the B register and the output routine goes round a loop displaying and decrementing B until B becomes exhausted. There's a nice little Z80 instruction, specifically for the B register, called Decrement and Jump If Not Zero which does it all for us in one go.

Taking our original example and rewriting this way it becomes

```
1                   ORG 78008
2                   LOAD 78008
3               CR: EQU 13
4 7800 210778           LD HL,MESSAGE
5 7803 CD1478           CALL VIDEOUT
6 7806 00               NOP ; For Breakpoint
7 7807 0C505245 MESSAGE:    DB  12,"PRESS ENTER",CR
7 7808 53532045
7 780F 4E544552
7 7813 0D
8 7814 46       VIDEOUT:    LD B,(HL)
9 7815 23       VID1:       INC HL
10 7816 7E              LD A,(HL)
11 7817 CD1D78          CALL OUTPUT
12 781A 10F9            DJNZ VID1
13 781C C9              RET
```

```
14 781D D5      OUTPUT: PUSH DE
15 781E FDE5            PUSH IY
16 7820 CD3300          CALL 33H
17 7823 FDE1            POP IY
18 7825 D1              POP DE
19 7826 C9              RET
20                  END
```

If we didn't want the CR at the end of the message it could easily be omitted provided we reduce the counter from 12 to 11. This particular technique is used in Scripsit to display all the various prompts and messages on the bottom line. It suffers, as does the earlier example, in needing an extra byte, be it the counter at the beginning or the special marker at the end, to indicate the length of the message string.

To do the job really efficiently we need a technique that somehow can manage without any extra bytes. The secret is in appreciating that all displayable ASCII characters have values less than 80H (decimal range 32 to 127), so if we add 128 to the last character of each string then simply by testing if greater than 127 we will know whether or not the whole message has been displayed.

To add 128 or 80H to an ASCII character causes just the seventh bit to become set, everything else stays the same so by simply testing the condition of Bit 7 will be enough to indicate the end of a string. We must of course make sure that we clear this bit before we try to display the character otherwise we would get some weird graphics instead of letters. Obviously our display routine is going to get a little bit longer and more involved, but the reward is in the saving in memory by not needing an extra byte with every message.

So once again we will rewrite our simple example:

```
1                   ORG 7800H
2                   LOAD 7800H
3 7800 210778           LD HL,MESSAGE
4 7803 CD1278           CALL VIDEOUT
5 7806 00               NOP ; For Breakpoint
6 7807 50524553   MESSAGE:  DB  "PRESS ENTE","R".80H
6 7808 5320454E
6 780F 5445D2
7 7812 7E       VIDEOUT:    LD A,(HL)
8 7813 E67              AND 7FH
9 7815 CD1E78           CALL OUTPUT
10 7818 CB7E            BIT 7,(HL)
11 781A 23              INC HL
```

```
12 7818 28F5            JR  Z,VIDEOUT
13 781D C9              RET
14 781E D5      OUTPUT:     PUSH DE
15 781F FDE5            PUSH IY
16 7821 CD3300          CALL 33H
17 7824 FDE1            POP IY
18 7826 D1              POP DE
19 7827 C9              RET
20                  END
```

Line 7 looks interesting and may take some working out.
First of all the CR has gone just to make it easier to explain.
DB "PRESS ENTE","R".80H the fun comes after the comma "R".80H.
Well here is the best explanation you're going to get.
As we want to manipulate the last character 'R' we have to separate it from the rest of the letters "PRESS ENTE". To generate the letter 'R' with its bit 7 set could have been achieved by using "R"+80H but there is a convention with some sense behind it (that I wont go into here) that it is better to use logical operators wherever possible in preference to arithmetical ones. So we are using the assembler's equivalent to the Z80's OR operation. The full stop '.' is used to signify that the bits corresponding to the letter 'R' are to be ORed with 80H, which will always result in bit 7 being set.

VIDEOUT, starting at line 7, gets the character pointed to by HL but before displaying it must ensure that bit 7 of the A register is not set. This could be done either by the command RES 7,A (RESet bit 7,A) but again the logical operation AND is preferred, so we have AND 7FH,
thus filtering out bit 7.
The character in A can now be displayed, but after being output the A register is of no use to us as a check on the bit 7 end of string marker. To do that we have to refer to the actual string of characters and fortunately the HL register is still pointing to the particular character we are interested in. To test bit 7 then is quite easy with the questioning command BIT 7,(HL). The result will be either a Zero or Non-Zero condition in the flag register. If it is zero we go back for the next character having incremented HL first, if it is non-zero then that was the last character and the subroutine having done its job can return.

There are of course, as one might expect, equivalent complete routines available in the ROM and also for disk users in the DOS, which will output complete character strings to the video with just a single call. But there are problems in that the conventions for treating the end of string markers are not

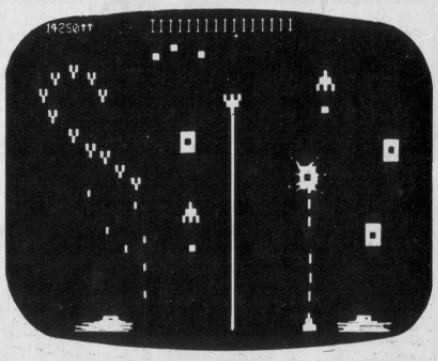
consistent, the ROM routine at 28A7H doesn't work under Newdos and as Tandy do not declare these routines as being available they might change them one day and not tell anybody.

As the amount of coding involved in setting up one's own general purpose output using the official supported 33H is quite minimal, then I would always recommend doing it. That way your programs will need little updating when the Model 99 TRS-80 comes out and should you change computers altogether then the alterations needed, will be kept to a minimum.

--== oØo ==--

They said it couldn't be done!

Galactic FIREBIRD



Those ever circling, swooping, diving Firebirds of the famous arcade machines, brought to the micro in Mike Chalk's ultimate achievement--Galactic Firebird

Yes, it's the latest in the Kansas Arcade series, but alas, his last, for Mike has been writing these right back from the original Arcade Invaders, and feels he wants a rest.

The reason being that he has always bettered his preceeding program, and feels that with Firebird he has reached a pinnacle.

Many programmers of many machines have tried to emulate the arcade game, Firebird, but nobody, the Americans included, have managed to get the circling, swooping firebirds on a computer. Until now!

For TANDY TRS-80 & VIDEO GENIE

It is incredible how the Firebirds peel away, circle around in numerous positions on the screen, then suddenly sweep in to attack, all the while raining down their bombs.

Even if you manage to dodge all their bombs whilst shooting them down, it is as though they realise they will not get you this way, for without warning they'll take kamikazi action and come straight for you!

Then come the Astro Blasters, shooting down a solid energy bolt, to fry you completely, in adition to their own brand of super bomb. Oh, you can fire at them, but a solitary hit will have no effect whatsoever on these nasties.

As though Firebirds and Astro Blasters are not enough, as the game progresses, come the Megon Bombs. With no less than three hits needed to destruct! And you can't dodge them easily either, as their explosions spread.

Scores are kept of course, and there is the now expected realistic sound effects. And--wait for it--it progresses to FOURTEEN levels of difficulty! Mike's last words: "Nobody will ever get there!"

It's ONLY available direct from Kansas but at a sensible £9.50 Vat and post paid

Kansas

As pulishers (we do not retail other people's programs) we have absolute quality control and thus give an unconditional guarantee on all our software. And of course all programs are always in stock for our famed return first class post service whether ordered by cheque or credit card. Ask for a copy of our free catalogue, you won't find any 'South Coast' prices there! Access and Barclaycard welcome.

Kansas City Systems, Unit 3, Sutton Springs Wood, Chesterfield, S44 5XF. Tel. 0246 850357

HULLFORTH

This new Forth compiler for NASCOM I/II is now also available for TRS-80 Level II and Video Genie HULLFORTH is a structured high level language which runs over 10 times faster than BASIC.

- Runs in under 16K
- Supplied on cassette with full documentation
- Nascom users please quote NAS-SYS or NASBUG

Hullforth is a Nascom approved product.

Price — £25

Send SAE for further information to:

Mr A.F.T. Winfield,
148 Goddard Avenue, Hull, HU5 2BP

TRS-80 MODEL I
ENHANCED VIDEO

* TANDY COMPATIBILITY mode (default) gives lower case without the need for switches.
* FULL MODE gives the FULL CHARACTER SET in POSITIVE and **INVERTED VIDEO** and all graphics characters.
* WHOLE SCREEN INVERSION including borders is independently controllable.

Double width capability is not affected.
Mode selection by port FE. Fits inside the keyboard case.

Technical manual with software patches £6. Assembled and tested PCB, 2102 chip £23. Parts and manual ordered together £26.45. The above prices include worldwide postage and United Kingdom VAT. Dealer discounts. Installation available, please enquire.

RHA (MINISYSTEMS) LTD., 83 GIDLEY WAY, HORSPATH, OXFORD OX9 1TQ. Tel: 08677-3625

IN MAV'S WORKSHOP

MODEL 1 TOROID POWER SUPPLY PROJECT

- HARDWARE PROJECT FOR ELECTRONICS ENTHUSIASTS -

by
Ian Mavric (words)
and
Dean Bear (design)

ASSEMBLY PROJECT

Legal note: if any of the following terms alarm you, please get a professional to build the power supply for you: "Lethal Voltages", "Poisonous, corrosive PCB etchant", "hot soldering iron", "possible damage to computer or yourself if built wrong", "the authors and editor bear no responsibility", and "make sure building a power supply like this is within your skill set".

PREAMBLE

The TRS-80 Model 1 uses an external power supply to bring the mains AC voltage (120V in the USA and 240V in Australia/UK) down to approx. 15VAC and 19VDC to power the computer. If you have an Expansion Interface, you need a 2nd one of these power supplies. The power supplies themselves are very simple and for the most part very reliable. Most were made back in the 1970s and still work now. Radio Shack made the US 120V power supplies, and Ferguson in Terrey Hills, NSW made the 240V ones supplied with Australian Model 1s.

Ira, from www.trs-80.com, had been emailing me about the growing number of requests he had been receiving for an alternative power supply for people who needed them. It turns out there was a large number of enthusiasts out there who had either lost their power supplies, or their great eBay purchase didn't come with them, or they only had 1 and needed 2 to run an Expansion Interface, or their own supplies had just plain worn out and failed. Ideally an over-the-counter wall-wort type power adapter would be great but due to the dual output voltage of the original power supply, this is not possible, or convenient.

I decided the best course of action was to design a supply based around 2 transformers (each providing one of the two voltages needed), that had enough power to run both the Model 1 computer and Expansion Interface, but was compact enough to fit inside the Expansion Interface if needed. A great side effect is that only one mains outlet would be needed instead of two.

Dean Bear, a TRS-80 enthusiast from Canberra, came up with a simple design that meets all the design requirements above, and all the parts are easily obtained. I have used one of these for over 6 months now on my main (daily use) Model 1 and have had no bad side effects. Others who have used the supply include David Cooper in Seattle, WA, and Andrew Quinn in Auckland, New Zealand.

In addition, Dean's design uses toroidal transformers which are the latest design in PCB mount transformers. Moulded in blue plastic they are compact, powerful, and don't generate much heat, vibration or electrical noise. They are ideal for installation in the Expansion Interface as they won't cause it to overheat like the US 120V ones tend to do. They also won't make your Model 1 display wobble all over the place.

PARTS LIST:
I bought all my parts for this project from Altronics (www.altronics.com.au) but all the parts except for the two toroid transformers are easily bought from any electronics supplier in the world. The toroid transformers are a little harder to locate but most countries will have a supplier who has them. Also note with the toroid transformers you need the 30VA versions. There are 10VA versions which are a little cheaper, but will not have the capacity to power a TRS-80.

Typeset text from your disks – NOW

Unlike some computer-typesetting services you've seen advertised, wordsmiths' Anvil service is ready for you *now* if you use any of the following machines:

Superbrain SD or QD (or any compatible system); ***Apple][*** under DOS3.3 or CP/M; ***Tandy*** TRS-80 Model I or III; ***North Star Horizon***; ***Comart***; or ***Ohio Scientific*** C2 or C3 series.
(We'll have links to other systems ready by the time you read this advert – ring for details).

We have special software support for ***WordStar, Spellbinder, Scripsit*** and ***Applewriter***: so even tricky jobs such as directories, technical publications, tables and foreign languages are no problem for your word-processor and Anvil.

If you're not sure how to do it, come to one of our ***word-processing workshops*** and see how we can help you to produce anything from a price list to a batch of bar-code labels.

(And since we're in central Somerset rather than on the Central Line, we might even offer you some cyder and a byte to eat as well!).

Quite simply, we're the specialists in print for micro-computer users.

For details of our typesetting, print and other services for the word-processor user, and for dates of our word-processing 'workshops', return the coupon or ring Mike Cooper on 0458 45359.

wordsmiths
computers in type



Name _____ Position _____
Company _____
Address _____
_____ Tel. No. _____

wordsmiths West End, Street, Somerset BA16 0LQ
Tel: Street (0458) 45359. Telex: 46401

QTY	PART		From Altronics
1		7.5v + 7.5V PCB 30VA Toroidal Transformer (Center Tap)	$A30.00 Each
1		6.0v + 6.0V PCB 30VA Toroidal Transformer (Center Tap)	$A30.00 Each
1		1K Resistor	$A0.80 Each
1		LED	$A0.21 Each
1		1000uF 35V Electrolytic Capacitor	$A2.00 Each
2		Fuses (250mA for 240V/500mA for 120V – on each transformer)	$A2.50 for 10
2		3AG/AGC Fuseholder (Optional)	$A2.30 Each
2		5-Pin DIN Line Plug	$A1.00 Each
1		PO4 6A 400V Bridge Rectifier	$A2.25 Each

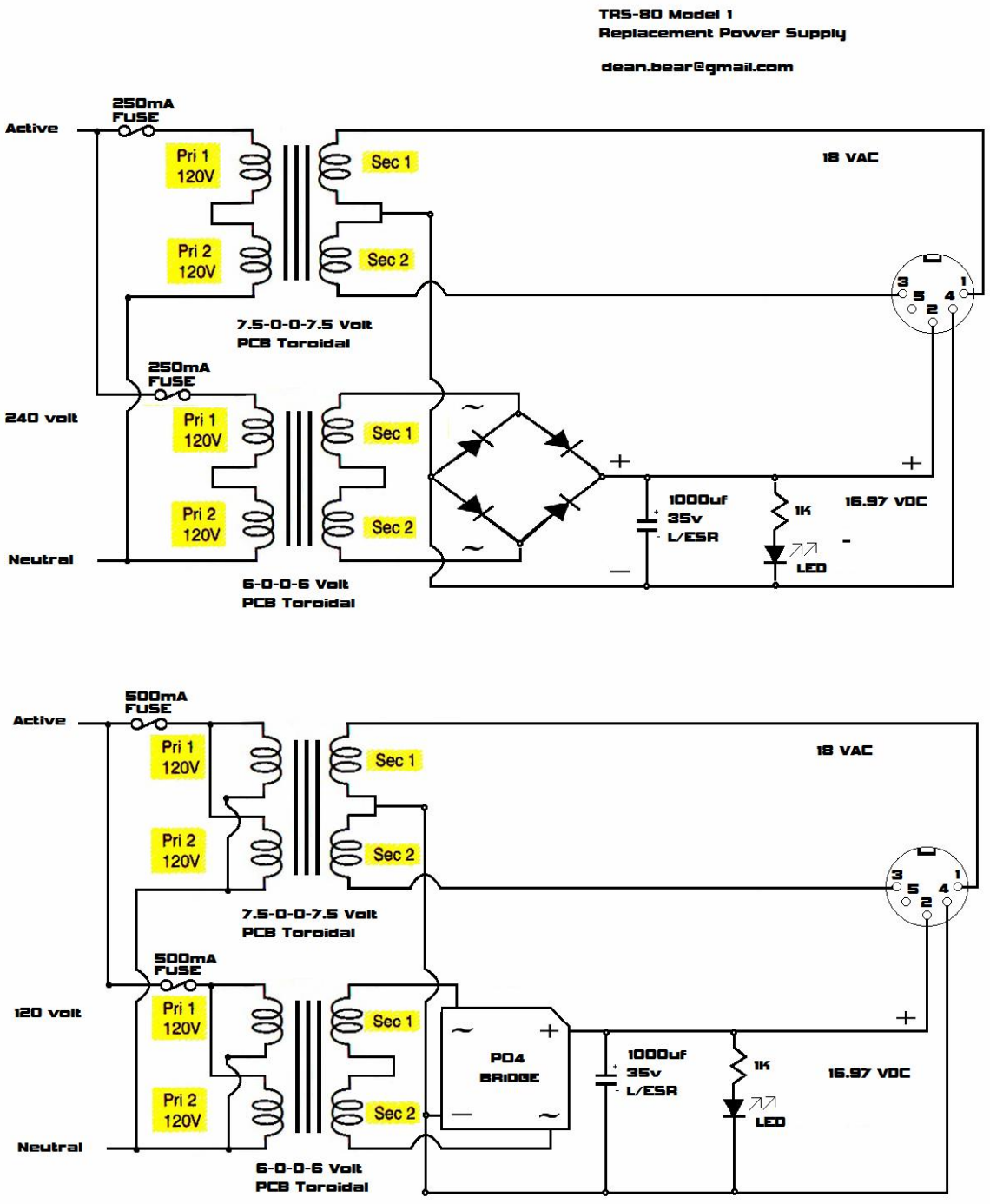
THEORY OF OPERATION

Basically one transformer (a 7.5-0-0-7.5) supplies the 15VAC output and the other transformer (a 6.0-0-0-6.0) supplies a bridge rectifier and an electrolytic capacitor supplies the 19VDC supply. The reason this is done is because the standard single transformer inside an original TRS-80 power supply has two secondary windings to give two different output AC voltages from the 1 transformer.

The bridge rectifier changes the AC output to DC, and the capacitor smooths out the ripple current coming from the bridge down to something palatable to the Model 1. The Model 1 itself, due to its 1970s design, is not very picky about how close to ideal the voltages are supplied to it, it is more concerned about the AC and DC being fed, so as long as 14-17VAC is supplied across pins 1 and 5 of the power connector, and 19-21VDC is supplied across pins 2 and 3, the computer will run.

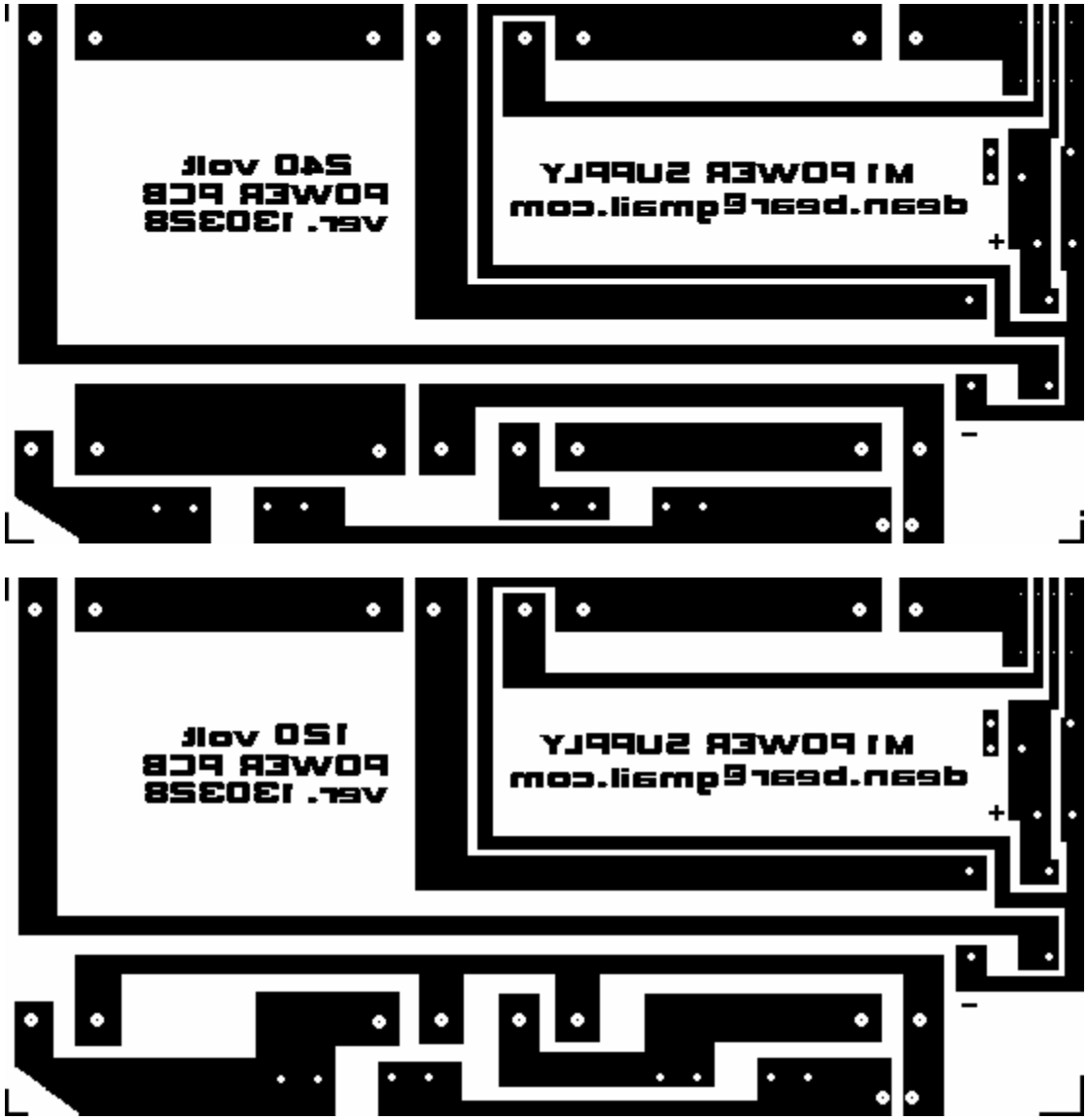
Since the power supply needs to work in either a 120V or a 240V environment, Dean came up with two PCB artworks and two schematic diagrams, the difference being how the primary side of the transformers is connected. On the 240V version the

transformer primaries are wired in series, on the 120V version they are wired in parallel. This allows us to specify the same parts list for both versions, it's only the PCB artwork which is different.



PCB ARTWORK

Scale up this artwork so that it fits onto a 150mm X 75mm single sided copper clad board and etch it any normal way (I use the toner transfer method and etch with good old Ferric Chloride). Note there are two different artworks, the top one is for the 240V design and the bottom one is for the 120V design.



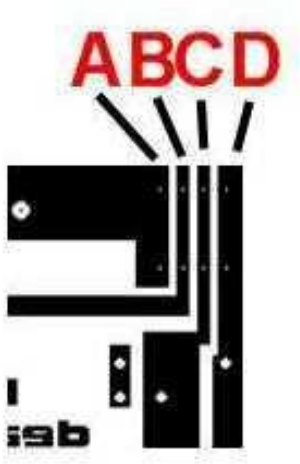
ASSEMBLY

The 7.5-0-0-7.5 transformer goes in the centre - over the area which reads: M1 POWER SUPPLY, and the 6-0-0-6 transformer goes on the end, over the area which reads: 120V or 240V POWER PCB. Pay attention to the polarity of the bridge and the capacitor, and the LED if you are installing it (it's optional but a nice touch).

TESTING

Prior to plugging it into a TRS-80, look for the correct voltages on the output connector, which is around 15-17VAC

across points A and B, and 17-19VDC across points C and D (D being ground and C being 17-19VDC). Make up your power cables to the TRS-80 and test one last time, and you should be in business.



MOUNTING IN THE EXPANSION INTERFACE

Dean noted if you remove the black power supply holder from the EI, some common sense will show that a small amount of carpentry is needed to mount the power supply in the compartment.



NEXT TIME: I talk about FreHD - the Hard Drive Emulator for the Model I/III/4/4P.

ianm@trs-80.com

IVAN KENNEDY'S MODEL IIIs

by

Ian Mavric, contributing writer

This article first appeared in SYDTRUG News.

Back in February Ivan emailed me saying he had a number of old TRS-80 systems which he would like to get fixed, and asked if I was up to the task. After a few more details were sent back and forth between Ivan and I, we made plans for a visit to my workshop with a Subaru Forester full of TRS-80s. The actual count was 6 Model 4s, 3 Model IIIs, a Model 1 system, a Model 4P portable TRS-80, a Tandy 200 notebook, and boxes or spare parts and a small consignment of software. My briefing was simple: fix as many as I could before Ivan had to return to Sydney two weeks later.

I first had a look at the Model III computers, two looked complete, a 16K cassette-based machine and a very rare 48K disk-based machine with the optional factory high-resolution graphics kit (cat. no 26-1125) installed. For those who can't remember, the graphics option was super expensive, $549 in 1982, and not many people bought it because there was no software support for it. The few computers which have the graphics option fitted are considered to be priceless restore-at-all-costs relics, so to me this one was a must-fix. Checking with Ivan, I found that this computer was owned by Graham Read and had not been in service since 1995.

First I thought I would deal with the easy machine, the 16K, and had a look inside to find it surprisingly clean. Nothing looked out of place so I plugged it in and was soon greeted with the familiar Cass? prompt and flashing block cursor. The picture was shaky but the machine seemed to work quite well, despite a few unresponsive keys. I made a list of the non-responsive keys and powered the machine down. Shaky picture on this type of computer is usually the result of ineffective ripple current filtering in the power supply and it didn't take long for me to find the culprit, a bulging electrolytic capacitor. I replaced it as well as the A/C line filter caps, re-installed it in the computer and was pleased to see a rock-solid picture. After fixing the keys I tested the machine further by loading some software the old fashioned way off cassette and was pretty pleased with my work. The computer had been stored in direct sunlight at some time in its life so both Enter keys were yellow, and it had also lost its Model III nameplate. After bleaching the Enter keys finding a spare Radio Shack TRS-80 Model III Microcomputer nameplate the computer was finished. Not a bad start I thought.

Next I had a look at the hi-res Model III and jeez they are

complicated beast. The hi-res board mounts to an aluminium chassis which sits over the motherboard. It's all a very tight fit inside the cabinet. Graham had modified his Model III with extra cables coming out of the hi-res board. I wasn't entirely sure what they were for but I'm guessing to drive an external monitor... perhaps for classroom presentations or the like. The cables were a mess and once I had removed them all I had the hi-res board removed from the machine. At this point it's just a regular Model III 48K disk system, which I powered up and it was in a very unhappy state. Model IIIs have 24 16Kx1 bit Ram chips arranged in three banks of 8 chips, each bank providing 16K of memory. After some swapping and changing, I ended up throwing out 14 bad Ram chips, replacing them with good used one and finally after some hours was greeted with Cass? on the screen. The power supplies both (Model III disk systems have dual 35W power supplies in them) had blown A/C line filter caps which were changed, but the disk drives were both unhappy. I sourced a couple of good used disk drives from Ivan's collection of spare parts and before long the computer was booting up to TRSDOS 1.3 and running memory test routines. Enough work for one afternoon so I called it a day.

The next day I ran memory testing routines for most of the day while I restored the trace cuts and re-wiring Graham had done to the hi-res graphics board. The moment of truth was when I attempted to re-connect the hi-res board and loaded up Tandy's Service Centre diagnostic disk which has test patterns for the hi-res board. A pattern loaded up, it was high resolution, but none of the test patterns looked remotely like they should. A closer look at the patterns that did show up displayed alternating black and white blocks interspersed with sections of what the pattern should have been. Knowing that the memory on the hi-res board is "visible" memory, white blocks indicate a bit stuck high and a black block represents a bit stuck low. Given the poor state of the main RAM in the Model III it was reasonable to suspect the RAM on the hi-res board suffered the same fate, and so replacing the memory chips on the hi-res board cured the problem and displayed the proper test pattern.

I put the computer back together and was working on it a bit when I decided the brightness needed to be turned up a little, so I reached under the computer where the brightness/contrast controls are, only to find the brightness control stuck solid. I've seen this before, where the control just seizes-up inside, and there is little you can do other than replace the control with a good used one. Once again the lid was removed from the computer and the faulty brightness control replaced. Satisfied everything had been done to make this computer happy again I put it together one last time and that is how it is now at Ivan's.

The last Model III? It was missing many parts, had no disk drives and no power supplies installed, it also had major scratches in the silver paintwork on one side and so I pretty quickly realised this one was going to be a parts machine to fix other TRS-80 Model IIIs. Key switches from it were used to fix the 16K cassette machine, and RAM chips, brightness control, and various other screws etc. were used to fix the hi-res Model III. Some parts were used to fix the Model 4, which I will talk about next time.

Ian.

ianm@trs-80.com





INNOVATIVE
TRS-80 SOFTWARE
FROM THE PROFESSIONALS



We are pleased to be able to announce the commencement of a new series of Adventure games. The series named "Mysterious Adventures" is written in machine language by B. Howarth, an English author. The first episode is entitled "The Golden Baton". The scenario is that you have been sent by the ruler of your own land to a strange province with the mission of discovering the whereabouts of the legendary Golden Baton of Ferrenuil, King of the Ancient Elf Kingdom. The baton mysteriously disappeared several years ago and whilst others have ventured to the land in an attempt to discover it, none have returned to tell their tale!

The program follows what has become the normal structure for Adventure programs. Like the original main frame Adventure, directions can be designated by just the first letter of the compass point and commands may be optionally entered with just the first three letters of the appropriate word. As usual provision is made for saving the game at any stage and such standard commands as Help, Inventory, Score and Quit are all available. Experienced adventurers will inevitably draw comparisons between this series and that of Scott Adams, so we will leave it to them to make their judgements! The only comment that we will make at this time is that we find it quite invigorating to play an Adventure game by a different author as obviously they construct their stories slightly differently. Mysterious Adventure 1, "The Golden Baton" is available on cassette for TRS-80 or Video Genie machines of 16K or more and on disk for 32K up machines. It occupies a full 16K. The tape versions save their game to tape and the disk to disk.

Tape version .....£8.75 Disk version £11.00

Both prices **plus** VAT and 75p P. & P.

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 [**refundable**] plus 50p postage.

A.J.HARDING (MOLIMERX)

MOLIMERX LTD.

1 BUCKHURST ROAD & 28 COLLINGTON AVENUE,
BEXHILL-ON-SEA, EAST SUSSEX.

TEL: [0424] 220391 **TELEX 86736 SOTEX G**

BARCLAYCARD VISA

TRS8BIT



WELCOME TO THE 2013, XMAS EDITION.
DEE AND I WISH YOU ALL A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR.
THANK YOU ALL FOR YOUR HELP, SUPPORT, CONTRIBUTIONS AND ENCOURAGEMENT THROUGHOUT THE YEAR.

SPECIAL THANKS GO TO MAV, DOWN THERE IN OZ, FOR ALL HIS CONTRIBUTIONS OVER THE YEAR. THEY ARE WRITTEN IN SUCH A PROFESSIONAL MANNER AND SHOW SUCH KEEN INTEREST AND ENTHUSIASM FOR ALL THINGS TANDY. I JUST CAN'T WAIT UNTIL THE NEXT ISSUE TO SEE WHAT HE'S COME UP WITH!

THANKS ALSO TO THOSE OF YOU WHO CONTACTED ME REGARDING THE M1 INTERNAL UPGRADE USING 4164 CHIPS. IT SEEMS THERE WAS, AT LEAST, 2 VERSIONS AND DETAILS OF THE ONE IN CQ MAGAZINE, SENT TO ME BY LARRY KRAEMER, ARE IN THIS ISSUE. IT ALSO REMINDED LARRY OF A MEMORY CHECKING PROGRAM WHICH FIRST APPEARED IN 80MICRO. HE REMEMBERED AMENDING IT FOR THE MODEL 3, SO, IN TRUE TRS8BIT FASHION, BOTH THE PROGRAMS ARE FEATURED IN THIS ISSUE.

LARRY ALSO SENT DETAILS OF REPLACING THE INTERNAL ROM'S WITH EPROMS. WITH HIS PERMISSION AND SUPPORT, DETAILS SHOULD APPEAR IN THE MARCH 2014 ISSUE. YEH, STUFF ALREADY IN THE PIPELINE FOR OUT 8TH YEAR!!

DUE TO THE SIZE OF THIS ISSUE, I'LL KEEP THESE NOTES ON THE SHORT-SIDE. A FULL CONTENTS LISTING, JUST FOR THIS ISSUE, NOW APPEARS PAGE 2.

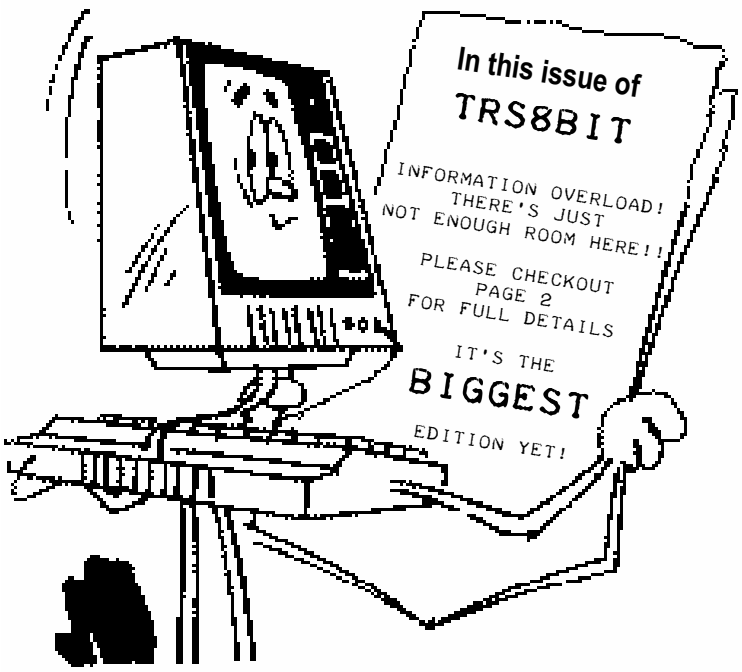
THE UK'S EBAY HAS BEEN REASONABLY BUSY OVER THE LAST 3 MONTHS, MUCH OF IT WITH VERY SILLY PRICES THOUGH, (IE £50 FOR A CASSETTE GAME!), HOWEVER, A VIDEO GENIE FETCHING £160 WAS NICE TO SEE.

WELL, AT 56 PAGES, THIS IS THE BIGGEST, AND I HOPE YOU'LL AGREE, BEST EDITION YET. PLEASE KEEP SENDING IN ANY TANDY RELATED ARTICLE FOR WHAT ID ABOUT TO BECOME OUR 8TH YEAR



BYE FOR NOW AND HAPPY CHRISTMAS EVERYONE

DUSTY



CONTENTS

PAGE 3 WATERMARKING BASIC PROGRAMS
AJ HARDING
THIS ARTICLE FIRST APPEARED IN THE MOLIMERX CATALOGUE AND IS A SIMPLE WAY OF PROTECTING YOUR BASIC PROGRAMS.

PAGE 5 IVAN KENNEY'S MODEL 4
IAN MAVRIC
MAV FIRST WROTE THIS ARTICLE FOR SYDTRUG NEWS. JUST HOW DOES HE SORT OUT ALL THOSE PROBLEMS?

PAGE 8 ASSEMBLER NOTES - PART 4
LAURIE SHIELDS
THE FINAL PART OF LAURIE'S BRILLIANT INTRODUCTION TO ASSEMBLER PROGRAMMING. LAURIE HAD KINDLY GIVEN PERMISSION FOR THE 4 PARTS TO BE MADE INTO A BOOKLET FOR DOWNLOADING. IT WILL BE AVAILABLE FROM THE WEBSITE (TRS-80.ORG.UK, JUST IN CASE YOU'D FORGOTTEN) IN THE NEW YEAR.

PAGE 24 IN MAV'S WORKSHOP
IAN MAVRIC
THE NEXT INSTALMENT OF MAV'S HARD DISK SERIES. THIS WILL MAKE ALL US MODEL 1 FANS SMILE.

PAGE 30 IT'S CHRISTMAS!
DEE AND I WISH YOU ALL A VERY MERRY CHRISTMAS

PAGE 32 UPGRADING A MODEL 1 USING 64K CHIPS
LEW MCCOY & FRED TREASURE
AN EXTRACT FROM CQ MAGAZINE

PAGE 34 VIRTUAL FLOPPY DISK USING A RASPBERRY PI
ALAN PAGE
ALAN 'BLOGS' DETAILS OF A MODEL 1 UPGRADE

PAGE 39 MISSILE DODGE
AUTHOR(S) UNKNOWN
A TRULY BRILLIANT ONE-LINER WITH FULL EXPLANATION. IT'S JUST HAS TO BE AN XMAS DAY SPECIAL TREAT!

PAGE 42 ASK MAV - HARDWARE QUESTIONS ON THE MODEL 4
IAN MAVRIC
MAV ITEMISES SOME OF THE M4 QUESTIONS HE'S BEEN ASKED. HOPEFULLY, WE CAN MAKE THIS A REGULAR FEATURE FOR ALL TYPES OF HARDWARE.

PAGE 44 GOODBYE HARD DISK, HELLO FREHD
GARRY HOWARTH
GARRY FINDS A NEW LEASE OF LIFE FOR HIS OLD HARDWARE, USING THE LATEST UPGRADE

PAGE 48 MAV'S BELIEVE IT OR NOT
IAN MAVRIC
MAV COMES UP WITH A REAL MODEL 1 'FIND'

PAGE 50 BABYROOT
DENNIS BATHORY KITSZ & LARRY KRAEMER
THE ORIGINAL ARTICLE BY DENNIS APPEARED IN 80MICROCOMPUTING AND WAS AMENDED BY LARRY FOR THE MODEL 3

WATERMARKING BASIC PROGRAMS

AJ HARDING
I BELIEVE THIS ARTICLE FIRST APPEARED IN THE MOLIMERX CATALOGUE

YOU WOULD BE SURPRISED AT THE AMOUNT OF FLAK I GET FROM CUSTOMERS FOR DEVOTING TOO MUCH TIME TO TAPE OWNERS AND CONTRARY WISE TO DISK OWNERS. THEREFORE, I WILL TRY AND CONCENTRATE ON ONE MATTER WHICH APPLIES TO BOTH AND IS, I ALWAYS THINK, A RATHER GLAMOROUS TIP!

IT IS A METHOD WHEREBY A LINE CAN BE HIDDEN IN THE PROGRAM SO THAT WHEN THAT PROGRAM IS LISTED A FALSE LINE IS SHOWN. IN FACT, ANY OF YOU WHO BOUGHT OUR VERY EARLY PROGRAMS - CIRCA 1978/79 - MAY HAVE COME ACROSS THE ANOMALY WHICH I AM GOING TO DESCRIBE, BECAUSE AT ONE TIME WE DID USE IT AS A METHOD OF PROTECTING OUR SOFTWARE. IT DOES NOT IN ANY WAY HINDER THE RUNNING OR COPYING OF A PROGRAM. WHAT IT DOES DO - IF IT IS NOT SPOTTED BY THE USER - IS TO SUCCESSFULLY PROVE BEYOND ALL DOUBT THAT THE PROGRAM EMANATED ORIGINALLY FROM A PARTICULAR SOURCE. IN OTHER WORDS, IF SOMEBODY ILLEGALLY COPIES A PROGRAM, THEN IF IT CONTAINS THIS WATERMARK THE TRUE OWNER CAN ALWAYS PROVE THAT A COPY WHICH HE COMES ACROSS LATER IS ILLEGAL. SOMEWHAT LIKE SHUTTING THE STABLE DOOR AFTER THE HORSE HAS BOLTED, BUT IN ONE COURT CASE WE USED IT SUCCESSFULLY, SO IT IS WORTH WHILE.

AS I HAVE SAID THE OVERALL CONCEPT IS THAT A LINE IS INSERTED IN THE PROGRAM WHICH IS OVER-WRITTEN BY A NONSENSICAL LINE. IT IS THIS LATTER WHICH APPEARS ON A VDU DISPLAY. UNFORTUNATELY THE ANOMALY SHOWS UP ON A LINE PRINT, BUT ONE CANNOT HAVE EVERYTHING. THE BEST WAY OF EXPLAINING HOW TO DO THIS IS TO ACTUALLY CARRY IT OUT ON A COMPUTER. IT DOES NOT MATTER WHETHER IT IS A TAPE OR A DISK MACHINE, NOR WHETHER IT IS MODEL I OR MODEL III. IT ALSO WORKS ON THE VIDEO GENIE AND GENIE I. I WOULD SUSPECT ALSO ON THE GENIE 11, BUT I HAVE NEVER TRIED IT.

SO SWITCH ON YOUR MACHINE AND GO INTO BASIC. LOAD ANY PROGRAM THAT YOU WANT, OR IF YOU PREFER SIMPLY CARRY OUT THE INSTRUCTIONS FROM THE KEYBOARD. WHICH EVER COURSE YOU ADOPT WE ARE INTERESTED IN INSERTING A LINE, SO FOR THE SAKE OF ARGUMENT WE WILL CALL IT LINE 10, BUT OF COURSE IT MAY BE ANY NUMBER. IF YOU ARE INSERTING INTO AN EXISTING PROGRAM THEN, OF COURSE, MAKE SURE YOU ARE NOT OVER-WRITING AN EXISTING LINE NUMBER.

BEFORE YOU ACTUALLY START TYPING YOU MUST DECIDE ON WHAT WATERMARK YOU WANT. IT IS BEST TO USE SOME SORT OF INSCRIPTION FOR OTHERWISE IF ONE GOES THROUGH MEMORY WITH A MONITOR THEN WHATEVER THE WATERMARK IS WILL QUICKLY SHOW UP. WE USED TO USE THE ASCII NUMBERS REPRESENTING THE FIRST FOUR LETTERS OF MY NAME. HARD IN ASCII IS 72 65 82 68. FOR THE SAKE OF THIS DEMONSTRATION YOU MIGHT AS WELL USE THE SAME, BUT YOU CAN USE ANYTHING ELSE IF YOU LIKE. AFTER YOU HAVE TYPED THE LINE NUMBER AND SPACE, TYPE REM. THIS MAKES CERTAIN THAT THE PROGRAM WILL IGNORE THE LINE, FOR YOU MUST

REMEMBER THAT IT IS THE LINE THAT WE ARE NOW ENTERING THAT WILL ACTUALLY FORM PART OF THE PROGRAM, IN OTHER WORDS IF IT WERE A VALID COMMAND THE PROGRAM WOULD RECOGNISE IT AND ACT UPON IT. SO WE TYPED REM AND CONTINUE, WITH OR WITHOUT A SPACE, TO THE ASCII REPRESENTATION MENTIONED ABOVE. HIT THE ENTER KEY AND WE WILL NOW HAVE THE FOLLOWING:
10 REM 72658268
THIS, THEREFORE, IS THE WATERMARK AND WE MUST NOW PROCEED TO COVER IT UP.

TYPE EDIT 10. AS YOU KNOW THE LINE NUMBER WILL APPEAR ON THE NEXT LINE. FROM NOW ON YOU MUST BE A BIT CAREFUL AND BEFORE PROCEEDING IT WOULD BE WISE TO DECIDE UPON THE CONTENTS OF THE MASKING LINE. I ALWAYS USED TO LOOK AT THE GENERAL CONTEXT OF THE PROGRAM TO MAKE SURE THAT THE LINE DID NOT STAND OUT. TRY AND AVOID, FOR INSTANCE, THE OPENING OF A LOOP BECAUSE YOU WILL NOT BE CLOSING IT. ANYWAY THIS I MUST LEAVE TO YOU.

FOR THE PURPOSE OF THIS DISCUSSION WE WILL ASSUME THAT OUR MASK IS GOING TO BE IF X= 1 THEN 500. WE LEFT THE PROCEDURE WHERE WE HAD THE LINE NUMBER AND THE MACHINE WAITING FOR ENTRY. PRESS THE X KEY. THIS WILL TAKE YOU TO THE END OF THE LINE, PRESS THE SHIFT KEY AND THE '. IN OTHER WORDS WE ARE MAKING ANOTHER REM STATEMENT. KEEP YOUR FINGER ON THE SHIFT BUTTON AND USING THE LEFT HAND ARROW, BACK SPACE TO THE BEGINNING OF THE WATERMARK, THAT IS TO SAY, TO THE R IN REM. NOTE THAT THE CHARACTERS HAVE NOT BEEN ERASED AS YOU BACK SPACE, BUT NOW THAT YOU HAVE GOT TO THE R IT HAS DISAPPEARED. RELEASE THE SHIFT KEY AND TYPE IN YOUR MASKING LINE. AS YOU DO THIS THE WATERMARK LINE WILL DISAPPEAR. WHEN YOU HAVE GOT TO THE LAST 0 OF 500, PRESS THE ENTER KEY AND YOU ARE FINISHED.

TO TEST IT, DO A LIST AND YOU WILL SEE THAT THE MASKING LINE IF X ETC. IS LISTED. THAT LINE HAS NO RELEVANCE TO THE PROGRAM WHATEVER.
IT WILL HAVE NO AFFECT. SO FAR AS THE DISPLAY IS CONCERNED THE ONLY WAY OF FINDING THE WATERMARK IS TO EDIT THAT LINE AGAIN, SO TYPE EDIT 10 AND JUST FOR FUN HIT THE L KEY FOR LIST. YOU WILL SEE THAT YOU STILL HAVE THE MASKING LINE. NOW, HOWEVER, SPACE THROUGH THE LINE WITH THE SPACE BAR AND YOU WILL SEE THE UNDERLYING WATERMARK APPEAR. AND THAT IS HOW IT IS DONE. A SIMPLE, EASILY INSTALLED, BUT NO LONGER LITTLE KNOWN GIMMICK OF THE MICROSOFT INTERPRETER.

ONE FINAL WORD. IF YOU INTEND TO USE THIS GIMMICK SERIOUSLY, MAKE THE LINE AS SHORT AS POSSIBLE FOR IN FACT AS IT IS LISTED IT DOES DISPLAY THE WATERMARK BUT THIS IS SO QUICKLY OVER-WRITTEN BY THE MASK THAT YOU DO NOT NOTICE IT. IF YOU HAVE A LONG WATERMARK AND, THEREFORE, A LONG MASKING LINE THERE CAN BE A SLIGHT IMPERCEPTIBLE FLICKER WHICH, TO THE KNOWLEDGEABLE, WILL BE A DEAD GIVE AWAY.

--== o0o ==--

IVAN KENNEDY'S MODEL 4

by

Ian Mavric, contributing writer

This article first appeared in the Oct 2013 edition of SYDTRUG News

Back in February Ivan emailed me saying he had a number of old TRS-80 systems which he would like to get fixed, and asked if I was up to the task. Among the initial delivery was 6 Model 4s. My briefing was simple: Ivan wanted a good working Model 4 to return to Sydney with.

The Model 4s dropped off had clearly been in heavy usage by passionate enthusiasts at Sydtrug. It was explained that some had belonged to Greg Read, some from Ivan, at least one was from a local airbase, but something thing was clear, they had all been modified to keep running and some had been ratted for parts to keep others alive. Some were missing disk drives, others were missing power supplies or motherboard, and others we missing even more. Rather than talk about each machine individually, I'll talk about how I assess a machine's suitability for repair, restoration or junking.

First thing to do was to set up each machine, take the lids off and assess what was inside, and also take stock of what would potentially make a nicely restored machine. Things which get noted at this point in time:
- Case condition (is it damaged or will it come up nice after cleaning? has it been modified?)
- Serial number and Catalogue number (low serial numbers are generally more desirable and the catalo number helps determine what the original configuration of the Model 4 was - 26-1067 is a 16K diskless machine, while a 26-1069 is a 64K dual drive Model 4)
- Does the CRT have screen burn (it makes sense to use a CRT without annoying images burnt into it)
- Which version video board is installed (the later digital board gives a clearer image than the early analogue board)
- What is the keyboard like (yellowed? missing keys? missing its RAM size badge?)

Next thing I do is put power to a system which looks the most complete. In most cases with Ivan's computers, the mains filter capacitors had blown, so at this point I removed a power supply which looked pretty good and replaced its filter caps (C1, C2 and C12) and found myself greeted with a happy Cass? prompt from one of the motherboards. I earmarked this system to be one to fix for Ivan to return to Sydney with. Once at this point its easy to test the CRTs just be plugging them in and seeing what sort of image you get. Three showed a

decent image (two had green CRTs and one had a B&W CRT), two showed nothing at all - no signs of life - and one had arcs and sparks going on - what I call signs of life... just the wrong ones. A nice case with a clear green CRT was chosen for Ivan.
At this point one can plug in all the keyboards one after the other and make notes about which keys don't work. I was fortunate that in addition to all the keyboards in the machines, Ivan supplied me with an additional, new/unused keyboard from a US TRS-80 Model 4D. A quick test showed it to be perfect in way so it was given the nod to go in Ivan's pride-and-joy. With a sharp CRT, new board, and re-capped power supply, I was on the way to putting together a nice machine.

Finding good disk drives can be a deal breaker for people doing their own TRS-80 restorations. The Model III and 4 come with Texas Peripherals disk drives, which are a cost-reduced version of the Tandon TM100-1A disk drive. Cost-reduced meaning they are cheaply made, and while most Tandon TM100-1A and 2A disk drives I find are still working, the cheaper TPs don't stand the test of time very well. In this instance though Ivan supplied a large quantity of disk drives. Lots had markings like "formats but doesn't read" or "worked 1/4/87" etc. but I did eventually find some disk drives which worked well and just needed some cleaning and a little re-calibration. I installed a beige Teac FD-55BR 40 track double sided unit is drive 0 and a black Teac FD-55GFR 80 track double sided unit as drive 1.

Once I'm at this stage where the hardware has essentially been selected and tested, the next step, putting it all back together, is where the fun starts. Tandy made a really excellent set of Radio Shack Service Centre diagnostic disks which tests all facets of the machine. It is THE best program for testing the memory and located some bad memory chips in the upper 64K of this 128K machine. TRS-80s don't extensively test their memory, so a machine can appear OK but may crash on certain programs and run fine with others, if they reside in different areas of the memory. Only a proper test program can find them out.

With the computer together and booting regularly I was pretty pleased with the results but given the age of the machine I ran some diagnostics on the floppy disk controller which gave a pretty ordinary result. This happens a lot with old systems as parts degrade over time. Replacing the FDC on a Model 4 necessitates removing the motherboard, which I did, and replace the FDC and ran the tests again and got similarly ordinary results. Fiddlesticks! I noticed the results change on the test program and scope readouts as I leaned on the

white interconnect cable which joins the FDC to the motherboard. Eureka! I changed the flaky white cable and the problem went away. In no time it was running MultiDos, Ivan's favourite OS, and the computer was finished off ready to return to its happy owner.

Ian.



Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.



ASSEMBLER NOTES, Part 4

Laurie Shields

Chapter 8

Getting it from the Keyboard

Now that we are all at home putting it on the video, the next step is to get it from the keyboard. We got an inkling of the possible complications earlier and now it's for real.

Prising the cap of one of the keys (only if you have an early TRS-80) is the nearest you will ever come to seeing a 'bit'. The keyboard is logically connected to a block of memory and each key, arranged in sets of eight, makes or breaks a little contact. This sets or clears one bit of the eight that make up the byte at its memory address. It all sounds very complicated and believe me decoding the keyboard is no game for amateurs. Fortunately we don't have to as Tandy have provided 2BH for us to save all the trouble.

Now saving us all the trouble is perhaps not quite as you would express it if you knew what was coming, as all that Tandy have given us is a very fundamental call that just instantaneously scans the keys and returns with the value of whatever key was being pressed. If we weren't quick enough with our fingers then hard luck and what's more we have to save DE & IY first. That's the sum total of the official machine code keyboard input information.

Let's set ourselves the problem of accepting, from the keyboard, a string of characters that we need to save for use later in our program. As the characters are keyed in we will arrange to display them on the video as that isn't automatic either. We must consider carefully just which keys on the keyboard we are prepared to recognise. As apart from the normal visible characters from a space (value 20H) all the way through the numbers (30H to 39H) and the alphabet both upper and lower case, there are the arrow keys - penny plain and two pence shifted. If we don't set our thoughts out carefully at the beginning our programming will have some unpredictable effects.

This Basic program will demonstrate the decimal value returned by these keys:

```
10 X$ = INKEY$
20 IF X$ = "" THEN 10
30 PRINT ASC(X$);
40 GOTO 10
```

With the exception of the up-arrow by itself, which returns a value of 91 as it is a displayable up-arrow character, all of the others including <Clear> and the big white one give values less than 32. These are known as the various control codes for the output devices, such as the video or a printer, and do weird things like clearing the screen, or throwing a page, or blanking out a line, etc. Handling all of that first time is just too much, so to keep things relatively simple we shall restrict our input so that it recognises just the back-arrow (value 8) as a backspace/delete and the Enter key as the end of input.

As we want to retain the input, assume it's the name of the person who is playing our 'Vacuum Deserters' game, we need to allocate some memory space for storage. As soon as we do this however we are then faced with making sure that the number of characters keyed in does not exceed the size of this buffer. Just to keep up the excitement we must also provide a prompt on the video such as Enter Name :, we must make sure the rest of the line has been blanked out before we ask for keyboard input and last but not least make sure the cursor is switched on! Don't laugh, just see Zap 22 in Apparat's Newdos80 version 2 for their Edtasm fix.

A crafty peek at your Leve1 2 manual gives us the three very useful numbers 14, 30 and 31. The value 14 will switch the cursor on, 30 will clear from the current cursor position to the end of the line and 31 will clear from the cursor to the end of the video. We will use just 14 and 30 for the time being.

Taking our previous routine we can easily alter it to display our new message "ENTER NAME :" but lets add the 14 and 30 to the message and see what. happens. We may as well add the storage buffer as well, which introduces us to a new pseudo-op DS or DEFS for Define Storage. This must be followed immediately by the number of bytes to be reserved so let’s fix it now at say 25 bytes - just under half a line.

Now we get the tricky bit. We will use the C register to hold the maximum number of characters to key in and HL will have the address of the buffer, so it goes like this:
Set B reg. equal to zero, call keyboard and wait for key to be pressed. If it’s a CR jump to end of the routine. If it’s a backspace, deal with it. If it’s a character we are prepared to accept (if not ignore it), is there room in the buffer for one more and if so put it there. Display it and go back for another one.

```
1                       ORG 7800H
2                       LOAD 7800H
3                       CR: EQU 13
4                       BS: EQU 8
6 7800 213D78               LD HL, MESS1
7 7803 CD6578               CALL VIDEOUT
8 7806 214C78               LD HL, BUFFER
9 7809 OE19                 LD C, 25; Length of BUFFER
11 780B 0600    TEXTIN:     LD B, 0; Character count
12 780D CD7B78  KEYIN:      CALL KEYSCAN
13 7810 B7                  OR  A
14 7811 28FA                JR Z, KEYIN
15 7813 FEOD                CP  CR
16 7815 2825                JR Z, TEXTEND
17 7817 FE08                CP  BS
18 7819 2000                JR NZ, CHECK
19 781B 05                  DEC B   ; If lst key, makes B
                                      negative
20 781C FA0B78              JP M, TEXTIN ; so back to
                                         ; beginning
21
22 781F CD7178              CALL OUTPUT ; Display backspace
23 7822 2B                  DEC HL      ; Go back one
24 7823 3620                LD (HL)," " ; clear character in
                                          Buffer
25 7825 18E6                JR KEYIN 26
27 7827 FE20    CHECK:      CP 32       ; If less then 32 dec.
28 7829 38E2                JR C, KEYIN ; we don't want it
29
30              ; We now need A reg to check if Buffer full
31              ; so save the character on the stack first
32 782B P5                  PUSH AF
33 782C 78                  LD  A, B
34 782D B9                  CP  C
35 782E 3803                JR C, NOTFUL    ; Carry if B is
                                              less than C
36 7830 F1                  POP AF      ; Restore the stack
37 7831 18 DA               JR  KEYIN   ;Wait for CR or BS
38
39 7833 F1      NOTFUL:     POP AF      ; Restore character
40 7834 77                  LD (HL), A  ; Put in Buffer
41 7835 23                  INC HL      ; Bump the pointer
42 7836 04                  INC B       ; Add 1 to the
                                          counter
43 7837 CD7178              CALL OUTPUT ; Put it on the video
44 783A 18D1                JR KEYIN    ; Go for next one
45 783C 00      TEXTEND:    NOP         ; For Breakpoint
46 783D 454E5445  MESS1:    DS  "ENTER NAME :`, 30,14,0
46 7841 52204E41
```

```
46 7845 4D45203A
46 7849 1E0E00
47              BUFFER:     DS 25
48
49 7865 7E      VIDEOUT:    LD A, (HL)      ; As we had
                                              before
50 7866 23                  INC HL
51 7867 B7                  OR  A
52 7868 CS                  RET Z
53 7869 CD7178              CALL OUTPUT
54 786C FEOD                CP  CR
55 786E 20F5                JR  NZ, VIDEOUT
56 7870 C9                  RET
57 7871 D5      OUTPUT:     PUSH DE
58 7872 FDE5                PUSH IY
59 7874 CD3300              CALL 33H
60 7877 FDE1                POP IY
61 7879 D1                  POP DE
62 787A C9                  RET
63 7878 D5      KEYSCAN:    PUSH DE
64 787C FDE5                PUSH IY
65 787E CD2B00              CALL 2BH
66 7881 FDE1                POP IY
67 7883 D1                  POP DE
68 7884 C9                  RET
69                          END
```

The first point to watch with this program is for disc users in particular but it applies in a general way to all others. With the source file growing larger the end is getting very close to 7800H (use Q to ask Zen for the Start & End) and should the end of the file go beyond 7800H some funny things will happen when we assemble. As the object code is generated the Load command will place it in memory from 7800H onwards obliterating the letter part of the source file and, whilst being perfectly understandable it is damned annoying when it happens. So whenever you are using the Load pseudo-op ALWAYS check on memory usage to ensure you are loading into free memory. Free memory being between the end of the source file and the top limit of high memory, everything below the start of the source file is in use, broadly as follows:

Start	Stop	Used by
0000H	3FFFH	Rom, Keyboard, video, Printer etc.
4000H	4350H	(Mod 1)
4450H	(Mod 3)	Level 2 cassette operating system.
4450H		Start of Source Cassette Zen.
		Or for disc
4000H	51FFH	DISC operating system.
5200H		Start of source Disc Zen.

Meanwhile back at the program, TEXTIN starts with setting B to zero and calling the keyboard until a key is pressed. Check first of all if it is the big white one as in that case we can pack up. Next check for a back-arrow as this is the only other key less than 32 that we will accept. Now if it is a back-arrow there are two possibilities, either it is the first keystroke and there isn't anything there to rub out or it isn't and there is. We will decrement B anyway and see what happens.

If B was zero to begin with then we dropped a clanger and it has gone minus so the simple way to recover is to jump back to the very beginning of TEXTIN and start afresh. Note we have to use JP M, instead of JR C, as just DECrementing a register does not trigger the Carry flag. If B hasn't gone negative then there must be a character to delete, so first of all CALL OUTPUT to do it on the video and then we must move the value in HL back one and zap the character in the buffer, before going round again for the next character.

That has disposed of the keys less than 32 that we are interested in so now to check on what is in the A register and simply ignore anything less by going back to KEYIN.
OK we have a valid character but have we got room in the buffer to store it? To find out means doing sums with the A register but we can't do that without losing the important character in A, so the best thing to do is PUSH it on to the stack. Comparing B with C by putting one of them in A first will give the answer in the Carry flag to the state of the buffer.

We now have two things to do and if we get them in the wrong order we are in trouble. The wrong way would be to POP AF to get the character back and then jump back to KEYIN on No Carry if the buffer was full. Why wrong? Well the important carry condition in the F register would be altered when the old values off the stack were popped, so we must decide on whether or not there is room in the buffer before Popping AF.

If there is room, then we put the character into the buffer using the neat instruction load the contents of the memory address in HL with the A register. Increment the value in HL to point to the next available space, add one to B as our character count and go back for the next character. Nothing to it really, so assemble it and jump to 7600H with breakpoint at 783CH to see it all work.

We can of course tidy up TEXTIN to a little subroutine that simply needs HL and C initialising to come back on return with the string input into the buffer and the number of characters in B:

```
    TEXTIN: LD B,0
    KEYIN:  CALL KEYSCAN
            OR A
            CP CR
            RET Z; This is the exit from the routine.
            CP BS
            JR NZ, CHECK
            DEC B
            JP M, TEXTIN
            DEC HL
            LD (HL),' '
            JR PUTONVDU
    CHECK:  CP 32
            JR C, KEYIN
            PUSH AF
            LD A, B
            CP C
            JR C, NOTFUL
            POP AF
            JR KEYIN
    NOTFUL: POP AF
            LD (HL), A
            INC HL
            INC B

 PUTONVDU:  CALL OUTPUT
            JR KEYIN
```

That was the easy one, in case you hadn't noticed, now it gets a little bit trickier as the next problem is inputting numeric data. Instead of any character from the keyboard we now only want to accept the digits 0 to 9, but once having keyed them in convert the digits in decimal format to a numeric quantity that the Z80 can understand. Actually it's not going to be that bad considering what we've already been through.

First of all we can re-use most of the preceding program for reading the keyboard with some additions and alterations to label names etc. The first change is to reject any characters less than '0' with CP '0' rather than CP 32; we must also add an extra test to reject anything bigger than '9' so Check becomes:

```
CHECKNUM: CP '0'
JR C,  KEYIN2 ;The new name for KEYIN
CP '9'+1       ;Anything bigger than a 9
JR NC, KEYIN2
PUSH AF etc
```

This will allow input from the keyboard and only accept the 0 to 9 digits. If you want to use plus, minus and decimal points then you're reading the wrong chapter. Having got them in though they are still just characters and we have to do some nifty footwork to turn them into a useful numeric quantity.

What we shall do is to start with zero and work from left to right through the string of digits multiplying the value we already have by ten and then adding the number represented by the digit to our value. As the final answer could end up bigger than 255 we shall have to use the HL register pair (remember they can do adds) for most of this operation. Let's address ourselves to the problem of multiplying the contents of HL by ten.

Doubling HL is easy as we have the instruction ADD HL,HL and if we did it again we would have four times. If only we had saved the first value we could add that to get five times and then double it again for ten. Well the DE registers have been lazing about doing nothing let's use them:

```
TIMESTEN:   PUSH HL
POP DE      ;DE has original HL value
ADD HL,HL   ; HL is now doubled
ADD HL,HL   ; HL is now four times
ADD HL,DE   ; HL is now five times
ADD HL,HL   ; HL is now ten times
```

Just hold it a minute. HL is doing sums, DE is helping, B knows how many digits to process - who's left to tell us where the buffer is with the characters in it? Time to use IX or IY, I suppose and since we've been pushing and popping IY like there was no tomorrow we'll use that one. Assume IY points to our string of digits - this is what you do:

Get the character pointed to by IY into the A register.
Subtract the 30H from it so that we're left with just 0 to 9
Add it to HL - no that isn't allowed!



The *BEST* in TRS-80s

Call The Right Stuff

Ask for Ian
The number is +61 416 184 893

That's The Right Stuff
And he's in Melbourne







http://ianmav.customer.netspace.net.au/trs80/

Add L to A, load L with A, if Carry then increment B
That is all just too messy! Try it this way:

Load the E register with A, make sure D equals zero
Add DE to HL - much tidier.

The only point to watch is that IY never expects to be pointing in the right direction and always thinks that you're going to want the one just to the left or half a dozen to the right like this: (IY-1) or (IY+6). If we really do want the memory address that IY is actually pointing to then we must use (IY+O); so the next bit:

```
    LD A, (IY+O)
    SUB 30H
    LD E, A
    LD D, 0
    ADD HL, DE
    INC IY
```

again for the next digit, so:

```
    DJNZ TIMESTEN
```

This will cause a monumental disaster if for some reason the user had just pressed <Enter> instead of inputting some numbers, as B would start at zero, after the first DJNZ would get the value 255 and we would do the conversion on the whole two hundred and fifty six bytes starting at the buffer address and ending up, who knows where.
We can handle that though by checking on B first; so putting it all together with a prompt, say 'ENTER QUANTITY :' it looks like this:

```
1                       ORG 7800H
2                       LOAD 7800H
3                       CR: EQU 13
4                       BS: EQU 8 5
6 7800 211978               LD HL, HOWMANY
7 7803 CD8678               CALL VIDEOUT
8 7806 212C78               LD HL, NUMBUFF
9 7809 0E05                 LD C, 5
10 780B CD5578              CALL NUMBERIN
11 780E 00                  NOP         ; For breakpoint
                                          halfway
12 780F 212C78              LD HL, NUMBUFF
13 7812 CD3378              CALL CONVERT
14 7815 223178              LD (QUANTITY), HL
15 7818 00                  NOP         ; For Breakpoint 16
17 7819 454E5445 HOWMANY:   DB  'ENTER QUANTITY :',14,30,0
17 781D 52205155
```

```
17 7821 414E5449
17 7825 5459203A
17 7829 0E1E00
18              NUMBUFF:    DS 5
19 7831 0000    QUANTITY:   DW 0
20
21 7833 FDE5    CONVERT:    PUSH IY
22 7835 E5                  PUSH HL
23 7836 FDE1                POP IY
24 7838 210000              LD HL, 0

25              ;Check first if B=0 i.e. no input

26 7838 76                  LD A, B
27 783C B7                  OR  A
28 783D 2813                JR Z, CONVERTEND
29
30 783F E5      TIMESTEN:   PUSH HL
31 7840 D1                  POP DE
32 7841 29                  ADD HL, HL
33 7842 29                  ADD HL, HL
34 7843 19                  ADD HL, DE
35 7844 29                  ADD HL, HL
36 7845 FD7E00              LD A, (IY+0)
37 7848 D630                SUB 30H
38 784A 5F                  LD E, A
39 784B 1600                LD D, 0
40 784D 19                  ADD HL, DE
41 784E FD23                INC IY
42 7850 TOED                DJNZ TIMESTEN
43 7852 FDE1    CONVERTEND: POP IY
44 7854 C9                  RET
45
46 7855 0600    NUMBERIN:   LD B, 0
47 7857 CD9C78  KEYIN2:     CALL KEYSCAN
48 785A B7                  OR A
49 785B 28FA                JR Z, KEYIN2
50 785D FE0D                CP CR
51 785F C8                  RET Z
52 7860 FE08                CP BS
53 7862 2009                JR NZ, CHECKNUM
54 7864 05                  DEC B
55 7865 FA5578              JP M, NUMBERIN
56 7868 2B                  DEC HL
57 7869 3620                LD (HL)," "
58 786B 1814                JR NUMBVDU
59
60 786D FE30     CHECKNUM:  CP '0'
61 786F 38E6                JR C, KEYIN2
62 7871 FE3A                CP '9'+1
```

```
63 7873 30E2                JR NC, KEYIN2
64 7875 F5                  PUSH AF
65 7876 78                  LD A, B
66 7877 B9                  CP C
67 7878 3803                JR C, NOTFUL
68 787A Fl                  POP AF
69 787B 18DA                JR KEYIN2
70
71 787D Fl       NOTFUL:    POP AF
72 787E 77                  LD (HL), A
73 787F 23                  INC HL
74 7880 04                  INC B
75 7881 CD9278   NUMBVDU:   CALL OUTPUT
76 7884 18D1                JR KEYIN2
78 7886 7E       VIDEOUT:   LD A,(HL) etc.
```

An alternative approach might be to use the original TEXTIN routine for any input and check it for numeric validity before conversion. This would probably produce a more universal solution but would have added to the complexity.
For storing the final answer we have asked for two bytes of memory to be reserved with the pseudo-op DW (Define Word). A Word in Z80 jargon is two bytes (one high order, one low) and in our case we have initialised them at zero. When we test this routine we will discover the peculiar way the Z80 puts the contents of a register pair into memory. In human terms it appears to be back-to-front, i.e. the most significant byte comes second when displayed left to right. A better way to think of it though is to realise that the highest order (most significant) byte goes into the highest address of the two bytes of storage and the lowest (least significant) byte goes into the lowest address.

Right oh - assemble and jump to 7800H with a breakpoint at 7818H (or 780EH if you want to stop half way), key in any number you want and then check with Q7831H on what has been stored at the location labelled Quantity.

Now that we've assembled some programs with a reasonable number of labels in them let's look at something else Zen has got for us. The first is the simple sorted Symbol Table listing which tabulates all the labels we have used and their hexadecimal values:

```
 Label value        Label value         Label value
    Label value

BS 0008          CR OOOD          CONVERT 7833    CONVERTEND 7852
CHECKNUM 786D    HOWMANY 7819     KEYIN2 7857     KEYSCAN 789C
NUMBUFF 782C     NUMBERIN 7855    NOTFUL 787D     NUMBVDU 7881
OUTPUT  7892     QUANTITY 7831    TIMESTEN 783F   VI DEOUT 7886
```

With the later versions of Zen there is a cross-reference listing capability that lists every label in the symbol table with its value, the line number where it is defined and the line numbers of every reference to it in the whole of the source file:

```
 Label      value   occurrence      references
CR          000D        3           50 83
BS          0008        4           52
HOWMANY     7819        17          6
NUMBUFF     782C        18          8 12
QUANTITY    7831        19          14 17
CONVERT     7833        21          13
TIMESTEN    783F        30           42
CONVERTEND  7852        43          28
NUMBERIN    7855        46          10 55
KEYIN2      7857        47          49 61 63 69 76
CHECKNUM    786D        60          53
NOTFUL      787D        71          67
NUMBVDU     7881        75          58
VIDEOUT     7886        78          7 84
OUTPUT      7892        86          75 82
KEYSCAN     789C        92          47
```

Anybody getting this far without the help of Assembler's Aspirin (Pt No 26-XXXX at your local Tandy shop) is entitled to celebrate.

Appendix 1

R O M C A L L S

The following list is simply a catalogue of the Model 1 Rom calls that have come to my attention through various publications and other means. Some of these may be Model 3 compatible and some are bound to be totally different.

Address	Action	Comments
000BH	Where am I ?	

Pops the return address into HL and Jumps to (HL). Used to determine where in memory you are:

```
    CALL 0BH
    HERE:PUSH HL
    POP IX;   IX now has value HERE
```

LOOKING FOR FAST, INEXPENSIVE, UNLIMITED MASS STORAGE FOR YOUR TRS-80 MODEL I/III/4/4P/4D?

The amazing

"FreHD"



-Emulates a TRS-80 hard drive, but faster than any hard drive!
-Works with your favourite DOS (LS-DOS, LDOS, CP/M, Newdos/80 2.5)
-Uses SD card for storage medium
-Bonus free Real Time Clock function!
-Designed in Belgium and proudly built and shipped from Australia
-Kit form or fully assembled

Order yours today:
http://ianmav.customer.netspace.net.au/trs80/emulator

0013H Device Input
Usually used for 1 byte read from disc file. LD DE, File control block of open file CALL 13H; Byte returned in A Zero flag set if no error JP NZ, Disc error routine.

001BH Device Output
Similar to above for 1 byte to disc

```
    :Character in A register
    LD DE, File control block
    CALL 1BH
    JP NZ, Disc error routine.
```

002BH Keyboard scan
Scans keyboard and returns character in A Very similar to Inkey$ from Basic.
Uses DE and IY registers, char in A reg and Zero set if key pressed. Does not hang around waiting, usually used with a loop:

```
    KBSCAN: PUSH DE
            PUSH IY
            CALL 2BH
            POP IY
            POP DE
            OR A
            JR Z,KBSCAN
            RET ; character in A
```

0033H Video Ouput
Outputs character in A to current cursor pos. Uses DE & TY:

```
;character in A
    DISPLAY:    PUSH DE
                PUSH IY
                CALL 33H
                POP IY
                POP DE
                RET
```

003BH Printer Output
Exactly the same as for video but this routine will hang up if printer not connected. On Model 3 the hang up can be broken with <Break>.

0040H Line Input
Keyboard line input terminated with <Entry>. Entry HL points to your buffer, B - No bytes. Exit HL -> buffer and B = No of characters. Uses DE register.

0049H Key Input
Similar to Keyscan but waits for key stroke.

0060H Delay
Decrements BC until = 0 then returns. Uses just BC and AF registers.

0072H Basic Ready
Jump to 72H instead of IA19H. Jumping to 1A19 is OK if you've tidied everything up first. If you haven't and you want to get back to Basic from a System load or other m/c routine 72H does all the housekeeping for you.

01C9H Screen clear
On model 3 all registers altered. Usually switches cursor off on Model 1.

01F8H Cassette Off Switches cassette off.

0212H Cassette On
Switches cassette on. It is probably best to XOR A before calling either of these. I don't know how the second cassette via the expansion interface is addressed. 022CH
Blink Flashes the '•'.

0235H Byte in Reads byte from cassette into A register.

0264H Byte out Outputs byte in A register to tape.

0287H Write sync
Outputs the leader and sync byte to tape. Uses BC registers.

0296H Reads sync
Reads the leader and sync byte from tape.

0314H Address read Reads two bytes into HL. Used for the load address in system tapes.

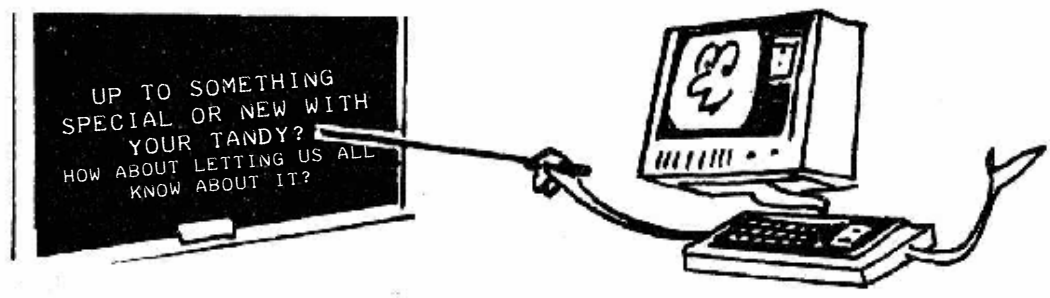
37E8H Printer addr.
Model 1 parallel printer connected here. Model 3 puts printer status here.
For Bit 7 = 0 Not busy.
both Bit 6 = 0 Paper OK.
Model 1 Bit 5 = 1 Device select (whatever that means)
& Model 3 Bit 4 - 1 No fault.

Keyboard Matrix

Address	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
3801	@	A	B	C	D	E	F	G
3802	H	I	J	K	L	M	N	O
3804	P	Q	R	S	T	D	V	M
3808	X	Y	Z					
3810		!	"	#	$	%	&	`
	0	1	2	3	4	5	6	7
3820	(	)	*	+	<	=	>	?
	8	9	:	;	,	-	.	/
3840	Enter	Clear	Break	UA	DA	LA	RA	Space Bar
3880	Shift							

3C00H to 3FFFH — Video Memory — If Bit 7 = 1 (is Set) then Graphics

Note :- UA, DA, LA and RA are the arrow keys! Viz. up, down left and right



IN MAV'S WORKSHOP

by Ian Mavric

Part 4 of the HARD DRIVE Trilogy

HARD DRIVES AND THE TRS-80 MODEL I

Last time[1] I concluded my trilogy of hard drive articles for the Model III and 4. I focussed on the Model III and 4 since 99% of hard drives sold to the 8-bit Z80 systems were used on those computers. In fact not many people know that a hard drive could be even connected to a Model I. Obviously I own one, I use it a few times a week, and I was part of the team who helped make the FreHD[2] hard drive emulator work with the Model I, so I think I know a bit about them.

HISTORY

In 1977 when Tandy designed and built the first few machines with Level I BASIC and 250 baud cassette storage they could hardly have imagined that only 5 or 6 years later Winchester hard disk storage, similar to that available on mini-computers, was going to be affordable and store megabytes of data. When you are designing a 4K microcomputer 5 megabytes must have sounded like a miracle. When Tandy went back to the drawing board to design the Model III, it was clear that the large 50-way I/O bus was designed around connecting a hard drive. The Model III hit the showrooms in 1980 and the hard drive in late 1982 as our old favourite 5 Meg Disk System 26-1130 in RSC-8 Sept 1982.

Something as expensive as a hard disk drive normally needs a killer-app in order to hook in potential buyers and in the case of the Model III it was Profile III Plus HD, and early database management system that was fast and with a hard drive, capable of building enormous databases. There was no such app for the Model I and since the computer had been forced into retirement in 1980 by the FCC, Tandy could probably have overlooked making the hard drive work with the Model I and not many people would have cared. This is where you see the clever minds at work at Radio Shack pulling yet another trick out of their hat for Model I owners. A small adapter kit called, blandly enough, the "Model I Hard Disk Adapter Kit" (26-1132) was sold for only $70 (see below). Once you own the hard disk, you are kind of locked-into the TRS-80 product for the next few years. It is no co-incidence that 5Mb hard drive works perfectly on the soon-to-be-released Model 4. So what if the owner had a lot of money invested in their Model I, and didn't feel like

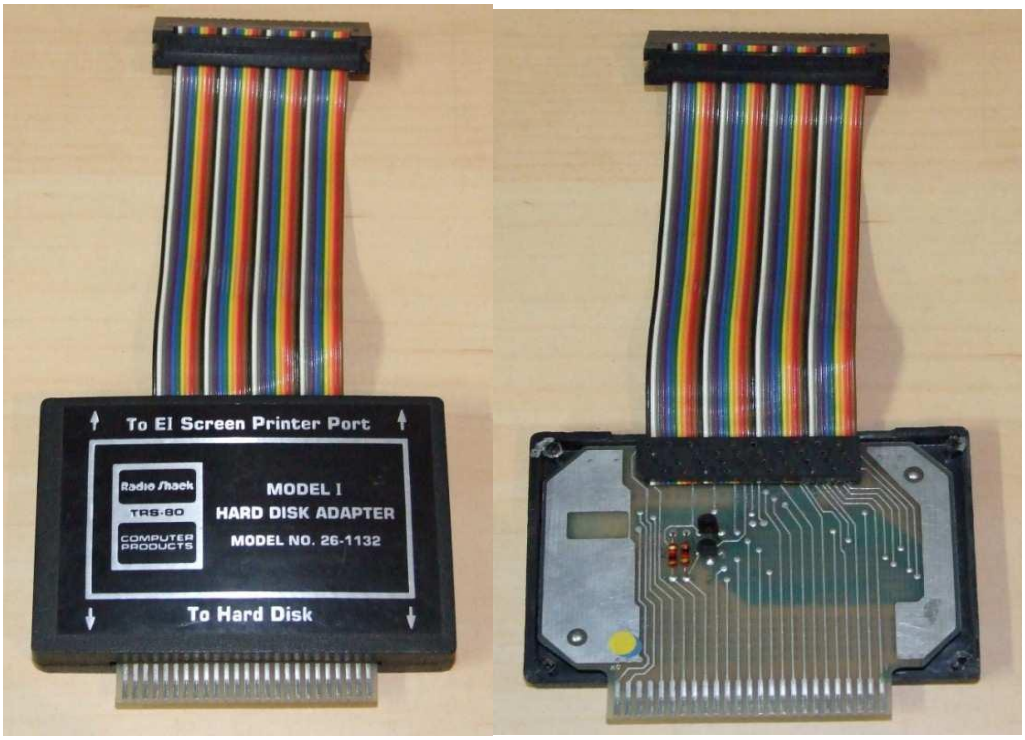
buying a Model III - no problem - our next machine will blow your socks off and in addition, your expensive hard disk drive will plug straight in. Tandy also gets kudos for producing products well into the 1980s for their first computer, much the same as Apple was still producing peripherals which worked on the original (1977-78) Apple][.

Model I Adapter Kit. Includes Adapter, cable and diskettes. Note: May require installation (not included). 26-1132. . . . . 69.95

THE INTERFACE

The 26-1132 kit comprises of a small adapter board (see photos, below) with 2 resistors and 2 transistors on it, a 3 disk-set of what Radio Shack calls "Hard Disk OS" but is really LDOS 5.1.3 with hard disk drivers (TRSHD1/DCT and TRSFRM1/CMD) and a JCL to install he hard drive on the Model I, and a manual. If you follow the instructions you can't go wrong and in about 15 minutes you are up and running with 5Megs of storage in four 1.25Mb partitions, and a boot floppy to get the system started.

While the kit was only listed in the RSCC for only 1982-3, it was available for a couple of years after that while the 5 Meg Disk Systems were still being sold. It's hard to say when it was officially no longer available but I suspect sometime around 1986.



INTERESTING QUIRKS

Yes, there are a few. If you have used a Model III or 4 with a hard drive, you know that if you want to use your floppy DOS and not the hard drive, you just leave the hard drive switched off and boot your floppy disk and play a round of Frogger or Zaxxon. NOT SO with the Model I - the adapter will not allow your Model I to boot unless the hard drive is switched on and up to speed. So if you want to play a floppy only game you have two choices: power up the hard drive and let it sit there humming away next to you, or disconnect it from the Expansion Interface. The first is a waste of electricity and the 2nd is just inconvenient.

The other one stems from a problem which I'm still trying to understand, but seems to manifest itself as an inability to properly write the hard drive configuration file on Model Is which use the later design Expansion Interface. Let me try to explain it.

When I bought my Model I it came with the early design EI, the one with the buffer box. When I bought my hard drive and read the instructions it clearly stated that if you had a very early EI it needed to be modified to use the buffer box, which in turn made it compatible with the hard drive. Since my EI had already been modified I just plugged in the hard drive and off it ran. Some years later I tried to use my hard drive on a Model I with the later design non-buffer-box EI and found that if you booted with a boot disk created with the early EI, it worked, but if you attempted to create a boot disk with the later EI, it wouldn't work properly. No biggie, I just put the later EI away and used the early buffer-box version all these years. Note to self, investigate in the future.

That future arrived when we were trying to make the FreHD hard drive emulator work with the Model I. I noticed the same problem - if I set the FreHD up on a Model I based around a buffer-box EI then I could configure boot disks easily. When I went to do the same thing on a similar system based around a late EI the same problems showed up as using a real hard drive. At least we know the FreHD emulates a TRS-80 hard drive so closely even its foibles are shared.

THE (TEMPORARY) SOLUTION turns out to be a bit awkward but does work. Plug in an early buffer box EI and create your boot disk, then replace it with the later model EI. That way you get the reliability of the later EI and hard disk storage. To backup the boot disk you DON'T need to plug in the old buffer box EI, just QFB :6 :7 and it will backup the disk without hesitation.

MODERN REPLACEMENT FOR THE M1 HD ADAPTER

Over the years people asked me about adding hard drives to their Model Is and my usual response was "Sure! Go find a M1 HD adapter kit 26-1132 and we can get started." I think two or three came through my workshop between 1987 and 2013. This year with the popularity of the FreHD hard drive emulator, people began asking us about making a FreHD version for the Model I, but to my mind this would require a re-design of the board and result in two products on the market. Most people who own a Model I also have a Model 4, and wanted to use the FreHD on either machine as they saw fit. It made more sense to use the 26-1132 adapter so work was taken to make FreHD work on the Model I that way. Without boring you with details, by changing one chip on the emulator (U3) to a 74LS245 the FreHD works on a Model I with the adapter.

So we had a workable solution but the same issue whereby we now needed those hard drive adapters again. Fortunately Bill Buzbee & Larry Fosdick[3] had done all the work in 1992 drawing up the schematic for the 26-1132 adapter, and Fred Vecoven designed a small PCB which does the job very nicely. I named it the Benson box, after the worlds most prolific collector/ hoarder of TRS-80 memorabilia.



VINTAGE DRIVER SOFTWARE

As with the Model III and 4, there are a number of different hard disk drivers out there and they all do pretty much the same thing the same way as each other, except that later drivers offer much more flexibility. Early drivers were called TRSHD1/DCT and only supported the old 5 Meg hard drive. They work fine if you only want 5 megabytes on your hard drive or FreHD. Since most people want to utilise more than 5 megabytes an article in CN80[4] shows how to patch various drivers to use on the Model I.

TRSHD1/DCT can be patched to configure a Tandy 15Meg (26-4155) hard drive called TRSHD15/DCT. Around the mid-1980s Radio Shack released a Model III hard disk driver called TRSHD5/DCT, which can also be patched to become TRSHD51/DCT and for the first time allows you to dynamically configure a hard drive with both heads and cylinders. But really, messing around with old drivers is not a consistent approach, and the best solution would be to use RSHARDx, which we already use on the Model III and 4.

MODERN DRIVERS

Fortunately the work has already been done for us and in the same CN80 article[4] the authors explain how to patch RSHARD5/DCT to become RSHARD1/DCT and it works well. As well as being able to specify heads and cylinders, you can also partition by head or cylinder, maximising the use of your hard drive or FreHD. 40Mb hard drive partitioning is easy and with some thought a hard drive can be partitioned up to 65Mb capacity over 7 partitions. Not bad for an ancient computer originally released with 4K of memory and 250 baud cassette tape storage.

CONCLUSION

Hard drives had been available for the TRS-80 Model I during a very brief period from 1982-84, and have since been rare and almost forgotten. With the advent of FreHD technology the desire to add a hard drive to the Model I generated new interest among enthusiasts, and now with my M1 Hard Drive adapter and updated drivers, adding a hard drive to a Model I is no more difficult than adding one to a Model III or 4.

NEXT TIME: FreHD and how to build it

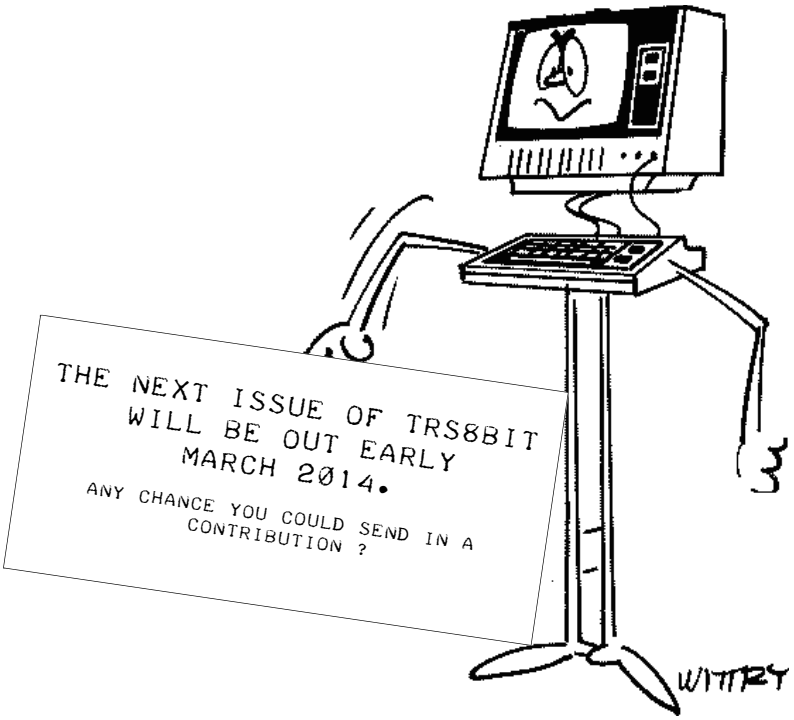
ianm@trs-80.com

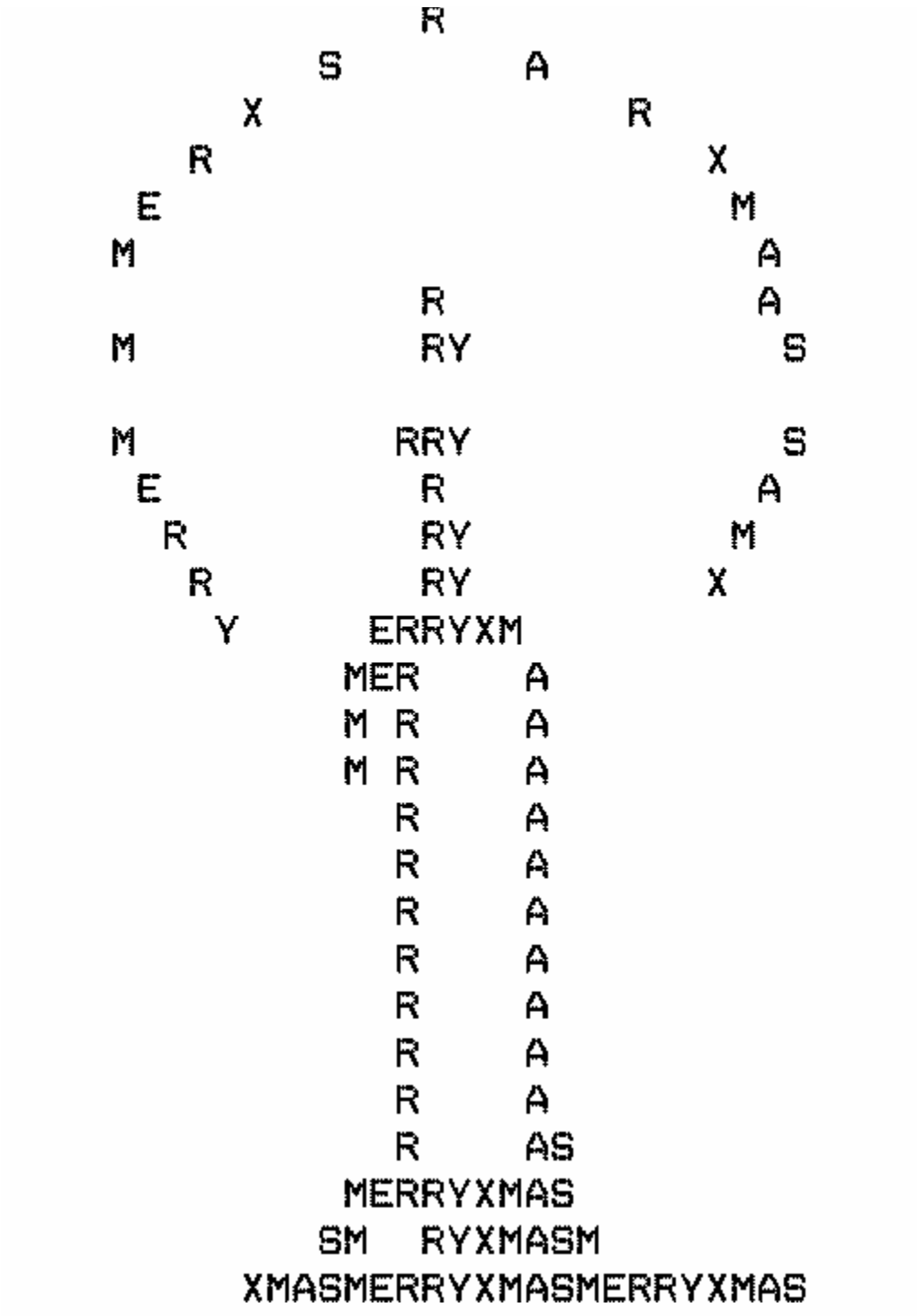
REFERENCES:

1. "In Mav's Workshop", TRS8Bit, Vol 7. No. 2, June 2013 pp.21-26.
2. http://ianmav.customer.netspace.net.au/trs80/emulator
3. Model I HD Adapter.pdf
4. "A Really Hard Model I ", Computer News 80, Vol 4, No. 5, pp. 7-10.



Ian Mavric is an IT Specialist who also restores and collects TRS-80's and classic cars. He live with his wife and kids in Melbourne, Australia.

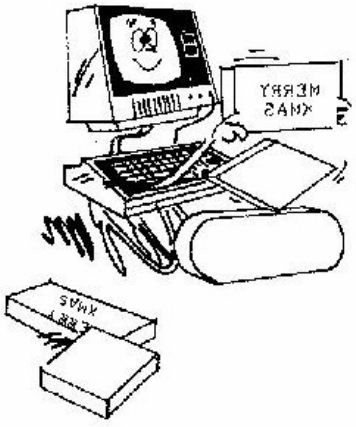



R
S A
X R
R X
E M
M A
R A
M RY S
M RRY S
E R A
R RY M
R RY X
Y ERRYXM
MER A
M R A
M R A
R A
R A
R A
R A
R A
R A
R A
R AS
MERRYXMAS
SM RYXMASM
XMASMERRYXMASMERRYXMAS

```
20 CLEAR500
30 M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMAS"
40 GOSUB 220
50 READ S
60 IF S=0 GOSUB 160
70 READ L
80 T$=MID$(M$,S,L)
90 X=PEEK(VARPTR(P$)+1)+PEEK(VARPTR(P$)+2)*256

100 REM THIS AND THE NEXT LINE ARE NOT NEEDED ON 16 OR 4K SYSTEMS
110 IF X>32767 THEN X=X-65536

120          FOR Y=1TOL
130          POKE X+S+Y,ASC(MID$(T$,Y,1))
140          NEXT Y
150          GOTO 50
160 LPRINT P$
170 GOSUB220
180 READ S
190 IFS=0THENLPRINTP$:READS
200 IF S=99 THEN END
210 RETURN
220 P$=STRING$(75,32)
230 RETURN
240 DATA 13,1,0,9,1,17,1,0,6,1,21,1,0
250 DATA 4,1,24,1,0,2,1,25,1,0,1,1,26,1,0,13,1,26,1,0
260 DATA 1,1,13,2,27,1,0,0,1,1,12,3,27,1,0
270 DATA 2,1,13,1,26,1,0,3,1,13,2,25,1,0,4,1,13,2,24,1,0
280 DATA 5,1,11,6,0,10,3,17,1,0
290 DATA 10,1,12,1,17,1,0,10,1,12,1,17,1,0
300 DATA 12,1,17,1,0,12,1,17,1,0,12,1,17,1,0,12,1,17,1,0
310 DATA 12,1,17,1,0,12,1,17,1,0,12,1,17,1,0,12,1,17,2,0
320 DATA 10,9,0,9,2,13,7,0,6,22,0
330 DATA 99
```



WE HOPE YOU LIKE THE XMAS CANDLE

IT'S JUST TO WISH YOU ALL A VERY
MERRY CHRISTMAS
AND A HAPPY NEW YEAR

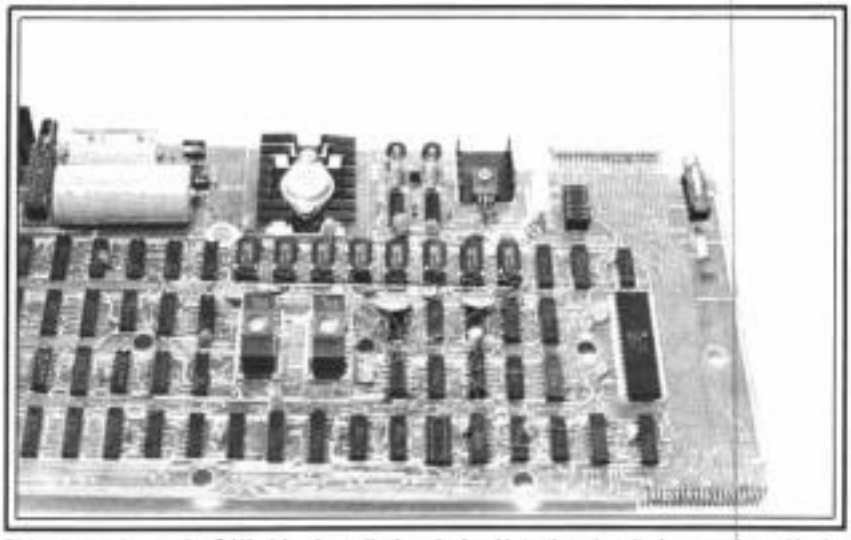
THE MODEL 1 WILL BE 36 YEARS
YOUNG IN 2014!

DUSTY & DEE

With a little effort and not much money you can increase the memory of your TRS-80® Model I to include RTTY and other sophisticated modes of communication in its repertoire.

Updating The TRS-80® Model I To 48K Memory With 64K Chips

BY LEW MCCOY*, W1ICP, AND FRED TREASURE**, KA5NJY



This photo shows the 64K chips installed and wired into the circuit. As mentioned in the text, the ROMs shown bear no resemblance to the original TRS-80® . (TRS-80 is a registered trademark of Radio Shack.)

There are many reasons why the owner of a TRS-80® Model I might want to install full memory in the computer proper (not in the interface). With the recent drop in prices on 4164 dynamic RAMs (about $50 for a complete set of 8) it becomes very attractive to add memory to your Model I. The 64K chips will bring your system up to the full 48K of RAM it is capable of running. Keep in mind that the bottom 16K is reserved for ROM, which holds BASIC, operating systems, etc.

One of the advantages of using the 4164 is that it requires much less power (cooler operation). In fact, the heat reduction is dramatic. Before making the modification, the power supply pass transistor got very hot. Now it runs so cool, that it is hard to detect that it is actually operating. Another advantage in the modification is that it cuts down on reboots from memory errors. There is a lot less chance of glitches with only 8 memory chips versus 24. And most important, the modification eliminates the memory problems so common with the interface units. This is particularly true of the earlier interface models, although the memory problem can exist with the newer interfaces.

We were lucky enough to obtain some factory rejects of interface boards, and almost invariably the problems with the boards were memory problems. By installing the 64K, 4164 dynamic RAM chips on the computer board itself, memory errors common to using the interface vanished.

4164 Variations—Watch Out!

The pin numbering on the 4164s is different from the existing RAMS used in the Model I. However, no socket changes are necessary to make the modifications. All that is required is to lift certain pins on the new RAMS (pins 8 and 9) so that the pins do not make contact with the sockets. These pins which were lifted or bent up will be wire-wrapped for their new connections. With some types of 64K RAMs, a few manufacturers require that pin 1 be connected to +5 volts. Be careful here because on the existing sockets on the TRS-80® Model I board, pin 1 is −5 volts. In our modification, such types of the 64K RAMs must have the pin lifted so that it is not engaged in the socket. They must be lifted and then strung together and connected to the +5 volt line. (The OKI chips we used do not have pin 1 connected internally and pin 1 is not connected to +5 volts, so this was not a problem in the modification shown in the photo.) Of course it would be remiss on our part if we tell you to be careful of the manufacturer and then not tell you where we got our chips. Here is the supplier: Microprocessors Unlimited, P.O. Box 8080, Beggs, Oklahoma 74421, phone 918-267-4796. The chips are designated 220 ns, OKI, plastic 64K dynamic RAMs. When we got ours, they were about $5.00 each.

Required Modifications

We could do the modifications in a step-by-step manner, but the changes are really simple. The project just requires some care. It is assumed that anyone desiring to make this change has long since voided the warranty by opening his or her computer to make changes. However, the changes should be clear so that any amateur with even basic soldering knowledge can make the mods required. First, please observe all the usual warnings and precautions about handling the 4164s. They are sensitive to static electricity and can be ruined through careless handling.

After removing the computer case, carefully remove the existing RAMs, Z13 through Z20. They are all in a line togeth-

*200 Idaho St., Silver City, NM 88061
**Route 8, Box 28E2, Silver City, NM 88061

er on the top row just below the large power-supply pass transistor. Next, carefully bend pins 8 and 9 on each of the new chips to slightly more than 90 degrees, enough so that you will be able to wire-wrap the pins. (If your 4164s have the pin 1 connected as mentioned above, you will need to lift these pins also.) Now carefully insert all the RAMs into the sockets, making sure that all pins make good connections into the sockets. Pin 8 on the new RAMs requires 5 volts, so these pins should be wire-wrapped, daisy-chained, and then connected to the 5 volt bus. In order to reduce noise pick-up, both ends of the line were connected to 5 volt points. Z20, pin 8 is connected to Z39, pin 16 (5 volt point), and Z13, pin 8 is connected to Z31, pin 16 (5 volt point). (Be careful to check your wire-wrapping to make sure that none of pins 8 are shorted to pins 9.)

In order to address this new memory we need to create a new line, **MA7** (**M**emory **A**ddress **7**). Fortunately, this is extremely simple. Z51, a 74LS157, is an address multiplexer, and if you examine it, you will find that pins 12, 13, and 14 are not connected to anything. These pins will be used to create our new line **MA7**. First, daisy-chain all the pins 9 together on the memory chips using your wire-wrap wire; then connect a lead from pin 9 of Z20 to pin 12 of Z51. We routed the wire through the large hole between Z51 and the ROM socket Z34 to make the connection to pin 12 of Z51 on the bottom side of the board. Be sure when making any of these connections to use a minimum of solder and heat. Naturally, one should check any new solder joint to make sure that it isn't shorting to any adjacent line or connection. Make a connection between Z51, pin 13 to Z38, pin 11. This is address line 14 and is needed for addressing the new memory. Likewise, connect Z51, pin 14 to Z38, pin 9, which is address line 15.

If your chips require pin 1 to have +5 volts, they must be wired together and connected to the +5 volt line. It is probably a good idea to connect both ends of the string to the 5 volt line as you did with pin 8.

One last modification is required. It is necessary to enable the memory locations above 32K by activating the signal **MEM***. We can do this by inverting A15 using a spare gate. This modification is a little trickier if we want to use the spare inverting gate that is available to us. This gate is located on Z37, pins 8, 9, and 10. The problem here is that pins 8 and 9 are grounded, and the ground connection is under the chip. We used a desoldering tool to clean out the pin holes. A small prod was used to lift the two pins from the board so we could get at them. You could get another 74LS02 and mount it piggyback and then wire it for +5 volts and ground rather than unsolder the existing pins, but that is up to you. Next connect pins 8 and 9 together and connect them to Z38, pin 9 (address line 15). Last, connect a signal diode such as a 1N4148 with the cathode (bar) to pin 10 of Z37. The anode end (arrow) is connected to pin 10 of Z74. That completes the modifications. There are no other modifications required to complete the project.

Some Conclusions

It is important to keep in mind that the first 16K of memory is reserved by the ROMs in the TRS-80® Model I. We understand that there are other operating systems available (in ROM), but presently we have no further information. In the photograph of the unit shown here, the ROMs used are a special program dedicated to various amateur radio functions such as RTTY, code copying, slow-scan, etc. But that is another story that may never be written—the system is always in a state of flux!

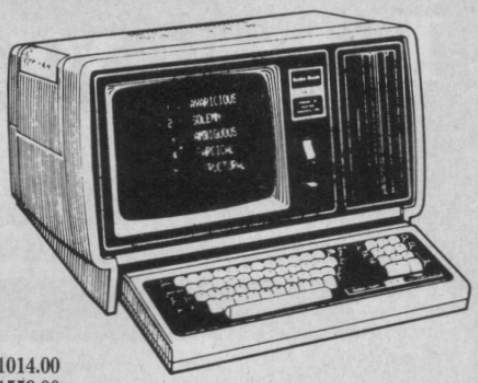
One last item: Those people who have interfaces with memory in them must remove the memory chips in the interface. This prevents memory contention between the two sets of memory.

We have modified three computers so far. The first took several hours of study and work. The last one took about two hours to modify using this article as a guide.

Is the work and expense worth the effort? We think it is. Getting rid of the heat and memory glitches was well worth it.

All Apple products in stock—Call for prices

SAVE BIG BUCKS



Model II 64K $3395

26-4160 1 Drive Exp	$1014.00
26-4161 2 Drive Exp	1559.00
26-4162 3 Drive Exp	2099.00
26-4530 Scripsit II	259.00
26-4512 Profile II	159.00
26-4511 Visicalc II	259.00
26-4501 Gen Ledger	179.00
26-4506 Mail List	72.00

MODEL III

26-1061 4K I	$ 599.00
26-1062 16K III	859.00
26-1066 48K III W/2 Drives, RS 232	2225.00
26-1162 1st Drive	765.00

IDS PAPER TIGERS

Dot Resolution Graphics quality print

IDS 445G 7 wire printhead, graphics	(List $ 985)	$ 795.00
IDS 460G 9 wire printhead, graphics	(List 1094)	886.00
IDS 560G 9 wire, wide carriage, graphics	(List 1394)	1129.00

Model I

26-1140 Expansion Interface	$249.00
26-1141 16K Exp. Interface	359.00
26-1142 32K Exp. Interface	469.00
26-1145 RS 232C Board	84.00
26-1160/1 Mini Disk Drive	419.00
26-1563 Scripsit-Disk	79.00
26-1566 Visicalc	83.00

PRINTERS

26-1167 9½ Line Printer VII	$ 360.00
26-1166 Line Printer VI	1080.00
26-1158 Daisy Wheel II	1798.00
26-1165 Line Printer V	1710.00
26-1401 Cable	36.00

Color Computer 4K $353

26-3001 4K	$353.00
26-3002 16K Ext. Basic	533.00
26-3009 Joysticks	22.50
26-3010 Color Video	353.00
26-1206 Recorder	54.00

All prices are subject to change without notice. Freight, handling, and insurance charges are extra. Most items are shipped United Parcel Service. Model II and other hardware weighting over 50 lbs. is shipped freight collect by truck. Certified Check for immediate shipment from stock. Master Charge, Visa or Bank Card add 3% surcharge.

Vern Street Products 114 West Taft Sapulpa, Ok 74066

The Computer Store, Inc. 4949 South Peoria Tulsa, Ok 74105

918-747-9333

Virtual Floppy disk using a Raspberry Pi

Alan Page

Introduction to the project

Last winter I started a project to create a Virtual Floppy disk emulator for the TRS-80 Model I using a Raspberry Pi.



The project was inspired by several similar projects based on Arduino and other micro-controllers and because my ancient TRS-80 Model I was having problems reading and writing floppy disks as the electronics and hardware aged. It's also rather difficult to transfer software to and from the TRS-80 without such a system.

The advantage of the Raspberry Pi is that it is a complete system with video, USB, keyboard, mouse and storage support. The disadvantage is that the current available operating systems are not ideal for real-time. But the hardware features of the processor help overcome that limitation as will be seen later.

TRS-80 Floppy Disk Interface (Model I)

The TRS-80 Model I expansion interface (which supported the first floppy drives for the TRS-80) was created around 1978. It was designed to use existing floppy disk drives (with perhaps some minor customizations) connected by ribbon cable to the expansion interface. It used a Western Digital 1771 floppy disk controller IC which supported single density only, running at a clock speed of 1 MHz. (The timing diagrams on the 1771 data sheet are for 2 MHz so you have to adjust them for the slower speed of the Model I.)

Interface Signals

For this discussion I will refer to output signals as signals output from the Raspberry Pi emulator that input to the TRS-80 expansion interface and input signals as signals coming into the Raspberry Pi emulator from the expansion interface. Because several floppy drives could be connected to the same line the TRS-80 used open collector TTL interfacing with 150 ohm pull-up resistors to 5 volts. This meant that 'active' signals pulled the voltage down to zero.

Note: The connectors on the TRS-80 Expansion Interface are flat two-sided card-edge connectors but the schematic in the technical reference manual refers to "Pin numbers" so "Cable pin" refers to the pin numbers from the schematic.

The output signals are:
Index Pulse. Cable Pin 8. This short (4 millisecond) pulse is sent at the beginning of every track. On a real floppy drive this is generated by a photocell pointed at a LED shining through the index hole on the floppy disk.
Track Zero. Cable pin 26. Active when the R/W/ head is at track zero (the home position). This is used by the floppy disk controller when seeking.
Write Protect. Cable pin 28. Also generated by a photocell/LED combination when the write protect notch on the disk is covered.
Read Data. Cable pin 30. This is the data from the disk R/W head that has been transformed from the raw analog signal to digital pulses by circuitry in the floppy disk drive. Because drives may vary slightly in speed from the ideal 300 rpm the floppy disk controller is somewhat tolerant of variations from the ideal signal. My system has an add-on data separator board from Percom that improves the performance in this area.
Because more than one floppy drive can be connected to the cable, a drive should only pull an output signal low when it has been made active (selected) by the drive select signal. The TRS-80 supported up to four drives, numbers from 0 to 3.

The input signals are:
Drive Select 0. Cable pin 10. Selects drive 0 when active. In an actual drive this would cause the R/W head to press against the disk surface. DS0 on the schematic.
Drive Select 1. Cable pin 12. Selects drive 1 when active. DS1 on the schematic.
Drive Select 2. Cable pin 14. Selects drive 2 when active. DS2 on the schematic.
Drive Select 3. Cable pin 32. Selects drive 3 when active. DS3 on the schematic.
Motor On. Cable pin 16. Turns all drive motors on. The emulator should only be transmitting track data on the Read Data pin and accepting data on the Write Data pin when both Motor On and the appropriate Drive Select signals are active. The expansion interface contained a timer circuit that would automatically shut off this signal after a few seconds if the operating system was not using the drive interface.
Direction Select. Cable pin 18. Selects the direction of movement of the R/W head when the step signal is pulsed.
Step. Cable pin 20. This short (about 8 microseconds) pulse indicated that the R/W head should be moved one step in the direction indicated by the Direction Select pin. Fortunately

the Raspberry Pi has programmable edge detect triggers that will watch for this and set a bit when the line is pulsed. Since an actual floppy disk takes a couple of milliseconds (or more) to move the physical head from one track to the next but the emulator can move the virtual R/W head almost instantly, the timing is not critical. I have been able to poll the edge detect signal rather than trying to use interrupts (which are not well documented).
Write Data. Cable pin 22. Not yet implemented. This would contain the data to be written to the disk surface.
Write Gate. Cable pin 24. Not yet implemented. Signals that the write data is valid and should be written to the disk. Stays low for the duration of the write operation.

Interface Circuitry

There were two main considerations when choosing what ICs to interface between the Rasperry Pi and the TRS-90 expansion interface. The first was the TTL logic (5v) used by the TRS-80 which is not compatible with the lower voltage (3.3v) logic on the Raspberry Pi. The second was the very low resistance (150 ohm) of the pull-up resistors. This results in a fairly high current and I needed to use 74LS06 to drive the output signals. Also, many level translation chips, especially bi-directional ones, would not work with such low resistance pull-up values. Fortunately I found I was able to avoid level translation chips in my design. While I could have used a 74LS05 for the input signals I found it was simpler to use a 74LS06 for both input and output so I could chose the IC that was most convenient.

Originally I put a level translation between the Pi and the 74LS06 for the output signals but it turned out not to be necessary and the Pi seems to be able to drive the 74LS06 directly. The input signals use the open collector of the 74LS06 (could be a 74LS05, but easier to use the same IC for both) and rely on the pull-up resistors on the GPIO pins to perform the level translation to the correct voltage. The Pi has programmable pull-up resistors on the GPIO pins so I just program them in the initialization code. Two GPIO pins also have pull-up resistors on the board itself, so it's convenient to use these two as input pins for the floppy drive interface.

Note: all Raspberry Pi pins refer to the Rev 2 pins.

Current configuration is shown below.

Read Data must be connected to GPIO pin 10.
Write Data must be connected to GPIO pin 9.
GPIO pin 11 (which is the SPI clock) should be left alone. I connect it to a pin for use with a logic analyzer to help me see what's happening.

I also leave GPIO pins 7 and 8 alone as they are part of the SPI interface and unavailable as I am using that interface for read and write data. GPIO 2 and 3 have pull-up resistors on the board itself so I use them as inputs.

Note: All outputs are pulled up to 5v by 150 ohm resistors on the Expansion Interface.

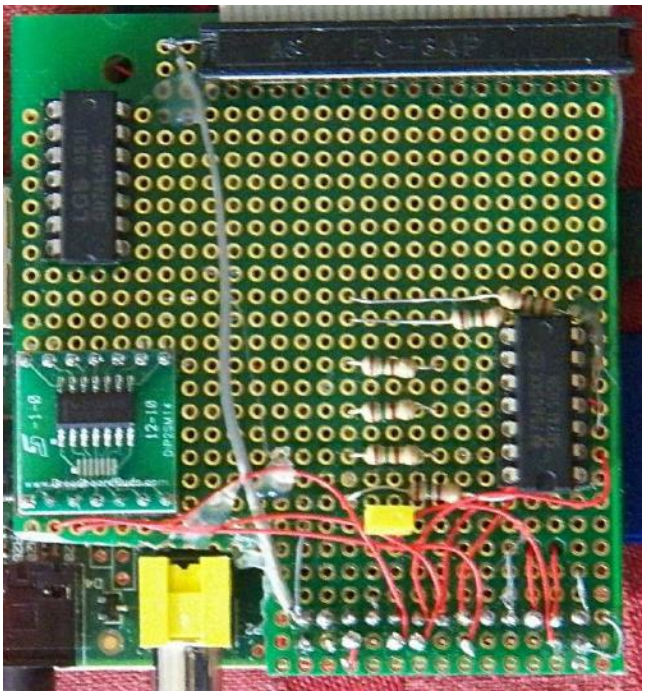
Write Protect (output) connected to GPIO 18 - Header pin P1-12.
Track 0 (output) connected to GPIO 23 - Header pin P1-16.
Read data (output) connected to GPIO 10 - Header pin P1-19.
Index Pulse (output) connected to GPIO 24 - Header pin P1-18.

Drive Select 0 (input) connected to GPIO 2 - Header pin P1-03. (has 1K8 pull-up on the board)
Drive Select 1 (input) connected to GPIO 25 - Header pin P1-22.
Motor On (input) connected to GPIO 4 - Header pin P1-07.
Direction Select (input) connected to GPIO 17 - Header pin P1-11.
Direction Step (input) connected to GPIO 27 - Header pin P1-13.
Drive Select 2 (input) connected to GPIO 22 - Header pin P1-15.
Write Gate (input) connected to GPIO 15 - Header pin P1-10.
Write Data (input) connected to GOIP 9 - Header pin P1-21.

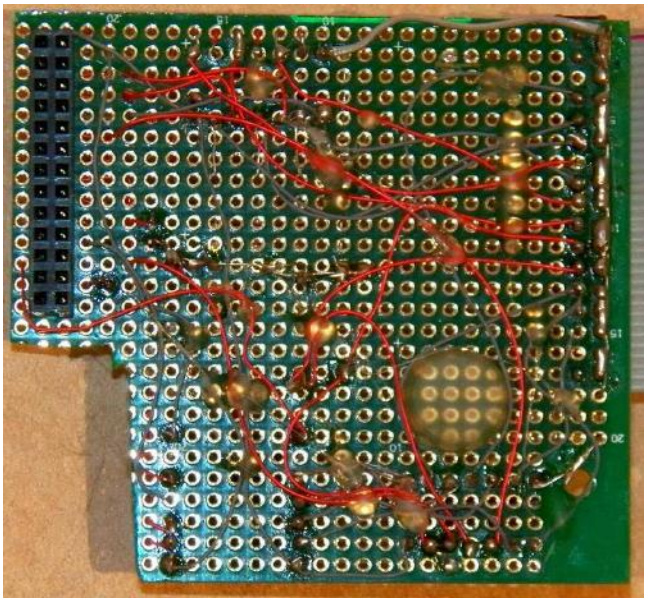
Update Sept 30. I found that using GPIO 3 as an input caused Rasbpian *to hang while booting so I have changed Drive Select 1 to use GPIO 25, header pin P1-22 instead. This problem doesn't exist under RISC OS so perhaps a driver needed to be disabled.*

One of the drives connected to the cable should have 150 ohm pull-up resistors to 5v. Only one drive should have the pull-up resistors and generally it is the last drive on the cable. I put a set of pull-up resistors on my interface board and a jumper between the resistors and 5v from the Pi power supply. The TRS-80 drive cable does not provide +5 volts so it's necessary to use +5 volts from the Pi board.

Not listed here are the ground signals from the cable which should be connected to ground on the Pi.



This is a picture of the top of the original prototype (plugged into the Raspberry Pi beneath it). The IC on the left is a level translation board which I found was not necessary. The connector to the Pi is at the bottom left. The cable that connects to the Expansion Interface is at the top. The resistors are pull-up resistors and the yellow jumper connects them to +5v. The board itself is just a plain prototype board with a cutout for one of the Pi connectors that is too high for the pin header.



This is a picture of the bottom of the original prototype. The connector that is plugged into the Pi is at the left. The floppy disk connector is at the right.

MISSILE DODGE

I'm sorry but I can't trace the author(s) of this super little program. The website from where it was found has now vanished! I think you'll agree, it's a really clever bit of coding and as you all know, I just love my one-liners!

I set myself a challenge of writing a program using a single line of BASIC code. After writing a few simple programs I wondered if it was possible to write a game in one line of code?

It turns out it was possible. I wrote "Missile Dodge", where a player at the top of the screen has to move left or right to dodge missiles being fired at him from the bottom of the screen.

Although my original code was long since passed into the ether, I was reminded of the challenge recently. I decided to dig out a TRS-80 emulator and try to re-create Missile Dodge.

```
>LIST
0 IFI=0THENCLS:PRINT"MISSLE DODGE: Z=LEFT, X=RIGHT":PRINT"DIFFIC
ULTY";:INPUTD:CLS:I=1:GOTOELSEFORX=1TOD:PRINT@959+RND(64),"!";:N
EXT:PRINT:P=P+(PEEK(14344)=4ANDP>0)-(PEEK(14344)=1ANDP<63):IFPEE
K(15360+P)=33THENPRINT"SCORE";SELSEPRINT@P,"*";:S=S+1:GOTO
READY
>_
```

It turns out all the old programming tricks came flooding back to me, and I'm pretty confident that this code is faithful to my original program:

I wrote the program on this emulator:
http://ht.homeserver.hu/bin/emu/ht1080z_v1.6.2.zip

But I also tested them on TRS32:
http://www.trs-80emulators.com/ so it should work on almost anything. (Although I have been told that some emulators run too fast for the game to be playable – the two emulators I mention seem to simulate the clock speed of the original TRS-80, so the game plays fine.)

A few notes:

(1) the maximum line length for a BASIC program on these is 254 characters, but the standard input buffer for entering commands isn't that big, so you have to input part of the line, then edit the line to complete it! To edit a line do "EDIT 0?. Then "x" will take you to the end to append more.

(2) in this basic the parser doesn't require spaces to correctly identify commands, so there is a saving there. Doesn't make it very readable though!

(3) the screen is 64 characters across by 16 rows down. The PRINT @ command allows you to place output anywhere on the screen, with position 0 at the top left and 1023 at the bottom right

(4) PEEK(X) allows you to read memory address X. Address 14344 is part of the keyboard memory map and allows me to see if a key is pressed or not. I don't use the built in INKEY$ function in BASIC because that doesn't allow you to hold down a key, you'd have to stab the key repeatedly to move.

(5) the 1024 characters of the screen memory are from 15360 to 16363, so I can PEEK at that to see if there is a missile at the point where I'm about to draw the player to detect a missile collision

(6) PRINT moves the cursor to a new line unless you terminate it with a ";". If you PRINT on the last line of the screen and don't use ";" then the screen will scroll. I use this functionality to scroll the screen and therefore animate the game. Because the player is on the top line it disappears so I don't need to blank it out before I draw it at the new position. Both of these give me very cheap animation! If I'd had it scrolling the opposite way (missiles coming down at you) then it would have been much harder – and indeed I wouldn't have been able to do it without using machine code. I know – I tried. Back in the day I ended up writing some machine code libraries to scroll the screen in an arbitrary direction, and it still wasn't quite as fast.

Anyone else want to have a go writing a game in this severely limited medium?

Here's a guide to the BASIC:
http://www.trs-80.com/trs80-info-level2.htm

These give details of that keyboard map:
http://home.online.no/~kr-lund/Genial_Joystick.htm

http://www.classic-computers.org.nz/system-80/manuals_tm3_page21.jpg

http://www.classic-computers.org.nz/system-80/manuals_tm3_page22.jpg

And some other useful technical stuff:

http://www.classic-computers.org.nz/system-80/manuals_technical.htm

Tips for beginners :-
(1) write the program on multiple lines and only condense it when it works!
(2) using the emulators you can copy the code out of the emulator, edit it in your favourite editor, and then copy it back – although I kinda like the clunky editor in the BASIC, it takes me back to my youth!



ASK MAV - MODEL 4 HARDWARE QUESTIONS

by Ian Mavric

1.
Q. I have a 16K cassette-based Model 4, how do I upgrade it with disk drives?

A. To upgrade a Model 4 disk drives, in addition to the disk drives themselves, you need the following: disk drive towers, disk controller (FDC), disk drive cable, motherboard-to-FDC cable, 2nd power supply, power cables for the FDC and each disk drive, grounding cables, and assorted screws and washers to mount the new parts. Since a 16K disk-based computer makes no sense you also need 8x 4164 Ram chips to upgrade the computer to 64K. While the computer is open you may as well upgrade it to 128K, so make that 16x 4164 Ram chips. Your motherboard will need a PAL chip installed at U72, so contact me if you need one, or any instructions on how to upgrade the Model 4 to a 64K disk system.

2.
Q. I imported a Model 4 from the USA and it has a B&W CRT not a green one? It's got screen burn and I want to change it, how do I change the CRT?

A. The first few production runs of Model 4 computers for the US domestic market came with B&W CRTs, so this is normal. The rest of the world got green CRTs from the get go. You can easily replace the CRT if you watch and follow all the steps in my YouTube video: http://youtu.be/9sMXSMSsCZU Finding replacement tubes can be difficult to find. I sell them but if you are looking for them from electronics parts supply firms in your locale, the green tube carries the part number 12VCMP31, and the B&W tube is part number 12VCMP4.

3.
Q. My Model 4 RS232 serial board doesn't work, can I use one from a Model III?

A. Yes, you can use a serial board from a Model III, they are interchangeable. If you have one of those Model IIIs or 4s sold without a serial board, you can install one from any Model III or 4 computer, or any serial board listed in eBay for a Model III on a Model 4 and vice-versa.

4.
Q. Where are the brightness and contrast controls on a Model 4?
A. Under the left-hand side of the computer.

5.

Q. The cassette cable from my Model I won't plug into my Model 4, is this right?

A. Yes that is right, the cassette cable supplied to Model I computers has a black hard plastic boss on it with a thick plastic ring. It fits into the Model I "Tape" plug but it won't fit in anything else. You need to buy the later type cassette cable which has the thin metal ring protruding from the black flexible plastic boss. Tandy sold the Model I cassette cable up until about 1980, so all cassette cables made after that time is the one you want.

ianm@trs-80.com

GENIE OWNERS – LOAD AND EVEN COPY SYSTEM TAPES ONTO YOUR SECOND CASSETTE WITH THE

SYSTEM LOADER

Though it is possible to both load and save Basic programs onto Genie's internal and external cassette systems, the inherent fault is that machine language programs can only be loaded via the internal cassette, which sometimes is not so reliable.

Mike Chalk has altered all that! For with his System Loader not only can you actually load machine language program through the second cassette port, but can even copy them as well!!

Yes, you can actually make copies of system tapes, and have the choice of either the second or internal cassette to do it. It's all made very easy too, just by pressing a single key, with no need for a file name.

Still more goodies in the program, for Mike has devised it so that it prints out the name, the entry, start and end addresses in the bargain. And there's a facility to jump to any address during the procedure.

It's a Basic program which can therefore be loaded from any source and it creates a machine language program to do the work. It's even useful for the TRS-80 too!

It's from Kansas and only from Kansas – **£8.50**

DISASSEMBLER

This program is the ideal introduction to machine language, for it allows you to 'list' system tapes and even the resident Microsoft Basic in ROM, allowing the start and stop addresses to be specified with each line displayed slowly and a facility to halt the display.

There is the option to either disassemble parts or whole machine language programs from memory, or strings of machine code can be types which will be translated into standard assembly language.

It's another from Mike Chalk, so has a bonus – of allowing conversion of hexadecimal to decimal and decimal to hexadecimal.

Suitable for both the Genie and Tandy TRS-80 and only from Kansas – **£8.50**

Kansas

All prices VAT paid and post free. First class return post service. Barclaycard, Visa orders dispatched same day if phoned in by 3pm. Answering machine ordering service after 6pm and Sundays to allow use of cheap rate. Catalogue available on request.

Kansas City Systems, Unit 3, Sutton Springs Wood, Chesterfield, Derbys. Tel 0246 850357

Goodbye hard drive, Hello FreHD

By

Garry Howarth, Sydney, NSW

After 25+ years of trouble free service my faithful old Toshiba hard drive finally went out in a blaze of glory. Time to look for a replacement MFM drive. Calls to contacts in the computer game yielded zero success and I wasn't prepared to chance my luck on a 'it worked when it was removed' hard drive from eBay.

Not long after the old Toshiba went to god the FreHD appeared on the scene.
The FreHD is a basically a solid state drive with a battery backup RTC for the TRS-80 and System-80/PMC/Video Genie computers that uses a common garden variety SD card for storage. Designed by Fred Vercoven, built and sold by Ian Mavric with input from many including Andrew Quinn, David Cooper, Malcolm McLeod, Raymond Whitehurst and a cast of thousands. Apologies to anyone I forgot to mention but if you have contributed to this project then take a bow.

A solid state drive for the old 4P, this could be the way to go.

I swapped a few emails with Ian Mavric, exchanged the necessary down-under dollars and a few days later the postie arrived with a present.

Installation was as easy as. Unplug the smouldering remains of the old hard drive and using the existing 50 pin cable, plug the FreHD into the expansion slot on the rear of the 4P and connect it to a suitable 5 volt supply. ATM I'm using a variable bench supply but any common REGULATED 5 volt wall wart should do the job.

Ian supplied a CD-ROM with various utilities, a 40trk LS-DOS 6.3.1 boot floppy and a 128meg SD card with the appropriate hard drive image.
Other operating systems and boot disks are available including LDOS 5.3.1, Newdos/80 2.5, and CP/M 2.2. I have also tested DOSPLUS IV, DOSPLUS 3.5 for the Model 3 as well as Newdos/86 and all seem to work flawlessly.

The hard drive image on the supplied SD card is configured as 6 partitions named HARDA - HARDF. Each partition having

140 cylinders * 6 heads giving a total of 40 megs of storage.
With a little bit of work it is possible to max it out to 1024 cylinders and 8 heads giving a total of 64 megs.

Insert the SD card and power-up the FreHD, after a couple of seconds the green led will flash and she's good to go.
Place a BACKUP of the supplied boot floppy into drive '0' and press reset. We can now remove the floppy and run the computer entirely from the FreHD.
Backing up data from the FreHD is as simple as inserting the SD card into a card reader connected to a PC. All very quick and easy and no more wasting hours copying files to floppy.

Now while all this is fine and dandy, the whole point of having a hard drive or a FreHD is so we don't have to play around with floppies especially boot floppies.

The boot rom on my 4P has been hacked to allow loading of the ModelA/III rom image from either the hard drive or floppy and Adam Rubin's patches have been applied to LSDOS 6.3.1 to allow booting directly from the hard drive without the use of a floppy.
Naturally I wanted to see if I could get the 4P to both load the Model III Rom image and to boot LS-DOS directly off the FreHD.

Diving head first into the junk box I found the backups of my old hard drive and transferred them over to the FreHD. Took a deep breath and pressed reset.
BINGO
The computer booted directly off the FreHD without a boot floppy. Smiles all round. Boosted by this I next loaded a Model III system disk into drive '0' took another deep breath, pressed reset and the 4P loaded the MODELA/III file from the FreHD in about one second. Not having a working hard drive I could not do any comparison tests but according to my notes it took the computer nearly 3 seconds to load the MODELA/III file from the old hard drive, so this is a big improvement.

Happy that all was working OK, I sent a copy of the 'self booting' FreHD image to Ian for testing. Having used Adam Rubin's mods to LS-DOS for over 20 years without having a single problem I was surprised when Ian could not get them to work on his computer. A problem with the later GA versions of the 4P was causing drive '0' to hang. After a lot of work trying different drivers I managed to track down a later copy of Adam's patches that cured the problem.

Fred Vercoven has now developed patches for the Model III and 4/4P ROM that will allow booting any operating system directly from the FreHD via a menu, which makes all of the above hard work redundant - oh well, as Edward Kelly said "such is life".

Getting back to the aforementioned utilities.

VHDUTL/CMD allows for reading of the root directory of the SD card, setting and reading the built in RTC, displaying the firmware version as well as creating and mounting various hard drive image files.

IMPORT2/CMD and EXPORT2/CMD are used to transfer individual files to and from the TRS-80 and SD card. These utilities





make the job of transferring files between the TRS-80 and a PC extremely easy.

FUPDATE/CMD is used to update the FreHD's firmware.

A quick search of the net will show that nearly all the software for the TRS-80 is now available for downloading from various websites. The problem has been how to get this software transferred to disks that will run on the 80.

Using a combination of Matthew Reed's TRS32 emulator, Trstools, Newdos/80 and Super-Utility I have been able to make images of .DSK files including self-booting disks based on Percom's MicroDOS and transferring these image files to and from the TRS80/System-80 and PC via the RS232. Slow and cumbersome but it worked and worked well. MULTIDOS being the only exception and that is possibly due to conflicts with the XLR8er card in my 4P.

The job of transferring these .DSK images has now been made extremely simple thanks to a programme supplied with the FreHD - DSK/DCT

Copy the required .DSK image files from the PC to the SD card and install DSK/DCT on the TRS-80 using the following LSDOS system command

```
SYSTEM (DRIVE=x,DISABLE,DRIVER="DSK/DCT")
```

Reply to the prompt with the name of the appropriate .DSK image file.

DSK/DCT will allow you to load any .DSK image file saved on the SD card and have it treated as a normal floppy which can then be copied to a physical floppy using DISKCOPY/CMD

After the failure of the old hard drive I was convinced that my 4P would be a floppy-only machine but the FreHD (funny name) has put some life back in the old beast. If anything it is now a faster more versatile computer and jobs that were previously a major pain have become a very simple task.

Finally, many many thanks must go to all the people that freely contribute their time and expertise in keeping these machines alive, long after their use by date.

Gazza

gazzajh@optusnet.com.au

MAV'S BELIEVE IT OR NOT!
by Ian Mavric
SYDTRUG'S MODEL I COMPUTER

Sydtrug, the Sydney TRS-80 and MS-DOS user's group[1] were in the process of moving premises and found they had a lot of old TRS-80 equipment, software, and manuals which I managed to acquire. You find the most unique things this way, and under a dust cover I found what seems to me to be the most modified TRS-80 Model I that I have ever seen.

Looking at the photos I can see the following modifications:

* Orcim Systems hi-resolution graphics character generator upgrade board
* the 16K ram bank has been replaced with 4164 ram chips for 64K on board
* CP/M rom/switchout which allows full use of 64K Ram all ram mode
* Z80B CPU replaced the standard Z80, and the computer has a s speedup mod
* Radio Shack lowercase modification
* Radio Shack Level II ROM upgrade (computer carries cat. no. 26-1001 so was originally Level I)
* Screen de-glitchifier
* relocated reset button
* pin header connection on the expansion bus
* keyboard connector so the keyboard can be disconnected from the mainboard

Does it work? Unfortunately some wires were disconnected and the Level II Rom chips removed so I think its fair to say this machine was not running when put aside. Without the schematics to the modifications it's hard for me to work out what is wrong so I thought I would just let this computer be, and it's now a showpiece in John Benson's TRS-80 collection.

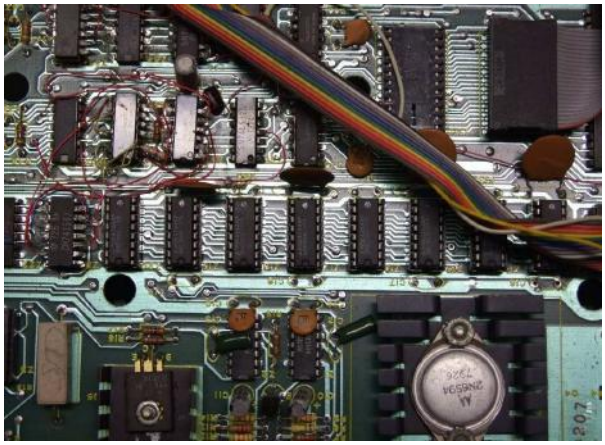
How many super modified Model Is are out there? I can only think of who which are more modified than this one, neither of which I have seen in person. The most well known one would be Dennis Bathory-Kitsz Custom TRS-80 he wrote the book about[2], and the other would be Don McKenzie's original Model I which he talks about on his web site[3].

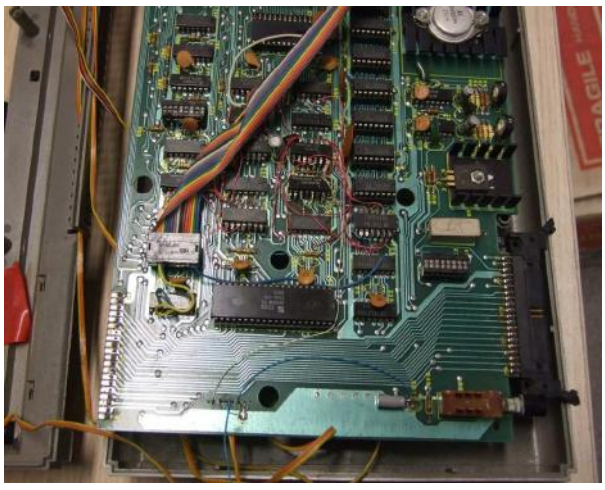
ianm@trs-80.com

REFERENCES:

1. www.sydtrug.org
2. http://www.trs-80.com/cgi-bin/down-ok-books.pl?Custom_TRS-80_and_Other_Mysteries_(1982)(Dennis_Bathory_Kitsz).zip
3. http://www.dontronics-shop.com/is-this-australias-first-pc.html





Radio Shack
TRS-80
MICROCOMPUTER SYSTEM
BREAK
ENTER
CLEAR
SHIFT
SHIFT
ENTER

Not mathematical—this program 'roots out' your defective memory locations.

Babyroot

Dennis Bathory Kitsz
Roxbury, VT 05669

Babyroot is a program that exercises the TRS-80 RAMs. I developed this program because, during a particularly warm and humid day, my programs started to crash for no apparent reason. I scratched out an inelegant but workable first version and located a wayward address, 4A02.

The program exercises all combinations of bits in three neighboring bytes simultaneously. Locations are repeatedly written to and read from to assure that they are functioning.

The Program

Prepare Babyroot as a BASIC program or use T-BUG to enter the machine code and create a SYSTEM tape.

Load Babyroot using the SYSTEM command; enter the name RAMCHK and when the program has finished loading (about 8 seconds), enter /20480. The program, now located at 5000 hex, works over the RAM from address 6000 to 7FFF hex.

At the end of this sequence, it transfers itself to 7000 hex, exercising RAM from 4000 to 6FFF hex. Each complete cycle takes 12 1/2 minutes on an unmodified TRS-80.

At the beginning of the program, the screen clears and a star appears in the bottom right-hand corner. About five minutes later, the program transfers to high memory. The star vanishes and a graphics block appears in the next screen position. Address 3FFF is presented at the top of the screen, indicating that address 3FFF is "bad". The program reads that address as bad because the video memory bank is deliberately designed short one bit.

The video memory ends at address 3FFF and system RAM begins at 4000. This memory location is inadvertently exercised by the program when it begins working over address 4000 and its two neighbors. It serves as a check that Babyroot itself is operating correctly. (If your TRS-80 has a lowercase hardware modification that saves the extra bit, this memory location should read correctly. No bad bit will be presented.)

Errors in address 4000 to 41E5 are usually terminal. 41E6 to 42E8 might or might not cause significant grief. 42E9 and up is where the BASIC programs sit.

If the bad bit coincides with a fixed program bit, everything will look fine. If it is different from a fixed program bit, that program might never LIST or RUN correctly. If the bad bit coincides with a variable location, it will be invisible at first.

But watch those results!

Reading down from address 7FFF, you find control information, strings and the BASIC stack. Errors here can wipe out the program. So, there is a chance that in normal operation a bad bit might go unnoticed for months.

Keep this program running (it will continuously loop) for no less than 8 hours, if you suspect a bad bit. It took 9 1/2 hours before conditions were right and the balky bit turned up in my system.

The computer can be restored for normal use only by turning it off. All system RAM has been upset by Babyrooting around. If the RAM is too badly damaged, Babyroot itself may not load. Write me for a special version of the program for use in this case.

A Bit Error

When you've discovered a bad RAM, what do you do? A bit error is a sophisticated enough problem that you first might want to leave it to your friendly Radio Shack repair center. However, if you find this suggestion unappealing, you'll want to tackle it yourself.

First, you'll probably want to find the bad bit's position in the byte. Now that you know its address, Table 1 lists a short program to identify that bit.

Enter the least significant byte of the bad RAM address in-

```
5000 * 21 00 3C 11 01 3C 01 FF 03 36 20 ED B0 06 08 21
5010 * A0 3C 36 2A 23 10 FB 06 00 0B 21 ## 55 78 77 AE
5020 * 28 F7 C5 06 08 21 A0 3C 07 38 05 23 10 FA 18 05
5030 * 36 58 23 10 F3 21 A0 3C C1 1B DE C9 00 00 00 00
```

Table 1. Bit Search program.

to the position marked ##. Enter the most significant byte into the position marked §§. (You may relocate this program anywhere in memory, if necessary, to avoid balky addresses.) Now set this program in motion. A good byte will be displayed thus:

****

As bad bits are found, the stars will be replaced by letter X, so:

***X*X**

Bit 7 is left-most in this display and bit 0 is to the right. It will continuously loop, exercising this one address, just as the Babyroot program exercised the whole of the memory. After a given period of time, all the bad bits should be indicated by an X.

Next you'll need to buy replacement 4116 RAMs ($8 to $15), one for each bad bit position. Dynamic RAMs are curious devices. If you don't know how to handle these sensitive MOS devices, find out! Look in *Popular Electronics*, on data sheets, in books, ask a friend or whatever, but don't proceed until you know how to handle them! (And put your cats in the bathroom.) For greatest safety, use a static-free workbench.

Get a TRS-80 technical manual or RAM expansion instruction sheet. Open the case, find the RAM(s) corresponding to the bad bit(s). Pull the RAM(s). Replace each with a new 4116 and run Babyroot again. If the error is gone (and it should be), you're set. If not, double check your work, making sure you've replaced the correct RAM. Switch RAMs. See if the bad bit changes position. Go down the line. You may even have a bad new RAM.

If you're desperate, call me; I know the feeling of sitting awake at 3 a.m., wondering whether or not I should turn my TRS-80 into a flower pot. I'll try to help you with your diagnosis. I hope the program runs for days without turning up a bad bit. Good luck.■

```
5000  *  21 00 3C 11 01 3C 01 FF 03 36 20 ED B0 DD 21 80
5010  *  50 3A FE 3F CE 0A 32 FE 3F DD F9 D6 79 33 10 FD
5020  *  21 00 60 AF 77 4F F5 79 77 23 7C FE 80 28 3C E5
5030  *  00 E5 E1 BC 28 29 E1 F1 77 2B 28 77 47 7E B8 28
5040  *  05 16 1B D5 18 18 23 23 7E B8 28 05 16 10 D5 18
5050  *  10 2B 34 DC 7E 47 FE 00 20 CC 23 DE 00 18 C7 18
5060  *  64 7C D6 FD DF DF DF DF 16 22 18 22 7C E6 DF 16
5070  *  18 18 18 7D E6 F0 DF DF DF DF 16 10 18 10 7D E6

5080  *  DF 16 09 18 09 D1 3E 03 92 D0 77 00 18 FF D6 F5
5090  *  11 00 3C 1A FE 20 28 03 13 18 F8 7B FE 04 20 0A
50A0  *  3E 20 18 12 18 12 18 12 18 12 F1 FE 0A 30 04 C6
50B0  *  30 18 02 C6 37 12 C5 06 FF 10 FE C1 01 3E D9 92
50C0  *  00 77 44 18 FF DD E5 DD E5 E1 01 7A EE 20 57 D5
50D0  *  06 80 2B 18 10 FC 01 FF 00 ED B0 DD E1 3E 20 00
50E0  *  AE A2 B0 77 A2 3E FD DD AE AC DD 77 AC DD E5 E1
50F0  *  2E 11 E9 44 45 4E 4E 49 53 20 48 49 54 53 5A 00
```

```
10   FOR X = 20480 TO 20735 : READ A : POKE X,A : NEXT X
20   POKE 16526,0 : POKE 16527,80 : PRINT USR(0)
30   DATA 33,0,60,17,1,60,1,255,3,54,32,237,176,221,33,128
40   DATA 80,58,254,63,238,10,50,254,63,221,249,6,127,61,16,253
50   DATA 33,0,96,175,119,79,245,121,119,35,124,254,128,40,48,229
60   DATA 221,229,225,188,40,41,225,241,119,43,43,119,71,126,184,40
70   DATA 5,22,27,213,24,27,35,35,126,184,40,5,22,16,213,24
80   DATA 16,43,52,12,126,71,254,0,32,204,35,14,0,24,199,24
90   DATA 100,124,230,240,15,15,15,15,27,34,24,34,124,230,15,22
100  DATA 27,24,27,126,230,240,15,15,15,15,22,16,24,16,124,230
110  DATA 15,22,9,24,9,209,62,211,146,221,119,13,24,255,213,245
120  DATA 17,0,60,26,254,32,40,3,19,24,248,123,254,4,32,10
130  DATA 62,32,27,18,27,18,27,18,27,18,241,254,10,48,4,198
140  DATA 48,24,2,198,58,18,197,6,255,16,254,193,209,62,201,146
150  DATA 221,119,68,24,255,221,229,221,229,225,209,122,238,32,87,213
160  DATA 6,128,43,27,16,252,1,255,0,237,176,221,225,62,32,221
170  DATA 174,162,221,119,162,62,240,221,174,172,221,119,172,221,229,225
180  DATA 46,17,233,68,69,78,78,73,83,32,75,73,84,83,90,0

Total program cycle:  12 minutes 24 seconds
Recommended number of cycles:  40 or more
```

Babyroot versions: Machine (top) and BASIC (bottom).

DR. HOWE'S SOFTWARE

For the TRS-80 (*)

MACHINE LANGUAGE SOFTWARE

MONITOR #3 ..... $39.95
Disassembler; memory displays; memory move, search verify, and modify; read and write object tapes; hexadecimal arithmetic; object code relocator; unload programs for disk; symbolic output tapes. 41-page instruction manual.

MONITOR #4 ..... $49.95
Same as Monitor #3 but adds: save and read disk files, direct input and output of disk sectors; send, receive, or talk to another computer via RS-232-C interface; symbolic disassembly on disk.

SMART TERMINAL $49.95
Enables your TRS-80 to be used as a remote terminal to a time-sharing system. Supports lowercase and full range of control keys. Automatic transmission between memory and host computer. Much more.

PACK/UNPACK $24.95
Increase disk file capacity by 33% with NO NEW HARDWARE. Applies only to string data in random files. Ideal for mailing lists, etc.

GAME OF "LIFE" $5.95
John Conway's game of "life" shows patterns evolving and changing swiftly before your eyes. A dazzling demonstration program!

BASIC SOFTWARE

MAILING LIST $69.95
Maintains mailing list files of over 1000 names per diskette. Add, delete, change, find name, machine language sort, print file.

SMALL BUSINESS ACCOUNTING $49.95
Based on Dome Bookkeeping Journal #612, keeps track of income, expenditures, and payroll for a small business of up to 18 employees. Daily, monthly, year-to-date summaries.

HOME BUDGET $49.95
Checkbook maintenance combined with records of income and monthly bills. Monthly and year-to-date summaries showing tax deductions.

DATABASE MANAGEMENT $29.95
Define files of any description and maintain on cassette or disk. Add, change, delete, find, sort, justify, print, line print, total fields, write.

HOWE SOFTWARE ↙103
14 Lexington Road
New City, New York 10956

(*) TRS-80 is a registered trademark of Tandy corp.

16K DYNAMIC RAMS FOR MEMORY EXPANSION

8/$87.20!!

Expand memory in TRS-80* -I and -II, Exidy, Apple, new PET, Heath H89, and other machines with our low power, high speed (250 ns/4 MHz) 16K dynamic RAMs. Add $3 for two dip shunts plus instructions for TRS-80* expansion.
*TRS-80 is a trademark of the Tandy Corporation.

Thinking about easier expansion? Higher speed? More power? Then think S-100. We stock an innovative and extensive line of S-100 bus products, from computer enclosures to color graphics boards. Send today for our free catalogue (foreign requests: add $1 to help cover postage and handling costs).

GODBOUT

GODBOUT ELECTRONICS
Bldg. 725, Oakland Airport, CA 94614

TERMS: Cal res add tax. Allow 5% for shipping, excess refunded. VISA®/Mastercharge® call our 24 hour order desk at (415) 562-0636. COD OK with street address for UPS. Prices subject to change without notice.

↙75

accounts receivable *TWICE as fast!* with

the Spooler for your TRS-80*!

We have benchmarked processing of customer statements and aging reports and found that these reports run nearly twice as fast with **THE SPOOLER** installed.

THE SPOOLER requires no modification to your user application program.

Price: $100. Add $4. for credit card orders. Add $5. for C.O.D. orders
Documentation available separately for $10.
Model II version now available.

Phone (513) 891-4496 or write:

Snapp Inc. ↙232
8170 Corporate Park Drive
Cincinnati, Ohio 45242

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

↙Reader Service – see page 162

```
;
;Modified Babyroot for Model 3, and to continuously move
;from Low to High testing the remaining memory.
;
;You will need to set this Diagnostic for the RM size that
;is in your Model 3/4
;
;Original babyroot was in the May 1980 80Microcomputing
;Pages 76 & 77.
;
;Assembled with Z80asm on Debian 7
;z80asm -l -i babyroot.asm
;
;Larry Kraemer
;
    ORG 5000H
    LD  HL,3C00H
    LD  DE,3C01H
    LD  BC,03FFH
    LD  (HL),20H
    LDIR
    LD  IX,5090H
    LD  A,(3FFEH)
    XOR 0AH
    LD  (3FFEH),A
    LD  SP,IX
    LD  B,7FH
H501D:
    INC SP
    DJNZ    H501D
    LD  HL,6001H
    XOR A
    LD  (HL),A
    LD  C,A
H5026:
    PUSH    AF
    LD  A,C
    LD  (HL),A
    INC HL
    LD  (3C22H),HL
    LD  A,H     ;Change this for RAM Size xxK
    CP  00H     ;80         C0          00
    JR  Z,H5062 ;16K        32K         48K
    PUSH    HL  ;4000+4000  4000+8000   4000+C000
    PUSH    IX
    POP HL
    CP  H
    JR  Z,H5062
    POP HL
    POP AF
```

```
    LD  (HL),A
    DEC HL
    DEC HL
    LD  (HL),A
    LD  B,A
    LD  A,(HL)
    CP  B
    JR  Z,H5049
    LD  D,1BH       ;MAKE 20 TO WATCH IT WORK
    PUSH    DE
    JR  H5064
H5049:
    INC HL
    INC HL
    LD  A,(HL)
    CP  B
    JR  Z,H5054
    LD  D,10H
    PUSH    DE
    JR  H5064
H5054:
    DEC HL
    INC (HL)
    INC C
    LD  A,(HL)
    LD  B,A
    CP  00H
    JR  NZ,H5026
    INC HL
    LD  C,00H
    JR  H5026
H5062:
    JR  H50C8
H5064:
    LD  A,H
    AND 0F0H
    RRCA
    RRCA
    RRCA
    RRCA
    LD  D,22H
    JR  H5091
    LD  A,H
    AND 0FH
    LD  D,1BH
    JR  H5091
    LD  A,L
    AND 0F0H
    RRCA
    RRCA
    RRCA
```

```
    RRCA
    LD  D,10H
    JR  H5091
    LD  A,L
    AND 0FH
    LD  D,09H
    JR  H5091
    POP DE
    LD  A,0D3H
    SUB D
    LD  (IX+00H),A
    JR  H5091       ; VERIFY this instruction with
                    ; babyroot.src
H5091:              ; 18FFH vs 1800H
    PUSH    DE
    PUSH    AF
    LD  DE,3C00H
H5096:
    LD  A,(DE)
    CP  20H
    JR  Z,H509E
    INC DE
    JR  H5096
H509E:
    LD  A,E
    CP  04H
    JR  NZ,H50AD
    LD  A,20H
    DEC DE
    LD  (DE),A
    DEC DE
    LD  (DE),A
    DEC DE
    LD  (DE),A
    DEC DE
    LD  (DE),A
H50AD:
    POP AF
    CP  0AH
    JR  NC,H50B6
    ADD A,30H
    JR  H50B8
H50B6:
    ADD A,37H
H50B8:
    LD  (DE),A
    PUSH    BC
    LD  B,0FFH
H50BC:
    DJNZ    H50BC
    POP BC
```

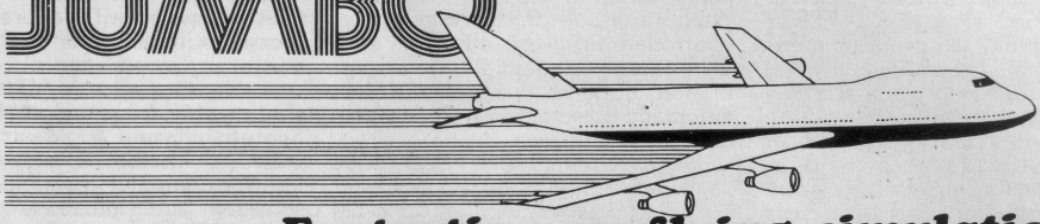
```
    POP DE
    LD  A,0C9H
    SUB D
    LD  (IX+37H),A
    JR  H50C8       ; VERIFY this instruction with
                    ; babyroot.src
H50C8:              ; 18FFH vs 1800H
    PUSH    IX
    PUSH    IX
    POP HL
    POP DE
    LD  A,D
    XOR 20H
    LD  D,A
    PUSH    DE
    PUSH    DE
    LD  B,90H
H50D6:
    DEC HL
    DEC DE
    DJNZ    H50D6
    LD  BC,00FFH
    LDIR
    POP HL      ;
    POP IX      ;CHANGED FOR MODEL 3
    LD  L,22H       ;
    LD  A,20H       ;
    XOR (HL)        ;
    LD  (HL),A      ;
    LD  BC,000DH
    ADD HL,BC
    LD  A,70H       ;F0     B0      70      30
    XOR (HL)        ;16K    32K     48K     64K
    LD  (HL),A
    LD  L,11H
    JP  (HL)
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    END
```

Larry Kraemer is an ex-Military USAF Officer, and a retired Electronics & Instrumentation Technician, that has a hobby of Amateur Radio (Advanced Class), and Computers, with Computer Repair experience.



INNOVATIVE
TRS 80-GENIE SOFTWARE
from the professionals

JUMBO



Fantastic new flying simulation

Occasionally a program comes along of such magnitude that it is hard to describe it, especially within the space allowable in an advertisement. Jumbo is such a program. There have, of course, been flying simulation programs before, but they have all rather fallen into the trap of trying to produce a graphic representation of the ground at some stage or other in the program, thus taking up space, and/or they have concentrated on the single act of flying. In other words, although one gets the feeling of flying a small aircraft, one is not going anywhere.

Jumbo is a fantastically accurate simulation of flying a Boeing 747. These planes are not small, and are not flown by eye. They are flown by eyes and instruments, and the instrument graphics in this simulation are really first class. As you may know, the primary instrument of an aircraft is the artificial horizon. This is simulated very well, and includes instrument landing approach indicators. Graphic maps of very high quality are produced and it is possible to fly in all of the British Isles including Southern Ireland, and to New York on the Eastern U.S. seaboard. The actual airports are at London, Birmingham, Manchester, Prestwick, Edinburgh, Belfast, Shannon and New York.

The program was written by two authors who combined their joint skills to produce a unique piece of software. The programmer got the last byte of performance out of the machine and the pilot the last drop of authenticity out of the simulation. Other simulation programs produce at best a similarity towards flying. With Jumbo you really feel that you are behind the controls.

The authors have used a number of gimmicks and programming techniques. For instance, it is possible to switch on an automatic pilot and to jump forward in time in increments of one minute or one hour — otherwise of course a flight from London to New York or vice versa would take some 8 hours to simulate. Whilst in this time skip the various controls are left as they are but fuel consumption and distance to go are taken forward at their correct rate.

A chart is supplied containing various items of data which you will need, including the take-off data for a 747 with various take-off weights, flap retractions, climb and cruise speeds and descent distances.

Realising that not everybody can fly a 747 and that there is even a large proportion of the population that cannot fly anything, two important items of assistance have been added. First of all is the documentation. This is split into two parts. The second is a "manual" on flying the 747. In other words, the instructions for running the program. The first part, however, is what amounts to an instruction manual for flying. It assumes that the user knows nothing about aircraft and although we do not purport that it gets anybody up to any sort of standard after it has been assimilated, the reader should have sufficient knowledge of flying and the theory of flight to fly in the program itself. The second feature of assistance to the novice pilot is a feature in the program which enables the user to practice landings. When the program is started, if one presses the P key, the aircraft is automatically put 11 miles out from London Airport approaching on an instrument landing.

The controls are pretty well complete, even to dive and wheel brakes. The flying track may be continuously monitored on the map display. Bearing and distance to your intended point of landing are available all the time. The instruments, incidentally, consist of:

Artificial horizon	Attitude	Fuel
Aeleron indicators	Compass	Elapsed time
Indicated airspeed	Turn indicator	Distance to landing
Power setting	Flap indicator	Rate of climb
Elevators	Altimeter	

Six maps may be chosen, as follows:

Scotland, Northern England, Southern England, Ireland, Eastern U.S.A., The whole of the U.K.

Route information and present position are shown on the map displayed. It is very difficult to think of any factor that the authors have overlooked. Even the quite meaningful thinning of the air with increasing height, which greatly affects the airspeed indicator in real life, is taken into consideration. Rather than having a fixed stalling speed, this continuously changes with the flight configuration, the weight, height and power setting, again as it does in real life.

We like to think that we publish good programs. Jumbo is outstanding. It is available on tape or disk for 16K or 32K memory machines. It is compatible with the Model I Tandy, Video Genie, Genie I and Genie II machines.

Tape ..... £15.00 + VAT = **£17.25**

Disk ..... £17.00 + VAT = **£19.55**

Plus 75p P & P please.

MOLIMERX LTD

A J HARDING (MOLIMERX)

1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

TEL: [0424] 220391/223636 **TELEX 86736 SOTEX G**

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 [**refundable**] plus £1 postage.