

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](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 ND80 V2 'BOOT'
COMMAND TO WORK.
THE SECOND IS FROM DAVE
WASHFORD, FONDLY
REMEMBERED BY US ALL AS
'WASHY' ON HOW TO
PERSONALIZE YOUR ND80 DISKS

AND FINALLY ..

PETER STONE HAS COME UP
WITH THE FIRST (OF MANY, I
HOPE) INSTALMENT OF HIS
RE-LEARN Z80 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 Z80 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

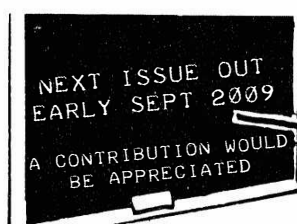


THANKS TO ALL WHO
REMIND ME THAT THE
ASSEMBLER STATEMENT LD A,(IY)
NEEDS THE +0 INCLUDING, VIZ.
LD A,(IY+0). I'D FORGOTTEN
THAT IY AND IX FOR THAT
MATTER, USUALLY DON'T POINT
TO THEIR ACTUAL MEMORY
ADDRESS, BUT ONES THAT ARE
ROUND ABOUT. THEY NEED WAKING
UP SO TO SPEAK!

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 ND80,
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=0/0H.
THIS WAS QUITE OFTEN CHANGED
TO SOMETHING LIKE 62128 WHEN
USING SOUTHERN SOFTWARE'S
EDIT PROGRAM.

MATTHEW REED HAS UPGRADED
TRS32,THE TRS-80 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...

==== 0000 ====

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=1T01E9 AS APPOSED TO FORT=0T01STEP0. 11 BYTES AS AGAINST 14! 3 BYTE SAVED, WELL DONE LADS.

```
1 P=26:C#="X":FORT=1T01E9: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)=790RP>320RP<21PRINT@P,"BOOM":FORW=1T099:NEXT:
RUNELSEK=PEEK(14400):IFK=32P=P-1:NEXTELSEIFK=64P=P+1:NEXTELSENEXT
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 0. 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 0,17 (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 20 (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 0,18 <ENTER> AND YOU WILL SEE A SIMILAR PICTURE. ON THE RIGHT YOU WILL SEE THE MESSAGE 'VERSION 2.0' ,AND LOWER DOWN (SPREAD OVER TWO LINES) 'MODEL.I'. I CHANGED MY MESSAGE TO READ 'MOD 1 VN2,0' AND 'ZAP 079' 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 20(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 0,164 <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 CHANGE
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!

==== 000 ===

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.

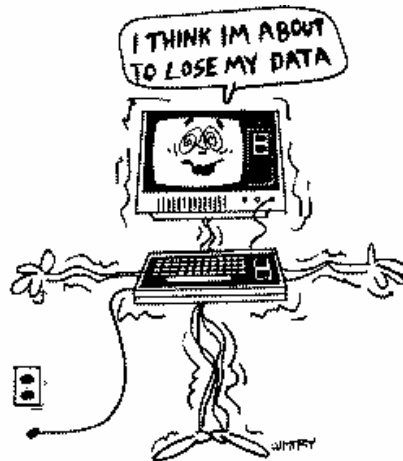
I'LL NEED LOTS OF LUCK TO BE ABLE TO READ THE DISKS. I DON'T KNOW WHETHER THE ENHANCED NEWDOS86 OR NEWDOS90 DISKS ARE AROUND STILL, BUT I NOTICED I HAVE UTILITY DISKS BELONGING TO ND90. I STILL HAVE THE 4P DOWNSTAIRS, WHICH MAY STILL WORK, BUT I WOULD HAVE TO REMOVE THE FLOPPY DRIVES

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 MISOSSYS) 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.



---- 000 ----

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 4Ø2DH        ;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 002BH ;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 (002BH) SCANS THE KEYBOARD & RETURNS ZERO IN REGISTER A IF NO KEY IS PRESSED, ELSE RETURNS THE CHARACTER. USES AF & DE.

INPUT (0049H) SCANS THE KEYBOARD & WAITS FOR A KEY TO BE PRESSED. RETURNS CHARACTER IN A. USES AF & DE.

LININP (0040H) 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 (0033H) PRINTS THE ASCII CHARACTER IN A AT THE CURRENT CURSOR POSITION ON THE VDU. TH CURRENT CURSOR LOCATION IS STORED AT 4020H. USES AF, DE, IY.

CLRSCR (01C9H) 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 = 0DH), OR NULL (00) 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 (402DH) 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   0H
;*** VARIABLES STORED HERE ***
CURLOC DEFW   3C00H      ;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 Z80 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 7000H)

EQU NN - SETS THE VALUE OF A LABEL TO NN, IT CAN ONLY OCCUR ONCE IN ANY PROGRAM (EQU 002BH)

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 00H OR DEFB FFH)

DEFW NN - DEFINES THE CONTENTS OF A TWO BYTE WORD NN (DEFW 3C00H)

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 0 > 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 WHERE WE WANT TO PRINT THE TEXT MESSAGE. REMEMBER THAT THIS IS A MEMORY LOCATION, & THAT THE SCREEN MEMORY BEGINS AT 3C00H SO THE VALUE THAT NEEDS TO BE STORED IS 3C00H PLUS 1DAH (474 BASE10) 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 Z80 ASSEMBLER PROGRAMMING SINCE 1994. & I HADN'T DONE ANY TRS80 ASSEMBLER SINCE THE MID TO LATE 1980'S. SO WHEN I CAME TO REFRESH MY MEMORY BY REREADING THE PRINT-OUTS 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
DSP01   LD     A,(HL)       ;GET MESS CHAR
        OR     A           ;TEST FOR ZERO
        JR     Z,DSP02     ;RETURN IF DONE
        LD     (BC),A       ;STORE CHARACTER
        INC    BC          ;BUMP SCREEN POINTER
        INC    HL          ;BUMP MESSAGE POINTER
        LD     (CURLOC),BC  ;SAVE POINTER
        CALL   DSP01       ;CONTINUE
DSP02   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 DSP01 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 0H), IT JUMPS TO THE LABEL DSP02 (ANOTHER INTERNAL LABEL). THE OR IS A LOGICAL ONE, & IS DONE ON A BIT BY BIT BASIS ON THE REGISTER.

A	B	OR	AND
0	0	0	0
0	1	1	0
1	0	1	0
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 NEXT CHARACTER TO BE DISPLAYED). THE NEXT LINE SAVES THE CURRENT (INCREMENTED) VALUE OF SCREEN MEMORY, FOLLOWED BY A LOOP TO DSP01 IN THE NEXT, REPEATING THE WHOLE PROCESS AGAIN, UNTIL A NULL (0H) CHARACTER IS ENCOUNTERED, AT WHICH POINT A JUMP TO DSP02 IS CARRIED OUT. AT THE LINE DSP02, 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