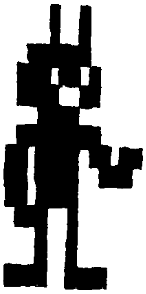
TRS8BIT

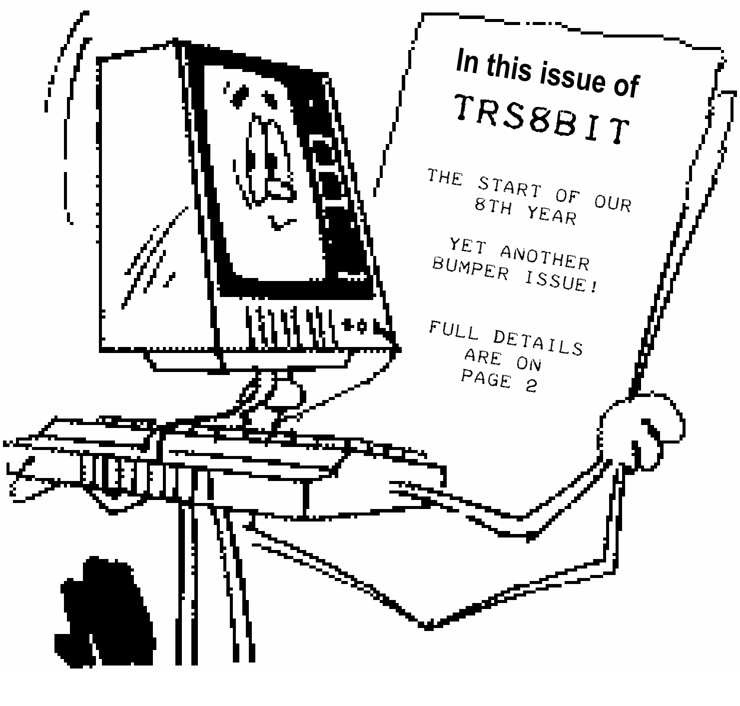


HI EVERYONE, AND WELCOME TO THE MARCH 2014 EDITION OF TRS8BIT AND THE START OF OUR 8TH YEAR!
YET AGAIN, ANOTHER BUMPER EDITION. SO A BIG 'THANK YOU' TO ALL OF YOU WHO HAVE SENT IN CONTRIBUTIONS!

HOWEVER, I REGRET THAT FIRST I'LL START OFF WITH A SAD BIT OF NEWS. IT HAS JUST BEEN ANNOUNCED THAT IAN MCNOUGHT-DAVIS HAS DIED. 'MAC' AS HE WAS KNOWN TO MOST PEOPLE IN THE UK PRESENTED THE BBC SUNDAY MORNING 'MICRO' SHOW IN THE EARLY 1980'S.

I READ A WONDERFUL TRIBUTE OF HIM FROM ONE OF HIS COLLEAGUES.

IT SUMS MAC UP BEAUTIFULLY:



"ANYONE WHO COULD USE THE WORD 'BOLLOCKS' AS HIS COMPUTER PASSWORD ON A LIVE TV PROGRAMME DESERVES MY RESPECT".

I HAD AN EMAIL FROM MALCOM RAMEY MENTIONING DETAILS OF PETER BARTLETT'S 'MISE' PROJECT. I THINK IT'S TRULY BRILLIANT - CHECK IT OUT AT

http://home.comcast.net/~bartlett.p/MISE/

ONCE AGAIN, WITH SO MANY ARTICLES, I'VE USED PAGE 2 AS A CONTENTS LISTING.
MARC BRUMLIK SENT ME AN EMAIL DETAILING MUCH OF THE EXPERIENCE HE ACQUIRED WHILE WORKING FOR RADIO SHACK, REPAIRING MODEL 1'S.
MAV HAS CONTRIBUTED 4 BRILLIANT ARTICLES RANGING FROM THE FREHD, M3'S AND TO DR WHO!
LARRY KRAEMER SHOWS US THE SECRET OF UPDATING M1 ROMS.
KEVIN PARKER SEND US THE FIRST, OF WHAT I HOPE WILL BE MANY, ARTICLE FEATURING THE 'HEAVYWEIGHT' (MODELS 2,12 & 16 ETC) TANDYS.
LAURIE SHIELDS COMES ACROSS THE 'LAST' VERSION OF THE ZEN EDITOR ASSEMBLER, ZEN85 V6.2 (UNLESS, OF COURSE, YOU HAVE A LATER VERSION TUCKED AWAY, IN WHICH CASE, I'D LOVE TO HEAR FROM YOU!!) AND FINALLY, GAZZA WALKS US THROUGH MAKING IMAGE AND .DSK FILES

WELL, THAT'S JUST ABOUT IT FROM ME, I HOPE YOU FIND THIS EDITION AS FASCINATING AS I DO!

BYE FOR NOW
DUSTY

CONTENTS

How to diagnose and repair a TRS-80 Model 1
Marc Brumlik

The following text is a set of notes I compiled in 1980 while working for Tandy Corporation (Radio Shack) as one of their first computer repair technicians. I came across this document in my archives, and felt compelled to post them. The procedures may help you repair almost any problem with the Model 1 at the component level.
Happy fixing! -- Marc Brumlik

(Schematics from the TRS-80 Technical Reference Handbook would be helpful, and an oscilloscope)

ADDRESS LINES

The address lines are outputs from the Z-80 used to specify memory locations. If any address line is not active (stuck on one state), there will be 'garbage' (random characters) on the screen at boot. To check the address lines, remove the DIP Shunt at position Z3 and power-up the system. If the screen fills with a pattern of “@9@9@9@9...”, then all the address lines A0 – A9 are good. The remaining address lines A10 – A15 can be checked for activity with a scope.

If the screen shows “@9@9” in some areas and garbage characters in others, in a repeating pattern, the defective address line can be found easily. For example, if the pattern is alternately four characters of @9 and four of garbage, then address line A2 is stuck because 2^2=4. If the pattern is a full line of @9 (64 characters) followed by a full line of garbage, then address line A5 is stuck because 2^5=64.

If an address line is found to be inactive, the cause us usually the tri-state buffer responsible for that line. For those without a schematic, these are:
A00 – A01 Z55
A02 – A03 Z22
A04 – A09 Z39
A10 – A15 Z38
If replacing the suspected buffer has no effect, cut the foil trace at the input to each of the gates connected to the output of the tri-state buffer for the bad address line, one at a time, checking the output of the buffer each time with a scope. When activity is seen, the offending chip has

been isolated - replace it and repair all the cuts. Don't forget the keyboard, which is also driven by the address buffers. Note that the ROM satellite board for Level II BASIC could also be responsible for inactivity on address lines A11, A12, and A13 even if the ribbon cable has been removed from Z33.

ADDRESS DECODERS

When the address lines specify a memory location then the address decoder then enables one memory area, for example ROM, Video, or keyboard. Z21 and its support chips Z73, Z74, Z36 and Z37 use the high order address lines to decode the signals ROMA*, ROMB*, RAM*, MEM*, VID*, and KYBD*. The outputs of Z21 break down the 32767 memory locations from the CPU into 4K blocks (dec. 0 - 4095, 4096 - 8191, 8192 - 12207, 12208 - 16383, etc). If any of these signals is not present, there will be garbage on the screen, @9@9's, or @S@S's on the screen with Z3 in place.

DATA LINES

The Z-80 uses data lines both to read and write data at any memory address. The control lines (RD*, WR*, IN*, and OUT*) output from Z23, are used in conjunction with the data lines to specify what is to be done with the data bus. Absence of IN* or OUT* will cause cassette I/O to fail, while RD* and WR* will cause garbage on the screen. Any data line without activity will also cause garbage. Remove Z3 and look at the display. An inactive data line will often cause a pattern on the screen like @1@1 or Y9Y9 and the bad bit can easily be found. Since the correct pattern is @9@9, the stuck bit can be determined mathematically (similar to bad address lines above, subtracting the binary of @ or 9 from that of the character seen). The most common cause of a stuck data line is a bad memory chip, the inputs of which are tied directly to the data lines. Removing all memory chips and checking the pattern will identify whether the problem is RAM or some other chip.

VIDEO DIVIDER CHAIN

Most of the time (unless video is in the process of being updated), the VID* signal is inactive , and video ram is being continuously addressed by the video divider chain. The divider chain sequentially accesses all of video memory so that it can be continuously displayed. It is derived from the system clock, and addresses each memory location from beginning to end in sync with the display monitor. Z65 and Z50 output the signals C1 - C32, specifying each of the 64 character positions in one line on the display. These are decoded to video RAM address lines V0 - V5 when the VID* signal is HI.

Z12 provides the signals L1 - L8. L1, L2, and L4 are used by the character generator chip Z29 (RS1, RS2, RS3) to specify one of the seven scan-lines of an alphanumeric character. L4 and L8 are used by Z8 to specify the top-, middle-, or bottom-third of a graphics cell. L8 is also used to create the blank scan lines between lines of alpha-characters (which are seven lines tall, vs 12-line graphics characters). The signals R1, R2, R4, and R8 from Z32 specify one of the 16 lines of characters on the CRT. They are video address lines V6 – V9. If any of the signals V6 – V9 are absent, the CRT will show a smear of dots. If L1 – L8 are missing, the printed characters will be properly located but illegible. Shorts between the lines will produce characters shown in two locations at once. Removing Z3 will show the bad video address lines similarly to bad memory address lines (alternating patterns).

VIDEO RAM

Ram errors are the most common video problem. To isolate the defective chip, first determine the ASCII code of that character that should be shown and the character that appears instead. For example, a blank screen (press CLEAR) is filled with ASCII code 32's (the code for a “space”). If some location shows a “$”, which is ASCII 36, subtract 36-32=4. The location with the error is displaying character 36 instead of character 32 because bit 2 is stuck on (2^2=4). Replace Z46. Similarly, if pressing “B” on the keyboard displays “J” on the screen: B=66, J=74, 74=66=8, and since 2^3=8, replace Z45.

SYSTEM RAM

A bad RAM chip can cause rapidly changing patterns on the screen, garbage on the screen, or a blank screen. An intermittent problem can cause the occasional program-line change or disk reboot. If the system is not operating at all, try substituting all eight chips with known good ones. Check for the presence of CAS*, RAS*, and MUX. No one ram test will find an error 100% of the time. Try various tests. Very intermittent failures can be caused by the multiplexers Z35 and Z51 or the tri-state buffers Z67 and Z68.

TROUBLESHOOTING

GARBAGE ON THE SCREEN

Check for activity of address decoder outputs (RAM*, MEM*, VID*, etc.)
Check control lines at Z23
Remove Z71 – Screen full of @S's suggests RAM errors
Remove Z3 – Screen full of @9's suggests ROM/address errors
Pattern of @9's is stuck A0-A9 address (check A10-A15 with scope)

Check for clock at Z-80 pin 6
Check Z-80 reset lines (pins 16, 17, 18, 24, 26 should all be high, pulses on pin 17)
Check CAS*, RAS*, MUX, and Z69 pin 5
Replace RAM with good set and downgrade to Level 1 BASIC
Try replacing Z35 and Z51, Z67 and Z68, Z-80 CPU, Z21

VIDEO

VIDEO JITTER
Check for output from Z66
Check each gate of Z5, Z6, Z57
Check for intermittent R20, R21
Check each output of video divider chain
NO VIDEO
Check power supply voltages (note that if the 12V line is out, 5V will be to.
Shorted RAM chip can bring down voltages
Check for output at Z30 pin 1. If present, check Z41 and Q1/Q2. (Z41 could be
bad if cassette relay is shorted).
Check Z9 pin 8 for SHIFT. Check output of Z26, pin 8. Check output of Z10 pin 13.
Check inputs RS1, RS2, and RS3 and from Z28.
OTHER
Wrong characters displayed (but recognized as correct by computer) - Video RAM.
Parts of characters illegible or missing - Z29 or RS1, RS2, RS3
No blank lines under alpha characters - DLY L8 inactive.
Video is a smear of dots - video address lines.

CASSETTE

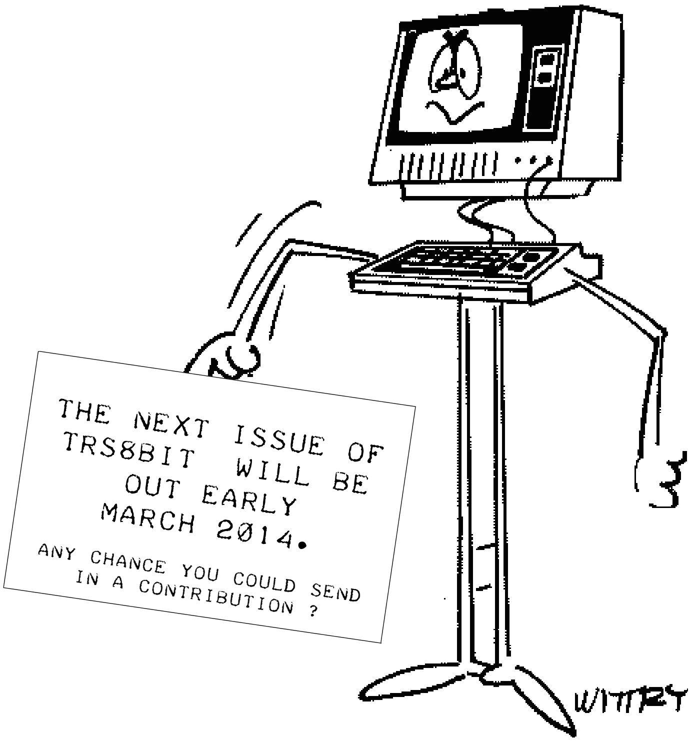
No "CSAVE" -- Replace Z59
Intermittent "CLOAD" -- Check XRXIII, check Z24, replace Z44
No "CLOAD" -- Check output from Z4 pin 10, check Z24 pin 9 "after" XRXIII, check Z24 pin 8, replace Z44
Dim video during CLOAD - Check Z41, K1, or CR3 for short
Tape motor always on - Check Z41, CR9 and CR10 for short, replace K1

HINTS

The input or output of any gate can be pulled low for testing. Pulling signals low can simulate conditions for testing logic. NEVER pull a gate high! If it is ever found that a short has pulled a signal high, replace any chip whose input or output was affected.

If a gate has active inputs and an inactive output, but replacing it did not correct the problem, cut the foil traces at the inputs to gates connected to that output. Check for activity between each cut. If no trace cuts correct the problem, scope the inputs to be certain that logic conditions are being met.

If symptoms suggest a bad gate but there is activity on the output of a suspect, use a scope to determine of that output is correct for that gate's logical function. For example, Z23's gate at pins 4, 5, 6:
put scope CH1 on pin 4, CH2 on pin 5, TRIGGER on pin 6
since the output is active when low, use NEG trigger
the scope will show the state of the inputs whenever the output goes low
(use 2V / division both channels, 1 us timebase)
To check a tri-state buffer like Z76 gate at pins 4 and 5:
CH1 on input pin 4, CH2 on output ping 5, TRIGGER on enable line pin 1
compare CH1 and CH2 for time lag or missed pulses



IN MAV'S WORKSHOP
by
IAN MAVRIC

HOW TO BUILD THE FreHD HARD DRIVE EMULATOR

It's coming up to a year since the FreHD was released to the public, and it's been a great success. If you don't have one, what is stopping you? Over 100 have been made and sold, and most popular product by far being the inexpensive Kit B, which comprises of a PCB, the two important chips (the PIC and GAL) and the SD card socket and the battery socket. Most people who can solder together an electronics project kit like you'd find in Elektor magazine, will be fine with Kit B as all the components are readily available from any electronics shop. In fact if you do a lot of electronics like I do, you may already have many of the resistors, capacitors, LEDs, and headers on hand. Chances are, in this case, you may only need to buy $10 of parts to complete a Kit B if you only need 4 chips, 2 crystals, and 2 transistors. This article will explain how to build Kit B and test it and get it up and running.

GATHER YOUR PARTS

On the CD which comes with your Kit B, in the root directory, is a TXT file called BOM for wd1010_rel1 - USE THIS WHEN BUILDING FreHD.TXT, which is a BOM (Bill of Materials ie. list of the parts) needed to build your FreHD. Many people ask me to email them this file while Kit B is in transit so they can accumulate the parts so they are ready to build the moment their product arrives.
Referring to the TXT file: U1, U2, J2 and the battery connector are included with Kit B and don't need to be ordered. D1 and D2 are LEDs, you can use any colour you like but we prefer Red for D1 and Green for D2. Also, the FreHD has a serial port which was incorporated for debugging the PIC software, and it possible it could be utilised by the TRS-80 in the future, but at the time of writing it's just on there for a future project. If you don't want the serial port you can leave out U5 and C6, C7, C8, C10 and C11. Finally U3, the BOM states it should be a 74HCT245, which is correct when using the FreHD but if you are building FreHD to use on a Model I via a hard drive adapter, you need to change U3 to a 74LS245.

The power connector is some source of questions to me, so I'll try to explain it.. On version 3.01 PCBs there is a 2-pin power connector, +5V and Gnd. They are clearly marked. Since there are no more 3.01 boards, the current production run is 3.02 which has a new 4-pin power connector. I chose to make it compatible with the power connector from a 3.5" floppy drive. In this case the +5V in is next to the 50-in I/O bus connector. Although there is a pin for the 12V line from the floppy drive power connector, it doesn't go anywhere. The two middle pins are Gnd.

WARM UP YOUR SOLDERING IRONS

These notes are from my building some 25 of the Kit C (complete) FreHDs, this is the sequence I use:

1. Start by soldering down the SD-card socket. It's surface mount, and you need good access to all its tiny tabs which can be difficult if nearby components are already in place.
2. I proceed to soldering in all the resistors and small capacitors (C1, C2, C3, C4, C9) and other low profile parts, like the sockets for the ICs, and crystal Y2.
3. 10MHz crystal Y1 should be mounted with a crystal spacer or just solder it slightly proud of the PCB so its metal canister doesn't short on anything.
4. Q1 and U4 are then soldered in, their flat sides facing R15.
5. Headers for J1 and J5 can be left out. The J1 header is a PICkit 2/3 compatible programmer connection which can be used to reprogram the PIC. Since most people use FUPDATE/CMD to reprogram the PIC, only people who use a PICkit 2/3 may want to install the J1 header. Similarly the J5 header is part of the currently-unutilised serial port and can be left out.
6. Lastly I install the tall electrolytic capacitors C5 and C12, and the two LEDs D1 and D2. The green one is D2, it's installed closest to the battery.

TESTING YOUR HANDY WORK

Before installing the ICs a good idea is to connect your +5V power source and check for +5V on the power pins of the ICs and +3.3V on the power tab on the SD card socket.
The power pins are as follows:
U1: +5V on Pins 11 & 32 and Gnd on Pins 12 & 31
U2 and U3: +5V on Pin 20 and Gnd on Pin 10
U5: +5V on Pin 16 and Gnd on Pin 15
U6: +5V on Pin 16 and Gnd on Pin 8
U7: +5V on Pin 8 and Gnd on Pin 4
J2 (SD Card Socket): On Tab 4 observe +3.9V with no SD card installed and +3.3V with the SD card inserted

We are nearly finished, so now is time to power-down and insert all the ICs. Bear in mind that U2, U3, U6 and U5 all face one way and U1 & U7 face the other - this is normal. Before connecting the FreHD to a TRS-80, power it up without an SD card inserted and observe the green LED flash once. Insert your prepared SD card, and notice the green LED flash once more. If you are using a small-capacity HP brand SD card there may be a delay of 3 seconds before the green LED flashes upon inserting. Most other SD cards flash the green LED in less than 1 second.

Power everything down and connect it to your TRS-80. Power up the FreHD, insert your prepared SD card, and power up your TRS-80. The TRS-80 should operate normally. If it crashes or won't start up examine your hard disk drive cable and make sure it isn't plugged in around the wrong way. (Having it plugged around the wrong way won't harm anything but it won't work either.) Proceed to preparing your boot floppy disk if you require one (TRS-80 Model I,III,4,4D) or copy the correct auto-boot image file to your SD card (Model 4P).

PREPARING YOUR SD CARD

The SD card needs to be prepared before use. It will probably be formatted in FAT and needs to be reformatted in FAT32. Also while formatting it, give the card a volume name. Any name will do, but I tend to name all mine TRS80HDD1, TRS80HDD3, TRS80HDD4, or TRS80HDD4P to remind me which system I'm intending to use it on. Next, referring to the CD supplied with your Kit B, look for the image file to match your system. Folder 'LS-DOS M4 Image' contains the LS-DOS 6.3.1 image for TRS-80 Model 4 system, and other image folders are similarly named for other TRS-80s and supported DOSes. In each instance the file name is hard4-0 and you need to copy that to the SD card. This is the image file and co-incidentally the same size as the TRS-80 hard drive it emulates: 40Mb. The only exception to this is the CP/M 2.2 image which comprises of two 15Mb files called hard4-0 and hard4-1 - make sure you copy both those files to your SD card if you want to use CP/M.

PREPARING YOUR STARTUP DISKETTE - 4P OWNERS CAN SKIP THIS SECTION

All TRS-80 Model I, III, 4 and 4D systems, in their standard form, need a boot floppy to get the system up and running. Radio Shack calls this a "Startup Disk" and back when they sold their TRS-80 hard drives, explained how the Startup Disk was generated in the manuals supplied with their hard disk systems. You used this disk at the start of the day to boot your hard disk system, then remove the disk and work off the hard drive for the rest of the day.

Model I: Looking through the supplied CD, you'll find for the Model 1 there are three DMK diskette image files in the LDOS M1 Image folder. One is for Model Is with No doubler, another is for Model Is with a Tandy doubler, and the third is for Model Is with a Percom doubler. Use a Catweasel to make the boot disk you require and you are set.

Model III (LDOS): In the root directory of the supplied CD, look for a file called FreHD RSHARD6 Setup.pdf In order to make the required boot disk make sure you have a LDOS 5.3.1 master disk and a copy of RSHARD5/DCT and RSFORM5/CMD and just follow the steps in the PDF, changing references RSHARD6 to RSHARD5 and references to RSFORM6 to RSFORM5 and you will be able to make your Startup disk.

Model III (Newdos/80 2.5): In the Newdos M3 Image folder look for two DMK files disk3-0 and disk3-1, just use a Catweasel to make the boot disk you require and you are set.

Model 4 (LS-DOS): In the root directory of the supplied CD, look for a file called FreHD RSHARD6 Setup.pdf In order to make the required boot disk make sure you have a LS-DOS 6.3.1 master disk and a copy of RSHARD6/DCT and RSFORM6/CMD and just follow the steps in the PDF.

Model 4 (CP/M 2.2): In the CP/M image folder in the root directory of the supplied CD, look for two DMK files disk4-0 and disk4-1, just use a Catweasel to make the boot disk you require and you are set.

4P OWNERS - NO STARTUP DISK REQUIRED

The designers of the 4P put an undocumented hard drive boot routine into the ROM of every 4P. Radio Shack didn't tell you about it but every 4P has it. FreHD design/debugging team exploits this feature by presenting a special 4P LS-DOS 6.3.1 image (in the LS-DOS 4P AUTO BOOT IMAGE folder) - just copy that one to your SD card, power up your FreHD and power up your 4P about 5 seconds later, it will automatically boot from the FreHD.

The same applies to CP/M 2.2 on the 4P, just copy the hard4-0 and hard4-1 image files (15Mb each) from the CPM Image folder to the SD card and start your 4P about 5 seconds after FreHD - auto instant boot CP/M.

IN CONCLUSION

Kit B provides a way for most people with a technical background to get fast, reliable mass storage for a very modest price. In fact even the full Kit C at $A250 is price competitive with similar solid-state storage systems for other 8-bit micro computer platforms, and around half the cost of a restored TRS-80 MFM hard disk drive.

My offer on my web site still stands, anyone who buys Kit B and can't get it working can send it to me and I will make it work. So far only two people have had to send me their FreHDs and I was able to diagnose and fix the problem in under 30 minutes, and return them a fully functional FreHD for a small fee to cover my time, parts and return postage.

ianm@trs-80.com

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.

Radio Shack Level II ROM UPGRADE to Version 1.3

Larry Kraemer

The Radio Shack Model 1's were shipped with a 3 Chip set of ROMS. Those ROMS are Pin compatible with the Texas Instrument TMS-2532JL-45 EPROMS. Some of the later Model 1's were shipped with a 2 Chip ROM set, as were some of the "G" Version Boards. I have a "G" version board with a 3 Chip ROM set.

This is an interesting read on the TI vs Intel EPROMS.
http://www.tomshardware.com/forum/80532-13-what-diff-eprom-eprom

SOURCE: Micro-80-Volume-01-Issue10

Tandy started shipping 2 chip ROM sets in their Model 1's. Now that 2 chip ROM's are here, a few things have changed. First of all, there are two different sets of ROM's.

Part Nos. 8043364 and 8043732 are for all except "G" boards and these ROM's still need to be used in conjunction with the XRX III loading modification.

Part Nos. 8044364 and 8044732 are for the "G" boards and do not require the XRX III loading modification.

The checksums for these latter ROM's are:
ROM A B078
ROM B DA45
ROM C 4006

ROM A and ROM B are in the first chip (8K), ROM C is in the second chip (4K).

My older Model 1 has a 3 Chip ROM set, and I wanted to upgrade it to Version 1.3. The ROMS are numbered from Right to Left as ROM3, ROM2, ROM1, with ROM1 being closest to the Ribbon Cable.

I purchased some TMS-2532JL-45 EPROMS and erased them. When the EPROM is ERASED it contains all ONES.
Pin 21 of the EPROM doesn't get inserted into the Socket. Instead it's bent slightly and I wire wrapped and soldered a wire from it to a Pull-up resistor I inserted in the EPROM Board. (This copies what ROM3

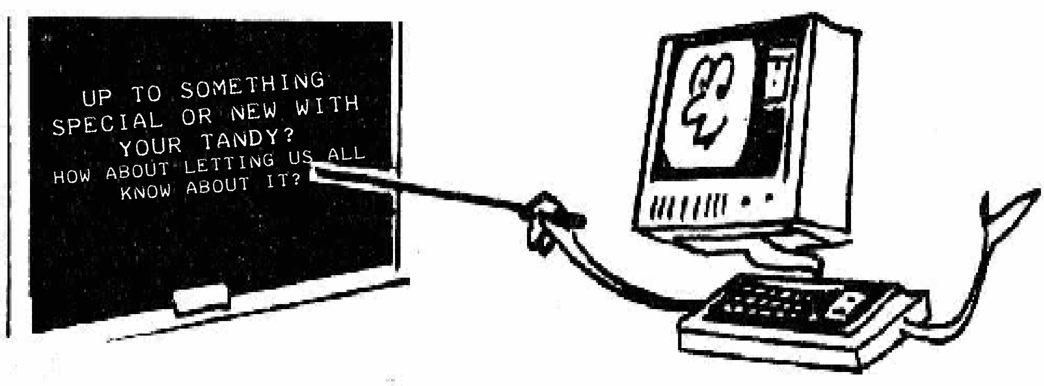
has on Pin 21 for enable, except it's factory wiring for ROM3).

There should be a command in your EPROM Programming Software to make sure it is properly erased. I used an EP1 programmer from BP Microsystems to do my EPROM burning.

If my Model 1 would boot I could have used Tasmon to move the ROM's HEX Code to 0x7000, and then write it to a disk file by using some assembly code like this:

```
LD HL,0x0000        ;point at SOURCE
LD DE,0x7000        ;point at DESTINATION
LD BC,0x3FFF        ;Number of bytes to move
DJNZ                ;move 'em all
LD A,01             ;Breakpoint junk instruction
NOP
```

If you have a Model 1, Revision "G" with two ROM Chips, you can read those into one file and then create one 8K file and one 4K file. The TMS2564 EPROMS are Pin Compatible for 24 out of the 28 Pins. The TOP Four Pins won't be inserted in the ROM Sockets. Power will be on Pin 24 instead of Pin 28. Check the Note. The Chip Select Pins will have to be wired to the proper signals. Vpp Might also need to be tied to a Pull-up Resistor using +5VDC.



My Model 1 wouldn't power up, so I needed to find a ROMIMAGE. I stumbled across these two sites.

http://www.vintagecomputer.net/fjkraan/comp/trs80/
http://www.vintagecomputer.net/fjkraan/comp/trs80/trs80Archives_diskpacks/

These diskpacks have similar contents, and a ROMIMAGE included in each.

```
Archive:  softinno.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
    21324  1994-09-19 23:54   MODEL1.EXE
     3074  1993-07-01 02:13   MODEL1.FNT
      234  1996-05-07 20:08   MODEL1.SET
   102400  1996-05-07 20:07   BOOT.DSK
   102400  1996-05-07 20:06   MICRO.DSK
    12288  1993-09-24 20:30   ROMIMAGE  (Model 1 - Version 1.3)
---------                     -------
   241720                     6 files
```

But, since I didn't know this ROMIMAGE was Version 1.3, I used my previous information to VERIFY each BYTE that was different than my Original ROMS.

That information is in the file roms.txt, but I needed the data in HEX and addresses in HEX, so I inserted the Decimal addresses and decimal data in LibreCalc (Linux) and converted it to HEX Address & Data. Model1ROM-13.xls

It took me a while to verify the ROMIMAGE is the latest Version 1.3 (identical to the image I burned in my Model 1). The name is ROMIMAGE.HEX. It's the three (4K) ROM's CODE in one BLOCK for Version 1.3 as one continuous HEX Byte file from 0x0000 to 0x2FFF containing all three (4K) ROMS for Radio Shack Model 1's.

ROM1 0x0000 to 0x0FFF (0 to 4095 = 4096 Bytes = 4K Each EPROM)
ROM2 0x1000 to 0x1FFF
ROM3 0x2000 to 0x2FFF

I've tried using CP/M Version 2.2 to extract the three ROM Images. I finally figured out how to use DDT to CROP the ROMIMAGE file, so I could make three (4K) HEX images.

Here is how I did it:

Ran CPM 2.2 in Linux, and copied the ROMIMAGE.HEX file to CP/M's B: Drive. I also copied DDT.COM there.
I couldn't read the ROMIMAGE.HEX file with DDT, so I renamed it to ROM with no extension. DDT was now satisfied.

Start of LOOP:
I executed DDT which gave me the - command prompt.

I did the following commands from DDT's prompt:
```
-f0100,7fff,00       ; Fill from 0x0100 to 0x7fff with
zero's, to make sure I know what is what!
-iROM                ; I had  to delete the .HEX so I
shortened file name to ROM - Input filename
-r0f00               ; read with offset of 0f00 to place at
1000 through 3fff for cropping into three files
-m1000,1fff,0100     ; move 4096 bytes to 0100 through end
of 4096 bytes for tpa and save
CNTL C               ; EXIT DDT and save the block by using:
SAVE 16 ROM1.HEX
```

Repeat all above commands for 2000,2fff,0100 & save as ROM2.HEX - go to LOOP:
Repeat all above commands for 3000,3fff,0100 & save as ROM3.HEX - go to LOOP:

That should do it. I spot checked the first sixteen bytes and the last 16 bytes, and they are correct.

Burn EPROMS & test in a Model 1.

Larry Kraemer
ldkraemer@gmail.com

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.

R/S ROM TO EPROM

Would you like to stop a lot of Cload trouble? Have you ever wanted to update your ROM to Version 1.3? Would you like to personalize your machine by adding your initials, or write your own Microcode? If you answered any of the above questions with yes, then read on. This article will explain how to do what you want with a minimum of trouble.

I will explain the procedures for changing the old R/S ROMS to Eproms, with the code from Version 1.3. This will apply to Model I Level II Version D boards with a three chip ROM set. This will be different on an earlier Version board, or a G model board. Check your board carefully before going on. All this can be done in five easy steps.

First, I identified the Version ROM I had. Second, I found the locations in my ROM which were different from Version 1.3, and patched them for Version 1.3. Third, I reprogrammed Eproms to replace ROMS. Fourth, I installed ROMS and tested. Fifth, I sleep good now, and dream of programs not checksums or bad Cloads.

The first step identifies the ROM Version, and can be accomplished by typing in and running Program #1. I ran the program on my ROMS, and found the checksums. They are listed in Program #1 also. Note that the new Version 1.3 ROMS are marked by an *. Future testing with most R/S tests will show the ROMS to be bad. This is because the latest checksums are not in the test programs. They are listed in the newer books.

The second step is to find the differences in your Version of ROM and Version 1.3. Program #2 will do this for you. If you do not have a printer change the Lprint lines to Rem lines.

You will need to find someone with a Version 1.3 machine, and move the code from 0000H thru 3000H into Ram. This is easily accomplished with Zbug or Tasmon. On my 48K machine I put it at 7000H (B=28672). This can be put on tape, or disk and then loaded in your machine. Now run Program #2 and watch the differences print out. My old ROMS were Version 1.1, and the differences are shown in Figure #1. Your locations will be different for ROMS other than Version 1.1. If you want to change the R/S to your initials, or change any of the Microcode do it at this time.

Step number three requires using an Eprom programmer to dump the new code to Eproms. I can program them for a small fee if you supply the Eproms, and the code.

Now that you know all the differences proceed with step four. The Eproms I used were Texas Instrument TMS-2532 (450-nano second chips). They have the same pinout as the R/S ROM and are 4k x 8 bit chips. Before installing the Eproms, take a good look at the pinout of the new Eproms (Figure #2), and the pinout of the old ROMS in the technical manual. There is one minor difference in that the chip enable pin (pin 21) must be held high to enable this chip. After checking the ROM pinout in the technical manual you will notice that two chips (Z1, Z2) will ahve to be modified for a high on the enable pin of the new Eproms. I did this by bending pin #21 up on both chips, and not inserting it into the socket. A small wire can be wire wrapped from this pin to the pullup resistor going to pin #21 on chip Z3. Note that chip Z1 is toward the outside right of the keyboard with Z2 in the center and Z3 next to the the keys. This will enable you to replace the old ROMS at anytime if you suspect trouble. With all this in mind insert the new Eproms, and proceed with the testing. Run a few test after turning on power to ensure all is ok. Now run Program #1 and note the new ROM Version. The checksum test will also show as listed in Program #1 if no mistakes were made, or no other changes were made.

Step five is the most important since a good nights sleep will let you find all the programming errors you made the night before, without a lot of Cload problems.

I have had a set of Eproms installed in my Model I for over two years, and have had no trouble. Now my machine powers up with MEM SIZ and R/S Level II.

PROGRAM #1

```
10 'THIS PROGRAM CHECKS TO SEE WHAT ROM IS RUNNUNG
20 'ROM 1.0 = 176  ROM 1.1 = 142  ROM 1.2 = 10  ROM 1.3 = 162
30 'VALID CHECKSUMS FOR ROMS A,B,C ARE:
40 'ROM A  AE5D, AE60, B078*
50 'ROM B  DAB4, DA45*
60 'ROM C  4002, 40BA, 40E0, 3E3E, 4006*
70 'THE LAST VALUE (B078*) IS THE NEW ROM CHECKSUM FOR ROM 1.3
80 'SOME OF THE R/S TEST WILL NOT TEST FOR THESE CHECKSUMS AND
90 'WILL SHOW THE ROM AS BAD IF THE TEST IS RUN
100 FOR I=11264 TO 12287:V=PEEK(I):S=S+V:NEXT I:X=S/16
110 A=(X-FIX(X))*16:Y=FIX(X)/16:B=(Y-FIX(Y))*256
120 PRINT (A+B)
```

PROGRAM #2

```
10 CLS:A=0:B=28672:REM COMPARE WITH 7000H
20 FOR I=00 TO 12287
40 IF B>32767 THEN C=PEEK(-1*(65536-B)):A=PEEK(I):GOTO 50
41 A=PEEK(I):C=PEEK(B)
50 IF A=C THEN GOTO 100
60 PRINT@10,I:PRINT@30,A;"      ";C
61 LPRINT,I;A;C
70 FOR J=1 TO 200:
71 PRINT:NEXT J
100 B=B+1:NEXT I
150 LPRINT,"DONE"
160 PRINT@10,"DONE"
200 END
```

Figure #1

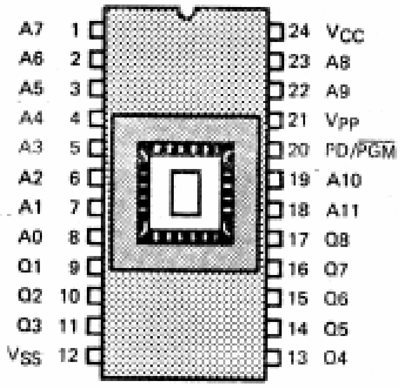
Location	1.1	1.3	Location	1.1	1.3
89	26	0	1020	7	28
253	17	14	1021	7	1
264	79	32	4684	183	209
265	82	83	4685	209	183
266	89	73	8301	155	124
267	32	90	8307	64	35
268	83	69	8309	25	6
269	73	0	8311	1	132
270	90	82	8312	43	2
271	69	47	8313	254	50
272	0	83	8314	4	156
273	82	32	8315	210	64
274	65	76	8316	74	43
275	68	50	8317	30	215
276	73	32	8318	209	204
277	79	66	8319	33	254
278	32	65	8320	0	32
280	72	73	8321	60	202
281	65	67	8322	25	105
282	67	13	8323	34	33
283	75	0	8324	32	256
284	32	197	8325	64	32
285	76	1	8326	123	254
286	69	0	8327	230	96
287	86	5	8328	63	32
288	69	205	8329	50	27
289	76	96	8330	166	205
290	32	0	8331	64	1
291	73	193	8332	225	43
292	73	10	8333	207	254
293	32	163	8334	44	4
294	66	200	8335	254	210
295	65	122	8336	35	74
296	83	7	8337	32	30
297	73	7	8338	8	229
298	67	195	8339	205	33
299	13	254	8340	192	0
300	0	3	8341	2	60
585	65	96	8342	62	25
592	118	133	8343	128	34
1019	122	195	8344	50	32

Location	1.1	1.3
8345	156	64
8346	64	123
8347	43	230
8348	215	63
8349	204	50
8350	254	166
8351	32	64
8352	202	225
8353	105	207
8354	33	44
8355	254	24
8356	191	199
8357	202	126
8358	189	254
8359	44	191
8360	254	202
8361	188	189
8362	202	44
8363	55	254
8364	33	188
8365	229	202
8366	254	55
8367	44	33
8368	202	229
8369	8	254
8370	33	44
8371	254	40
8372	59	83
8373	202	254
8374	100	59
8375	33	40
8376	193	94
8380	229	227
8439	155	124
8507	63	127
8551	160	129
8810	58	0
8811	169	0
8812	64	0
8813	183	0
8814	200	0

Location	1.1	1.3
11295	205	214
11296	147	178
11297	2	40
11298	126	2
11299	214	175
11300	178	1
11301	40	47
11302	2	35
11303	175	245
11304	1	126
11305	47	183
11306	35	40
11307	245	7
11308	43	205
11309	215	55
11310	62	35
11311	0	205
11312	40	19
11313	7	42
11314	205	26
11315	55	111
11316	35	241
11317	205	183
11318	19	103
11319	42	34
11320	26	33
11321	111	65
11322	241	204
11323	183	77
11324	103	27
11325	34	33
11326	33	0
11327	65	0
11328	204	205
11329	77	147
11330	27	2
12284	0	195
12285	0	195
12286	0	68
12287	0	178

- Organization . . . 4K X 8
- Single +5 V Power Supply
- Pin Compatible with Existing ROMs and EPROMs (8K, 16K, 32K, and 64K)
- JEDEC Standard Pinout
- All Inputs/Outputs Fully TTL Compatible
- Static Operation (No Clocks, No Refresh)
- Max Access/Min Cycle Time . . . 450 ns
- 8-Bit Output for Use in Microprocessor-Based Systems
- N-Channel Silicon-Gate Technology
- 3-State Output Buffers
- 40% Lower Power
  TMS 25L32 . . . 500 mW Max Active
  TMS 2532 . . . 840 mW Max Active
- Guaranteed DC Noise Immunity with Standard TTL Loads
- No Pull-Up Resistors Required

24-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A(N)	Address Inputs
PD/$\overline{PGM}$	Power Down/Program
Q(N)	Input/Output
V_{CC}	+5 V Power Supply
V_{PP}	+25 V Power Supply
V_{SS}	0 V Ground

Figure #2

description

MOS
LSI

TMS 2532-30 JL, TMS 2532-35 JL, TMS 2532-45 JL, TMS 25L32-45 JL
32,768-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORIES

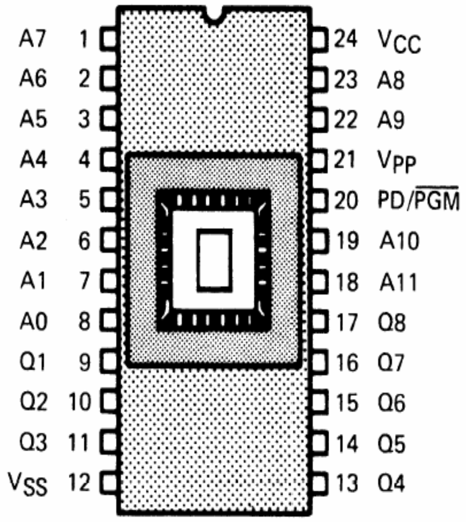
DECEMBER 1979—REVISED MAY 1982

- **Organization . . . 4096 X 8**
- **Single +5 V Power Supply**
- **Pin Compatible with Existing ROMs and EPROMs (8K, 16K, 32K, and 64K)**
- **JEDEC Standard Pinout**
- **All Inputs/Outputs Fully TTL Compatible**
- **Static Operation (No Clocks, No Refresh)**
- **Max Access/Min Cycle Time:**

TMS 2532-30	300 ns
TMS 2532-35	350 ns
TMS 2532-45	450 ns
TMS 25L32-45	450 ns

- **8-Bit Output for Use in Microprocessor-Based Systems**
- **N-Channel Silicon-Gate Technology**
- **3-State Output Buffers**
- **40% Lower Power**
  TMS 25L32 . . . 500 mW Max Active
  TMS 2532 . . . 840 mW Max Active
- **Guaranteed DC Noise Immunity with Standard TTL Loads**
- **No Pull-Up Resistors Required**

24-PIN CERAMIC DUAL-IN-LINE PACKAGE (TOP VIEW)



PIN NOMENCLATURE	
A(N)	Address inputs
PD/$\overline{PGM}$	Power Down/Program
Q(N)	Input/Output
V_{CC}	+5 V Power Supply
V_{PP}	+25 V Power Supply
V_{SS}	0 V Ground

description

The TMS 2532 series (TMS 2532-30 JL, TMS 2532-35 JL, TMS 2532-45 JL, and TMS 25L32-45 JL) are 32,768-bit, ultraviolet-light-erasable, electrically programmable read-only memories. These devices are fabricated using N-channel silicon-gate technology for high speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can be driven by Series 74 TTL circuits without the use of external pull-up resistors, and each output can drive one Series 74 TTL circuit without external resistors. The data outputs are three-state for connecting multiple devices to a common bus. The TMS 2532 series are plug-in compatible with the TMS 4732 32K ROM. The devices are offered in a dual-in-line ceramic package (JL suffix) rated for operation from 0°C to 70°C.

Since these EPROMs operate from a single +5 V supply (in the read mode), they are ideal for use in microprocessor systems. One other (+25 V) supply is needed for programming but all programming signals are TTL level, requiring a single 50 ms pulse. For programming outside of the system, existing EPROM programmers can be used. Locations may be programmed singly, in blocks, or at random. Total programming time for all bits is 200 seconds.

TMS 2532-30 JL, TMS 2532-35 JL, TMS 2532-45 JL, TMS 25L32-45 JL
32,768-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORIES

operation

FUNCTION (PINS)	MODE				
	Read	Output Disable	Power Down	Start Programming	Inhibit Programming
PD/$\overline{PGM}$ (20)	V_{IL}	V_{IH}	V_{IH}	Pulsed V_{IH} to V_{IL}	V_{IH}
V_{PP} (21)	+5 V	+5 V	+5 V	+25 V	+25 V
V_{CC} (24)	+5 V	+5 V	+5 V	+5 V	+5 V
Q (9 to 11, 13 to 17)	Q	HI-Z	HI-Z	D	HI-Z

read/output disable

When the outputs of two or more TMS 2532's are connected on the same bus, the output of any particular device in the circuit can be read with no interference from the competing outputs of the other devices. The device whose output is to be read should have a low-level TTL signal applied to the PD/$\overline{PGM}$ Pin. All other devices in the circuit should have their outputs disabled by applying a high-level signal to this pin. Output data is accessed at pins Q1 through Q8. Data can be accessed in 450 ns = $t_{a(A)}$.

power down

Active power dissipation can be cut by over 70% by applying a high TTL signal to the PD/$\overline{PGM}$ pin. In this mode all outputs are in a high-impedance state.

erasure

Before programming, the TMS 2532 is erased by exposing the chip through the transparent lid to high-intensity ultraviolet light having a wavelength of 253.7 nm (2537 angstroms). The recommended minimum exposure dose (UV intensity times exposure time) is fifteen watt-seconds per square centimeter. Thus, a typical 12 milliwatt per square centimeter, filterless UV lamp will erase the device in a minimum of 21 minutes. The lamp should be located about 2.5 centimeters above the chip during erasure. After erasure, all bits are in the "1" state (assuming high-level output corresponds to logic "1").

start programming

After erasure (all bits in logic "1" state), logic "0's" are programmed into the desired locations. A "0" can be erased only by ultraviolet light. The programming mode is achieved when Vpp is 25 V. Data is presented in parallel (8 bits) on pins Q1 through Q8. Once addresses and data are stable, a 50-millisecond TTL low-level pulse should be applied to the $\overline{PGM}$ pin at each address location to be programmed. Maximum pulse width is 55-milliseconds. Locations can be programmed in any order. Several TMS 2532's can be programmed simultaneously when the devices are connected in parallel.

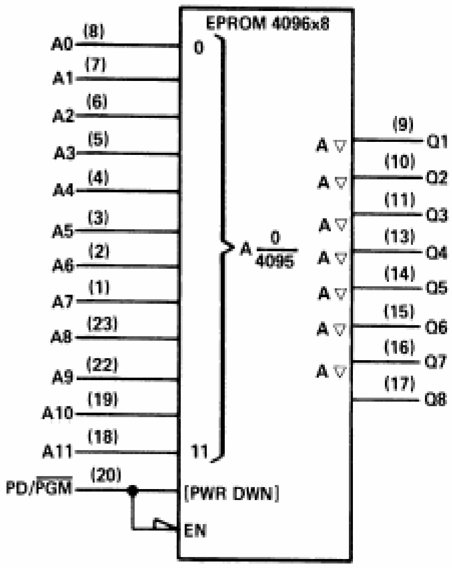
inhibit programming

When two or more devices are connected in parallel, data can be programmed into all devices or only chosen devices. Any TMS 2532's not intended to be programmed should have a high level applied to PD/$\overline{PGM}$.

program verification

The TMS 2532 program verification is simply the read operation, which can be performed as soon as V_{PP} returns to +5 V ending the program cycle.

logic symbol†



† This symbol is in accordance with IEEE Std 91/ANSI Y32.14 and recent decisions by IEEE and IEC. See explanation on page 289.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)*

Supply voltage, V_{CC} (see Note 1)	−0.3 to 6 V
Supply voltage, V_{PP} (see Note 1)	−0.3 to 28 V
All input voltages (see Note 1)	−0.3 to 6 V
Output voltage (operating with respect to V_{SS})	−0.3 to 6 V
Operating free-air temperature range	0 °C to 70 °C
Storage temperature range	−55 °C to 125 °C

NOTE 1: Under absolute maximum ratings, voltage values are with respect to the most negative supply voltage, V_{SS} (substrate).

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

TMS 2532-30 JL, TMS 2532-35 JL, TMS 2532-45 JL, TMS 25L32-45 JL
32,768-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORIES

recommended operating conditions

	TMS 2532-45			TMS 2532-30 TMS 2532-35 TMS 25L32-45			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC} (see Note 2)	4.75	5	5.25	4.75	5	5.25	V
Supply voltage, V_{PP} (see Note 3)		V_{CC}			V_{CC}		V
Supply voltage, V_{SS}		0			0		V
High-level input voltage, V_{IH}	2.2		$V_{CC}+1$	2		$V_{CC}+1$	V
Low-level input voltage, V_{IL}	−0.1		0.65	−0.1		0.8	V
Read cycle time, $t_{c(rd)}$	450			450			ns
Operating free-air temperature, T_A	0		70	0		70	°C

NOTES 2. V_{CC} must be applied before or at the same time as V_{PP} and removed after or at the same time as V_{PP}. The device must not be inserted into or removed from the board when V_{PP} is applied.

3. V_{PP} can be connected to V_{CC} directly (except in the program mode). V_{CC} supply current in this case would be $I_{CC} + I_{PP}$. During programming, V_{PP} must be maintained at 25 V (± 1 V).

electrical characteristics over full ranges of recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS 2532-30 TMS 2532-35 TMS 2532-45			TMS 25L32-45			UNIT
			MIN	TYP†	MAX	MIN	TYP†	MAX	
V_{OH}	High-level output voltage*	$I_{OH} = -400\ \mu A$	2.4			2.4			V
V_{OL}	Low-level output voltage*	$I_{OL} = 2.1$ mA			0.45			0.45	V
I_I	Input current (leakage)	$V_I = 0$ V or 5.25 V			±10			±10	μA
I_O	Output current (leakage)	$V_O = 0.4$ V or 5.25 V			±10			±10	μA
I_{PP1}	V_{PP} supply current	$V_{PP} = 5.25$ V, $PD/\overline{PGM} = V_{IL}$			12			12	mA
I_{PP2}	V_{PP} supply current (during program pulse)	$PD/\overline{PGM} = V_{IL}$			30			30	mA
I_{CC1}	V_{CC} supply current (standby)	$PD/\overline{PGM} = V_{IH}$		10	25		10	25	mA
I_{CC2}	V_{CC} supply current (active)	$PD/\overline{PGM} = V_{IL}$		80	160		65	95	mA

*AC and DC measurements are made at 10% and 90% points using a 50% pattern.

capacitance over recommended supply voltage and operating free-air temperature ranges, f = 1 MHz‡

PARAMETER		TEST CONDITIONS	TYP†	MAX	UNIT
C_i	Input capacitance	$V_I = 0$ V, f = 1 MHz	4	6	pF
C_o	Output capacitance	$V_O = 0$ V, f = 1 MHz	8	12	pF

†Typical values are $T_A = 25$ °C and nominal voltages.

‡Capacitance measurements are made on a sample basis only.

switching characteristics over full ranges of recommended operating conditions (see note 4)

PARAMETER		TEST CONDITIONS (See Notes 4 & 5)	TMS 2532-30			TMS 2532-35			TMS 25L32-45 TMS 2532-45			UNIT
			MIN	TYP†	MAX	MIN	TYP†	MAX	MIN	TYP†	MAX	
$t_{a(A)}$	Access time from address	$C_L = 100$ pF, 1 Series 74 TTL load, $t_r \leq 20$ ns, $t_f \leq 20$ ns, See Figure 1			300			350		280	450	ns
$t_{a(PR)}$	Access time from PD/$\overline{PGM}$				300			350		280	450	ns
$t_{v(A)}$	Output data valid after address change								0			ns
t_{dis}	Output disable time from PD/$\overline{PGM}$ ‡				100			100	0		100	ns

† All typical values are at $T_A = 25\,°C$ and nominal voltages.
‡ Value calculated from 0.5 volt delta to measured output level.

recommended timing requirements for programming $T_A = 25\,°C$ (see note 4)

PARAMETER		MIN	TYP†	MAX	UNIT
$t_{w(PR)}$	Pulse width, program pulse	45	50	55	ms
$t_{r(PR)}$	Rise time, program pulse	5			ns
$t_{f(PR)}$	Fall time, program pulse	5			ns
$t_{su(A)}$	Address setup time	2			μs
$t_{su(D)}$	Data setup time	2			μs
$t_{su(VPP)}$	Setup time from V_{PP}	0			ns
$t_{h(A)}$	Address hold time	2			μs
$t_{h(D)}$	Data hold time	2			μs
$t_{h(PR)}$	Program pulse hold time	0			ns
$t_{h(VPP)}$	V_{PP} hold time	0			ns

† Typical values are at nominal voltages.

NOTES: 4. For all switching characteristics and timing measurements, input pulse levels are 0.65 V to 2.2 V and $V_{PP} = 25\text{ V} \pm 1$ V during programming. All AC and DC measurements are made at 10% and 90% points with a 50% pattern.
5. Common test conditions apply for t_{dis} except during programming. For $t_{a(A)}$ and t_{dis}, PD/$\overline{PGM} = V_{IL}$.

PARAMETER MEASUREMENT INFORMATION

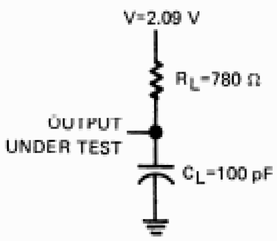
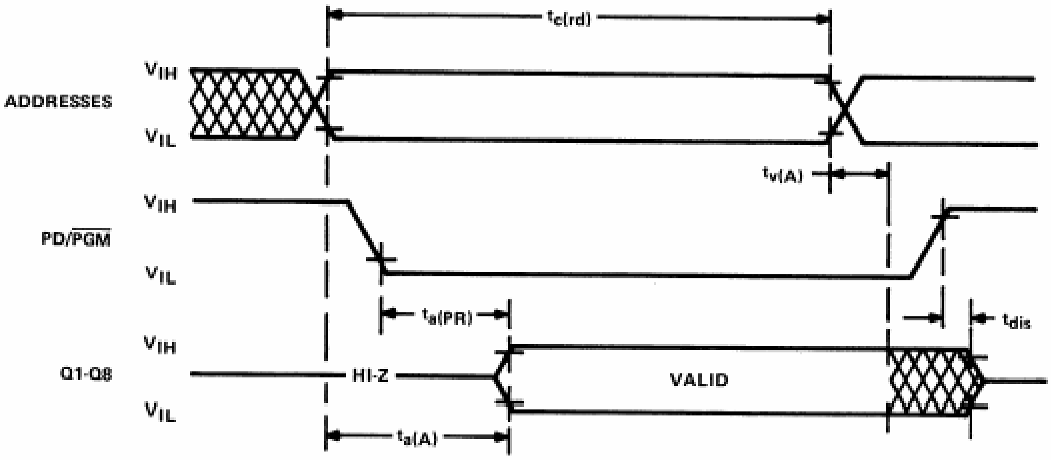


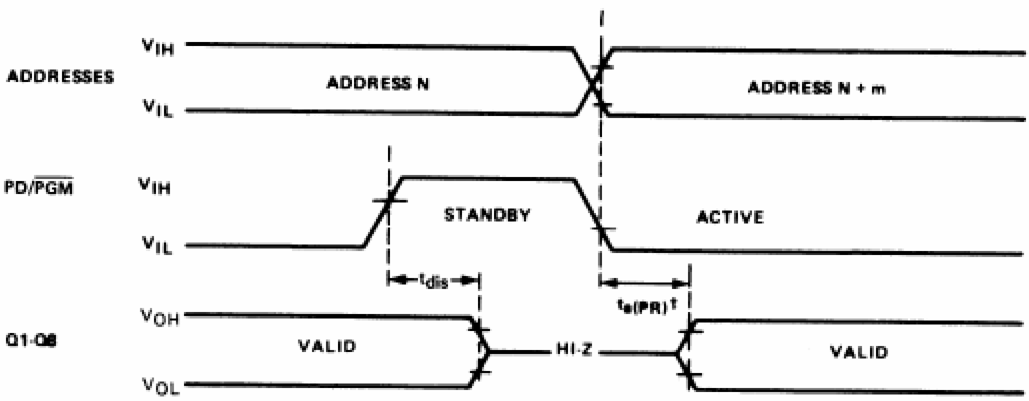
FIGURE 1 – TYPICAL OUTPUT LOAD CIRCUIT

read cycle timing



NOTE: There is no chip select pin on the TMS 2532.
The chip-select function is incorporated in the power-down mode.

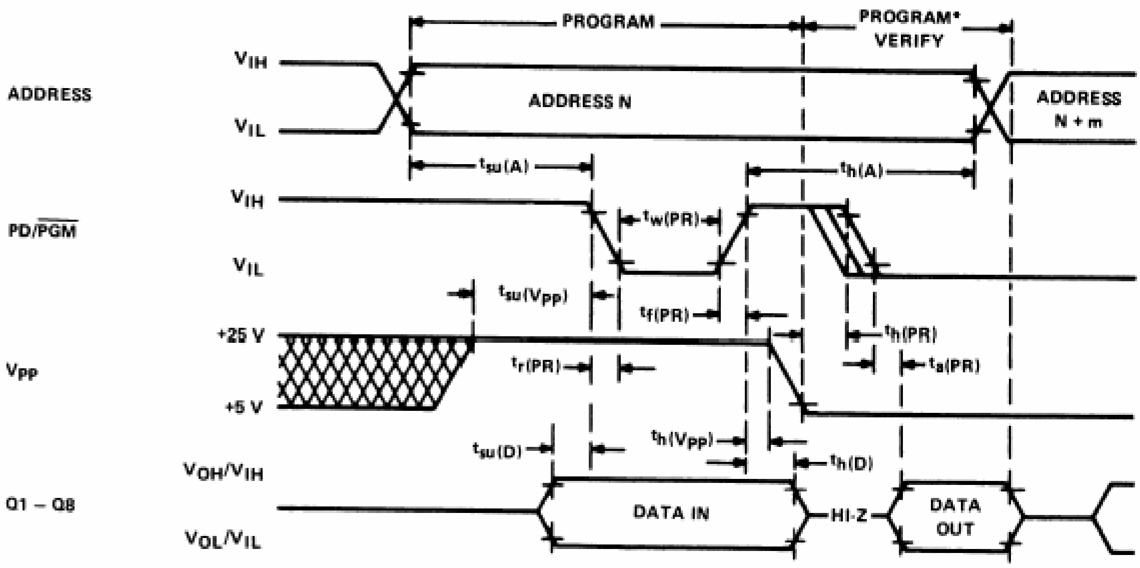
standby mode



† $t_{a(PR)}$ referenced to PD/$\overline{PGM}$ or the address, whichever occurs last.

All timing reference points in this data sheet (inputs and outputs) are 90% points.

program cycle timing



*Program verify equivalent to read mode.

MOS LSI

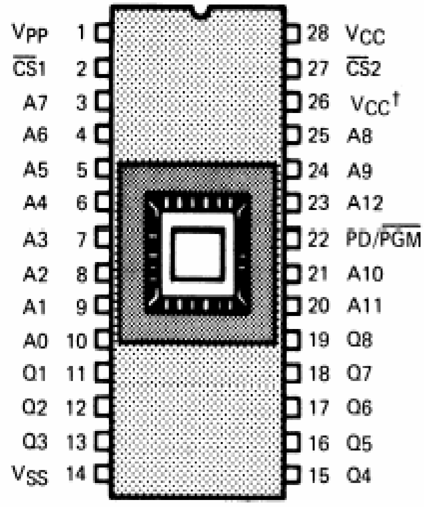
TMS 2564-45 JL
65,536-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

MAY 1981–REVISED MAY 1982

- Organization . . . 8K X 8
- Single +5 V Power Supply
- Pin Compatible with Existing ROMs and EPROMs (8K, 16K, 32K, and 64K)
- All Inputs/Outputs Fully TTL Compatible
- Static Operation (No Clocks, No Refresh)
- Max Access/Min Cycle Time . . . 450 ns
- 8-Bit Output for Use in Microprocessor-Based Systems
- N-Channel Silicon-Gate Technology
- 3-State Output Buffers
- Guaranteed DC Noise Immunity with Standard TTL Loads
- No Pull-Up Resistors Required
- Low Power Dissipation:

  Active . . . 400 mW Typical
  Standby . . . 75 mW Typical

TMS 2564
28-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



† V_{CC} may be connected to pin 26 for 24-pin ROM compatibility.

PIN NOMENCLATURE	
A(N)	Address inputs
CS(N)	Chip Selects
PD/PGM	Power Down/Program
Q(N)	Input/Output
V_{CC}	+5 V Power Supply
V_{PP}	+25 V Power Supply
V_{SS}	0 V Ground

description

The TMS 2564 is a 65,536-bit, ultraviolet-light-erasable, electrically programmable read-only memory. This device is fabricated using N-channel silicon-gate technology for high-speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can be driven by Series 74 TTL circuits without the use of external pull-up resistors, and each output can drive one Series 74 TTL circuit without external resistors. The data outputs are three-state for connecting multiple devices to a common bus. The TMS 2564 is offered in a dual-in-line ceramic package (JL or JDL suffix)* rated for operation from 0°C to 70°C.

Since this EPROM operates from a single +5 V supply (in the read mode), it is ideal for use in microprocessor systems. One other supply (+25 V) is needed for programming. Programming requires a single TTL level pulse per location. For programming outside of the system, existing EPROM programmers can be used. Locations may be programmed singly, in blocks, or at random.

The TMS 2564 is compatible with other 5-volt ROMs and EPROMs, including those in a 24-pin package.

operation

FUNCTION (PINS)	MODE							
	Read	Output Disable			Power Down	Start Programming	Inhibit Programming		
PD/$\overline{PGM}$ (22)	V_{IL}	V_{IH}	X	X	V_{IH}	Pulsed V_{IH} to V_{IL}	V_{IH}	X	X
$\overline{CS}$1 (21)	V_{IL}	X	V_{IH}	X	X	V_{IL}	X	V_{IH}	X
$\overline{CS}$2 (27)	V_{IL}	X	X	V_{IH}	X	V_{IL}	X	X	V_{IH}
V_{PP} (1)	+5 V	+5 V			+5 V	+25 V	+25 V		
V_{CC}* (26/28)	+5 V	+5 V			+5 V	+5 V	+5 V		
Q (11 to 13, 15 to 19)	Q	HI-Z			HI-Z	D	HI-Z		

X - Don't care.
* Do not use the internal jumper of 26-28 to conduct PC board currents.

read/output disable

When the outputs of two or more TMS 2564's are paralled on the same bus, the output of any particular device in the circuit can be read with no interference from the competing outputs of the other devices. To read the output of the TMS 2564, the low-level signal is applied to the PD/$\overline{PGM}$ and $\overline{CS}$ pins. All other devices in the circuit should have their outputs disabled by applying a high-level signal to one of these pins. Output data is accessed at pins Q1 to Q8. Data can be accessed in 450 ns = $t_{a(A)}$.

power down

Active power dissipation can be cut by over 80 percent by applying a high TTL signal to the PD/$\overline{PGM}$ pin. In this mode all outputs are in a high-impedance state.

erasure

Before programming, the TMS 2564 is erased by exposing the chip through the transparent lid to high intensity ultra-violet (wavelength 2537 angstroms). The recommended minimum exposure dose (= UV intensity X exposure time) is fifteen watt-seconds per square centimeter. A typical 12 milliwatt per square centimeter, filterless UV lamp will erase the device in about 21 minutes. The lamp should be located about 2.5 centimeters above the chip during erasure. After erasure, all bits are in the high state.

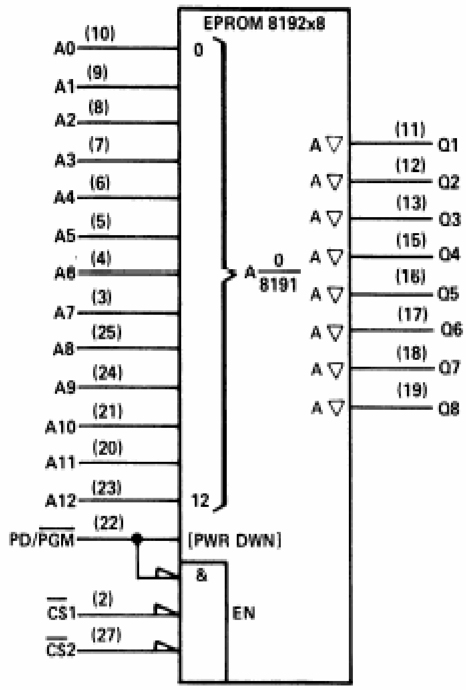
start programming

After erasure (all bits in logic high state), logic "0's" are programmed into the desired locations. A low can be erased only by ultraviolet light. The programming mode is achieved when Vpp is 25 V. Data is presented in parallel (8 bits) on pins Q1 to Q8. Once addresses and data are stable, a 50 millisecond low TTL pulse should be applied to the $\overline{PGM}$ pin at each address location to be programmed. Maximum pulse width is 55 milliseconds. Locations can be programmed in any order. More than one TMS 2564 can be programmed when the devices are connected in parallel. During programming both chip select signals should be held low unless program inhibit is desired.

inhibit programming

When two or more TMS 2564's are connected in parallel, data can be programmed into all devices or only chosen devices. TMS 2564's not intended to be programmed should have a high level applied to PD/$\overline{PGM}$ or $\overline{CS}$1 or $\overline{CS}$2.

logic symbol†



† This symbol is in accordance with IEEE Std 91/ANSI Y32.14 and recent decisions by IEEE and IEC. See explanation on page 289.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)*

Supply voltage, V_{CC} (see Note 1)	−0.3 to 6 V
Supply voltage, V_{PP} (see Note 1)	−0.3 to 28 V
All input voltages (see Note 1)	−0.3 to 6 V
Output voltage (operating with respect to V_{SS})	−0.3 to 6 V
Operating free-air temperature range	0°C to 70°C
Storage temperature range	−55°C to 125°C

NOTE 1: Under absolute maximum ratings, voltage values are with respect to the most-negative supply voltage, V_{SS} (substrate).

* Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

recommended operating conditions

PARAMETER	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC} (see Note 2)	4.75	5	5.25	V
Supply voltage, V_{PP} (see Note 3)		V_{CC}		V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH}	2		$V_{CC}+1$	V
Low-level input voltage, V_{IL}	−0.1†		0.8	V
Read cycle time, $t_{c(rd)}$	450			ns
Operating free-air temperature, T_A	0		70	°C

NOTES: 2. V_{CC} must be applied before or at the same time as V_{PP} and removed after or at the same time as V_{PP}. The device must not be inserted into or removed from the board when V_{PP} or V_{CC} is applied so that the device is not damaged.

3. V_{PP} can be connected to V_{CC} directly (except in the program mode). V_{CC} supply current in this case would be $I_{CC} + I_{PP}$. During programming, V_{PP} must be maintained at 25 V (± 1V).

† The algebraic convention, where the more negative limit is designated as minimum, is used in this data sheet for logic voltage levels and time intervals.

electrical characteristics over full ranges of recommended operating conditions

PARAMETER		TEST CONDITIONS		TMS 2564			UNIT
				MIN	TYP†	MAX	
V_{OH}	High-level output voltage*	I_{OH} = −400 μA		2.4			V
V_{OL}	Low-level output voltage*	I_{OL} = 2.1 mA				0.45	V
I_I	Input current (leakage)	V_I = 5.25 V				10	μA
I_O	Output current (leakage)	V_O = 5.25 V				10	μA
I_{PP1}	V_{PP} supply current	V_{PP} = 5.25 V	$PD/\overline{PGM} = V_{IL}$			18	mA
I_{PP2}	V_{PP} supply current (during program pulse)	$PD/\overline{PGM} = V_{IL}$				30	mA
I_{CC1}	V_{CC} supply current (standby)	$PD/\overline{PGM} = V_{IH}$			15	30	mA
I_{CC2}	V_{CC} supply current (active)	$PD/\overline{PGM} = V_{IL}$			80	160	mA

† Typical values are at T_A = 25°C and nominal voltages.
* AC and DC tests are made at 10% and 90% points using a 50% pattern.

capacitance over recommended supply voltage and operating free-air temperature range f = 1 MHz*

PARAMETER	TEST CONDITIONS	TYP†	MAX	UNIT
C_i Input capacitance	V_I = 0 V, f = 1 MHz	4	6	pF
C_o Output capacitance	V_O = 0 V, f = 1 MHz	8	12	pF

† All typical values are T_A = 25°C and nominal voltage.
* This parameter is tested on sample basis only.

TMS 2564-45 JL
65,536-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

switching characteristics over full ranges of recommended operating conditions (see note 4)

PARAMETER		TEST CONDITIONS (SEE NOTES 4 AND 5)	MIN	TYP†	MAX	UNIT
$t_{a(A)}$	Access time from address	C_L = 100 pF, 1 Series 74 TTL load, $t_r \leq 20$ ns, $t_f \leq 20$ ns, See Figure 1		280	450	ns
$t_{a(S)}$	Access time from $\overline{CS1}$ and $\overline{CS2}$ (whichever occurs last)				120	ns
$t_{a(PR)}$	Access time from PD/$\overline{PGM}$			280	450	ns
$t_{v(A)}$	Output data valid after address change		0			ns
$t_{dis(S)}$	Output disable time from chip select during read only (whichever occurs last) ‡		0		100	ns
$t_{dis(PR)}$	Output disable time from PD/$\overline{PGM}$ during standby ‡		0		100	ns

† All typical values are at T_A = 25°C and nominal voltages.

‡ Value calculated from 0.5 volt delta to measured output level.

recommended timing requirements for programming T_A = 25°C (see note 4)

PARAMETER		MIN	TYP†	MAX	UNIT
$t_{w(PR)}$	Pulse width, program pulse	45	50	55	ms
$t_{r(PR)}$	Rise time, program pulse	5			ns
$t_{f(PR)}$	Fall time, program pulse	5			ns
$t_{su(A)}$	Address setup time	2			µs
$t_{su(D)}$	Data setup time	2			µs
$t_{su(VPP)}$	Setup time from V_{PP}	0			ns
$t_{h(A)}$	Address hold time	2			µs
$t_{h(D)}$	Data hold time	2			µs
$t_{h(PR)}$	Program pulse hold time	0			ns
$t_{h(VPP)}$	V_{PP} hold time	0			ns

† Typical values are at nominal voltages.

NOTES: 4. For all switching characteristics and timing measurements, input pulse levels are 0.65 V to 2.2 V and V_{PP} = 25 V ± 1 V during programming. AC and DC timing measurements are made at 90% points using a 50% pattern.

5. Common test conditions apply for t_{dis} except during programming. For $t_{a(A)}$, $t_{a(S)}$, and t_{dis}, PD/$\overline{PGM}$ = V_{IL}.

PARAMETER MEASUREMENT INFORMATION

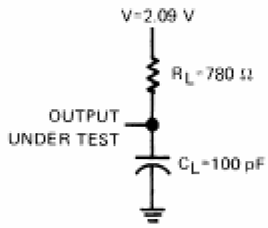
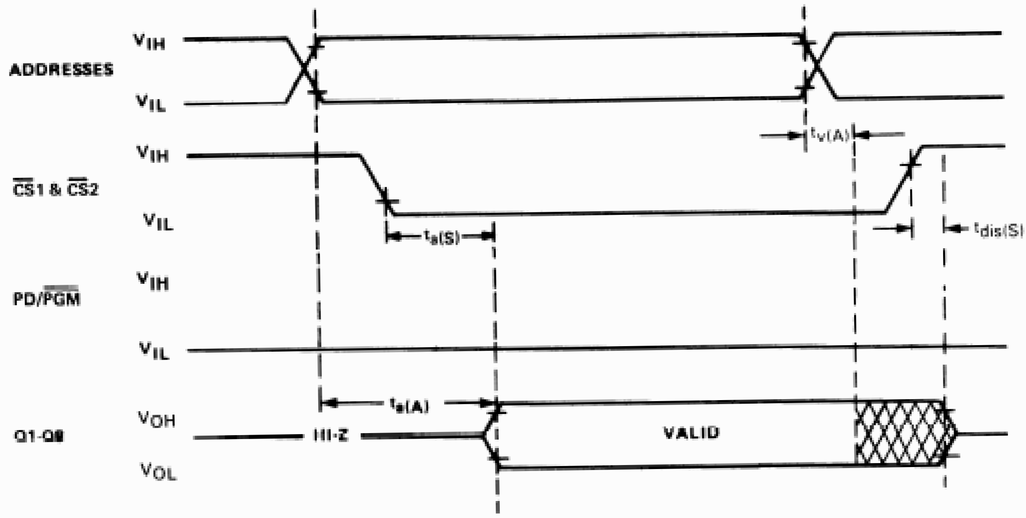
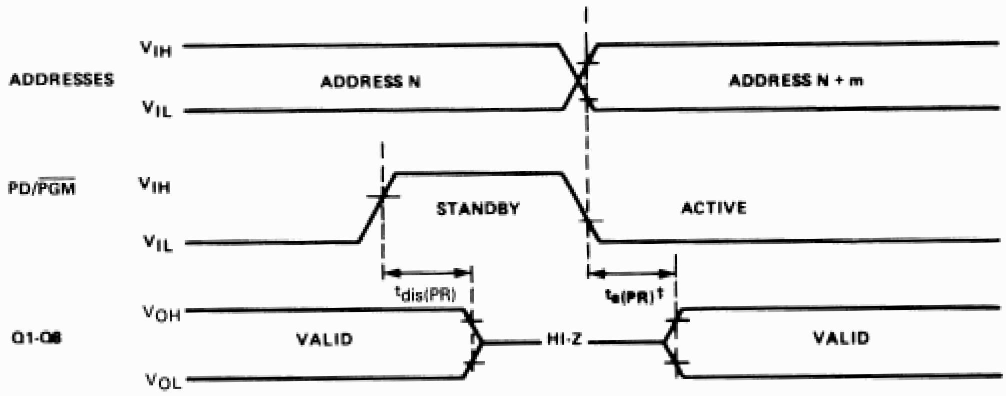


FIGURE 1 – TYPICAL OUTPUT LOAD CIRCUIT

read cycle timing



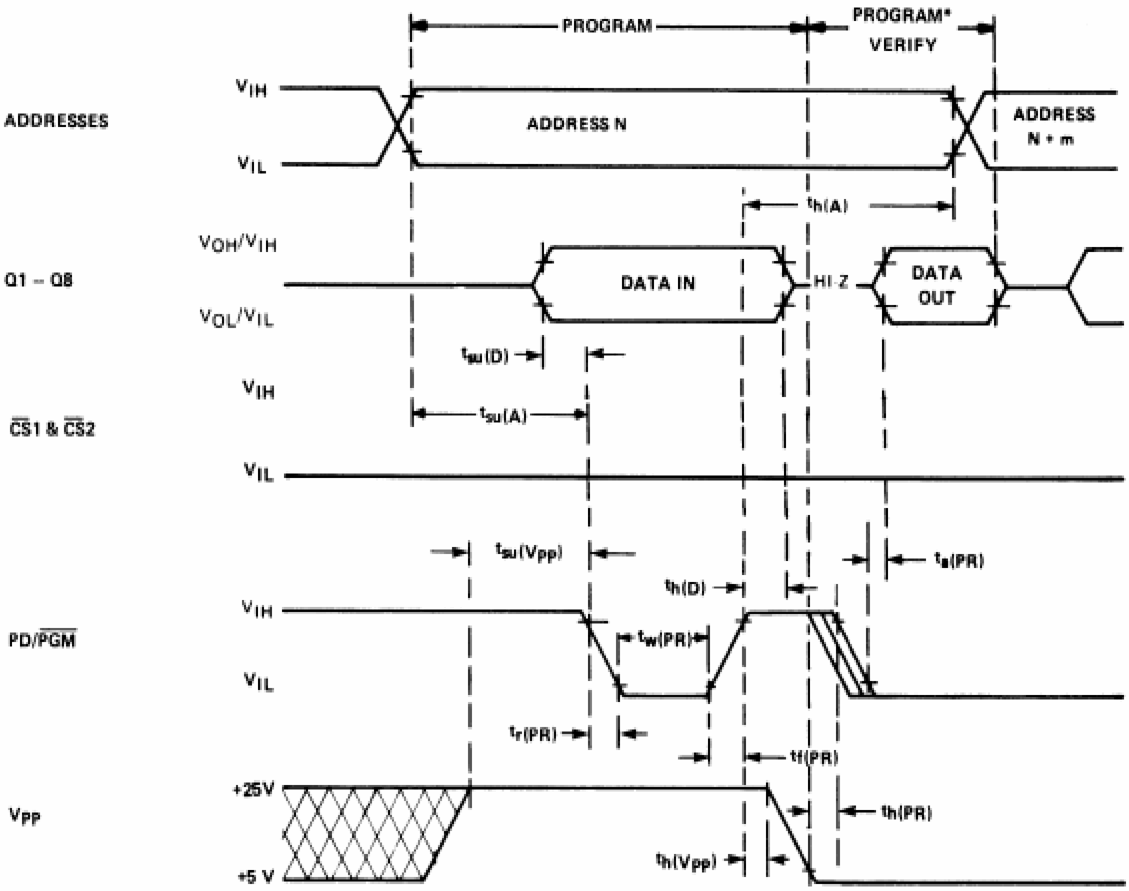
standby mode



† $t_{a(PR)}$ referenced to PD/$\overline{PGM}$ or the address, whichever occurs last.
$\overline{CS}$1 and $\overline{CS}$2 in Don't Care State in Standby Mode.

TMS 2564-45 JL
65,536-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY

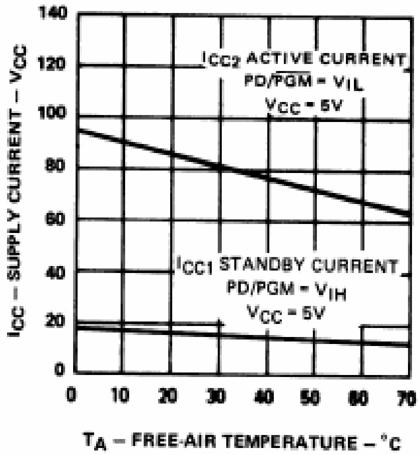
program cycle timing



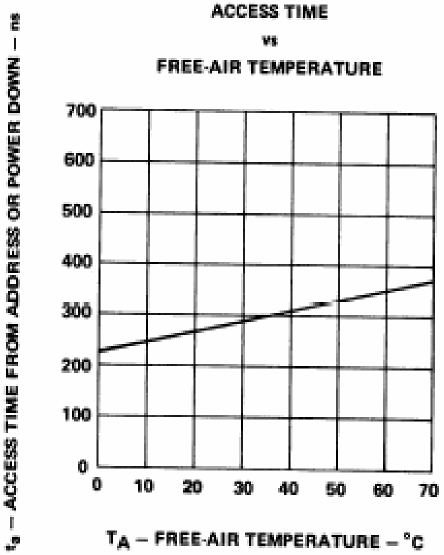
* Equivalent to read mode.

typical device characteristics (read mode)

SUPPLY CURRENT
vs
FREE-AIR TEMPERATURE



ACCESS TIME
vs
FREE-AIR TEMPERATURE



ASK MAV
ABOUT THE TRS-80 MODEL III MICROCOMPUTER
by Ian Mavric

1.
Q. I have a 16K cassette-based Model III, how do I upgrade it to 48K?
A. On the motherboard, up in the top right hand corner you will see 16 empty chip sockets. You need sixteen 4116 DRAM chips. The first row is already populated with the first 16K of memory, the 2nd row increases the memory to 32K, and the third row maximises the memory to 48K.

2.
Q. I hear there was an entry-level 4K Level 1 Model III - do you know much about it?
A. Yes it's very rare and did not sell many. It carried the catalog number 26-1061, and did indeed have 4K Ram and an improved version of Level 1 BASIC which had printing commands, since the Model III motherboard has a printer port on it as standard. Cassette baud rate was still 250 and the keyboard also lacked key-rollover. Contemporary collections seem to indicate that there are less than 50 left in existence. Just like 4K Level 1 Model Is are rare, it's entirely plausible that most were upgraded to 16K Model III BASIC or disk systems not long after purchase as most people found Level 1 BASIC far to restrictive to do anything serious.

Picture: early advert for the 4K Level 1 TRS-80 Model III

3.
Q. I have a 4K Level 1 Model III - how do I upgrade it to 16K and Model III BASIC?
A. Don't do it - you have a rare and collectable machine.

Ask around the internet forums or TRS-80 resellers for an already upgraded system.

4.
Q. Is there a High Resolution Graphics board for the Model III?
A. Yes, Tandy used to make one which carried the catalog number 26-1125 which provided 640 x 200 pixel resolution. The kit was expensive and so not many were sold. In Australia it was $529 and had not much software support, so it was a hard-sell. The board itself was a replacement for the rear RFI shield on US Model IIIs, or a RFI shield with a circuit board integrated into it which now fits over the motherboard. It connects to the 50-way I/O bus, runs of 5V, and has pass-through connectors for the monitor. There was very little software support from Tandy (Dow Jones Market Analyser and Business Graphics being two), though it did come with a very handy version of Basic with good graphics commands. Also 80micro featured programs from time to time which utilised the graphics board.

Photo: a high-resolution Model III

5.
Q. Tandy sold a Model III-to-4 upgrade kit but I never see them for sale on eBay?
A. This is an interesting one. Yes Tandy did sell a Model III-to-4 upgrade "kit". It was catalog number 26-1123 and comprised of a Model 4 motherboard, keyboard, nameplate, 64K

badge, sound beeper module, and TRSDOS 6 disk and manual. At over $1000 it was not popular but I do see upgraded systems appear on eBay in the USA from time to time and had a close look at one myself. I see that Tandy also replaces the aluminium chassis which supports the motherboard because the connectors for the keyboard, cassette and sound ports do not line up properly with the Model III aluminium chassis.

Photo: an upgraded Model III-to-4

Although Tandy called it an upgrade kit, it was not something which could be purchased over the counter. The hi-res board, by contrast, could be purchased over the counter even though Tandy recommended getting their technicians to fit it - it was pretty complex. The upgrade kit required you to trade-in your Model III parts and were only to be installed by technicians to make sure they retained the old parts. Hence why there was no over-the-counter upgrade kit purchasable, and why the kits themselves never show up on the marketplace.

ianm@trs-80.com

LAURIE SHIELDS CONTACTED ME TO LET ME KNOW HE'D FOUND A LATER VERSION OF ZEN ON THE INTERNET. ZEN85 V6.2, AS YOUR NO DOUBT AWARE, IS AVAILABLE FROM THE DOWNLOADS PAGE. PART OF THE .DSK FILE CONTAINS AN 'UPDATE' FILE WHICH, I BELIEVE, IS IN SCRIPSIT FORMAT. JUST IN CASE YOU'VE A PROBLEM WITH THAT, I'VE CONVERTED IT TO .DOC FOR THE NEWSLETTER.

Z E N 8 5 Ver. 6.2

Introduction

It is so long now since I started work on Zen85 that I've lost track of some of the minor alterations, in most cases simply making default entries to prompts meaningful. Hopefully the more important changes are all documented below and I suppose will eventually be incorporated into a new complete manual.

One of the big demands on time was the creation of the CP/M version that is now available making that operating system just about bearable. Should you discover anything not performing as expected please let me know, probably the fault will be in the documentation rather than the software.

The biggest change since Zen84 is the inclusion of a two pass labelling disassembler, at present only disassembling from memory, and unfortunately taking up 2K of Ram. I hope in the near future to put the assembler and disassembler into separate overlays but for the time being the end of Zen is way beyond the start of text in Scripsit and the in-memory transfer from Scripsit to Zen is no longer available.

The other significant alterations are in response to requests from a number of users some of which are given in no particular order:

a. Charley Butler of TAS, who requested the relaxation of the command syntax so that spaces are now allowed between command letter and any numeric parameters.
b. David Jones, assembling 150K of source at a time for Spectrum games, who wanted things a bit faster so Zen now can assemble from disc in whole files rather than a line at a time.
c. Leo Knaggs, who always outputs listing files to disc with the 'F' option and he wanted the facility to change the

output to different listing files during assembly. He then suggested the CHAIN pseudo-op to link files one to another and following those two I decided that we may as well have the facility to GET files from within another file, so that went in.
d. Thanks to Peter Knaggs who over the last year has come up with many suggestions and thought provoking ideas. A chance remark of his, pointed the way to save a few hundred bytes with no sacrifice on performance.
e. Thanks also to Geoff Smith who bullied me into doing something about the pedestrian way in which the old Zen 'Z'apped lines of text, and whilst on the subject Geoff's specialised Scripsit out-performs every other word processor I've seen.
f. Robert Madge of Intelligent Software who insisted that the Enterprise Zen should be equally happy with upper or lower case text. Having done it on that machine it looked so good that it had to go into the TRS80 version.

This latter point raised another complication of postfixes for numerical quantities when in the hexadecimal default mode. In other words as the letters "D" and "B" are legitimate hex characters we cannot use them to indicate decimal and binary numbers. To overcome this I have adopted "T" for base Ten and "I" for bInary. If anyone has a better suggestion let me know, or alternatively you can roll your own in the region of 6BE8.

Summary of extra commands and facilities

Editor Commands

RX Read Unknown — Reads any format disk file into End+1 upwards. Size of file remembered.
WX Write Unknown — Writes unknown format file to disk.

These two commands permit editing of any format disk file provided it can be fitted into available memory.

g — Global move of text (old command letter 'c').
DU — Duplicate source code. Prompts for Start, End and Destination. Destination line number must be at least two lines before/beyond block start/end respectively.

Note: DU and g assume default numeric base 10 for the input parameters.
R Read from disk now automatically recognises un-numbered Edas files. To speed up the disc read process, Zen now only checks for the availability of 256 bytes before each sector read, rather than, as before, on the availability of room for each byte as they were read from the buffer. This has

speeded up the read enormously but at the expense of not being able to use some 200 bytes at the top of memory.

```
RES       Resets the source file pointers to their initial
          values. Note any source file in memory is ignored
          and thereby killed.
```

2. Assembler Commands

The following have been added to the assembler:

```
BASE      sets the assembler default number base to parameter
          value, equivalent to RADIX on some American
          assemblers.
DEFW & DW now take multiple parameters.
SHORT     limits the listing of object code expansion to four
          bytes, this is the default state for the assembler.
LONG      allows full listing of object code, this was the old
          default state before version 6.0.
IFNOT     new conditional pseudo-op.
IFNOTEQ   new conditional pseudo-op.
DEFD & DD new Define Data pseudo-op, takes two parameters, eg:
          DEFD n,d   where n = number of data bytes and, d =
          value in range 0 to 255.
SLS       for the missing undocumented shift (Shift Left Set).
PAGE      simply forces a new page.
```

File handling pseudo-ops, each of these takes a "Filespec" parameter:

```
FILE      Closes the current listing file and outputs to
          "Filespec" Pseudo-op is ignored if not outputting
          with the 'F' option.
CHAIN     Closes the current source file, any remaining text
          being ignored and inputs from "Filespec". Pseudo-op
          is ignored if not assembling from disc.
GET       Closes the current source file but remembers
          filename and line number.  Input then taken from
          "Filespec" until eof, whence the original file is
          restored at the next line after the GET.  Only one
          level of Getfile is supported.  As with the previous
          pseudo-op this one is ignored if not assembling
          from disc. The reserved words for use as labels are
          now restricted to just register names, mathematical
          operators and condition codes.  This permits the use
          of labels such as END: and LDIR: etc. A new maths
          operator has been added # for xor, eg:  ld
          a,value#20h Obviously # is now included in the
          reserved list and so the indicator for local labels
          in Mac/Bas has had to be altered and is now the %
          character.
```

Assembler modes of operation

An extra mode of operation, namely of assembling whole files direct from disc, rather than files in a line by line process, has been added. This "lumps" rather than "lines" mode is signalled by an asterisk "*" as the first character of input to the 'Source>' prompt, examples:

```
Source>{enter}      As before, assemble in memory file.
Source>SCRIP{enter) As before, assemble SCRIP from disc by
lines.
Source>*SCRIP{enter}    Assemble SCRIP from disc as a whole
                    file, in effect no different from the
                    first example unless Scrip contains Get
                    statement(s) or finishes with a CHAIN
                    statement.
Source>=SCRIP{enter}    As before, where SCRIP is a linking
file.
Source>*=SCRIP{enter}   As above where SCRIP is a linking
                    file, but assembly is to be in lumps
                    rather than lines. In any of these
                    assembly modes, should Zen's symbol table
                    become full then Zen automatically makes a
                    1000 byte increase and starts at the
                    beginning again.  To indicate this action
                    Zen displays a "*" on the video
                    irrespective of output mode. The cross
                    reference listing from REF is now sorted
                    in the same fashion as the Sorted Symbol
                    Table.
```

3. Extended Debugger

A single stepping simulator has been added to the debugger. This operates by copying the user-instruction into an internal buffer and trapping the machine state after the instruction has been executed. Thus single stepping of both Rom and Ram with full control at the keyboard is possible. Where subroutines are required to be executed without delay their addresses can be preset. This extended debugger handles all of the legal Z80 op-codes and most of the illegal codes including the half register IX and IY ones. Whilst the simulator is operating the user-register state is displayed after each instruction together with the disassembly of the next instruction at the PC address. Optionally a 'window' on any 64 byte block of memory can also be shown in both hex and ASCII.

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

Commands:

c   Call address. This command accepts the following parameters:

0   clears all stored addresses.

1 to FFFFh stored as call address for execution in real time. Default display all stored addresses.

DE   Toggles the extended debugger on/off. When enabled the debugger defaults to hexadecimal number base and the Z> prompt is changed to Z?w. Without any parameter cancels the window mode, otherwise Parameter is stored as address of 64 byte window display block.

R   Run simulator. If the extended debugger is enabled via the 'DE'   command then 'R' will start the simulator in the 'Run' mode with the keyboard scanned after every user-instruction. Simulation starts at the supplied parameter address or if none supplied then at the default user PC address. The space-bar freezes the action and the 'X' key returns to command mode.

S   Single Step. Similar to above but after each user-instruction is executed action is frozen awaiting a key-press. Any other key than 'X' continues the simulation otherwise control returns to Zen's command mode.

Note: Whilst the simulator is active any Calls in any user program to 002B will be pre-empted by Zen's own keyboard scan even if 002B is set to be called using 'c002BH'. To overcome this difficulty use 0049H instead and set that address to be called with 'c0049H'. Zen monitors the user stack during simulation and will force an abort if it goes above its initial value. This allows subroutines to be stepped without the need to set a breakpoint.

Try the following command sequence:

Z>DE{enter}10Z?R33{enter}Whatever was in the A register should eventually be displayed on the video and control return on completion.

4. Disassembler

As mentioned above, Zen is now equipped with a comprehensive two pass labelling disassembler capable of handling code moved in memory from its executing address. The disassembler can be used in two modes, the first is a clean quick unscrambling of the next eight instructions to the video and the second is a more comprehensive disassembly of a defined block of memory with up to 64 separately identified data areas. The data areas produce simple DB statements. The output from the disassembler can be either to Zen's own source file buffer which naturally grows during the process, or to any of the listing devices. The command syntax is as follows:

u   Unscramble. Either from the byte following the last unscramble or from the parameter address. Output is to the video only.

DA   Disassemble. Prompts are given for start, stop and execution addresses. The latter address is relevant should the code to be disassembled be in a different place in memory from that which it normally executes. Zen assumes the start address for the execution one if a default entry is made. Likewise should defaults entries be made for the first two values then these are left unchanged from the previous disassembly. After settling these values Zen prompts for the start and stop addresses for any data areas. Defaulting on entry terminates the definition of these areas and Zen requires the output option. Output options "V", "E" and "F" are as described for assembler output. The default option directs the source file to Zen's own text buffer. During the first pass all referred addresses are stored in Zen's symbol table area that could run out of room and generate a "Full" error message.

Note. All numeric input to the assembler prompts is treated by Zen as being hexadecimal, there is no need to postfix with "H". If you intend decimal input then use the "T" postfix for base Ten.

5. Miscellaneous

I think all of the foregoing and the rest applies equally well to Zen4/Cmd 3.0 and the CP/M version of Zen. I've added default file extensions to Zen for most of the file orientated procedures. Zen applies these default extensions without regard to the possibility of correctly named files already existing without extensions. If you don't like this situation then simply replace the bytes from 569C to 56AE with 0D's. Assuming you don't then the extensions used are as follows:

/ZEN   Applies to all source file Write, Read, RI and RL operations.

/LNK   Applies to linking files.

/LST   Applies to listing files created with the "F" option.

/CMD   Applies to output "D" option, and RD and WD. Zen does not apply default extensions to other disk I/O such as RX, WX and kill. One small point to mention, always, when using Re-entry for debugging, do it with CALL REENTRY and also check the actual address with the 'r' command in Zen. Printer status is now stored as Bit 7 of 5534. During the page display, now modified to displaying a grahics line, the 'Y' key is recognised to toggle the page display on/off. The unallocated space in the last sector after the end-of-file in all disk writes is now filled with zeros. This permits Zen to be used for editing Basic programs, JCL files etc that do not contain their own internal end-of-file marker. We have not been supplied with any details regarding accessing the directory under Model 1 Dosplus. Consequently this customising defaults to the TRS 2.3 mode. This will generate a directory in single density if on track 17 but is unlikely to work in other configurations.

Address of pointers in Zen

	Zen85 v 6.0		Zen4 v 3.1
SOFP	86FB		62FB
EOFP	86FD		62FD
PSTAT	5534	(Bit 7)	3335
REENTRY	5705		352D
LIMIT	5634		3423
STACK	552C		332C
IMAGE	56E9		3511
MBASE	5629		3418
PAGENO	562C		341B
TBUFF	5400		3200

6. General notes

Zen only uses the Rom single character scan at 2BH for all keyboard input. With Newdos80 and Dosplus this routine is treated as a valid device for 'Chained' input and consequently Zen can be operated via chain-files with both these systems.

Newdos80 actually allows the user to decide whether the single key routine is 'chain-able' as well as the line-input routine. Unfortunately the authors of LDOS and TrsDos 6.x have decided, in their wisdom, that their chain file system cannot support the single character input device and therefore whilst chain files can be used to simply activate Zen they cannot provide any input. Since it is by the use of these single key commands that Zen provides the desirable facilities for scrolling through and editing the text it is unlikely that the input routines in Zen will be re-written simply to support these illogical implementations of chain files.
For Newdos80 users there is the facility (/./5Y) in their chain files to drop one level of Doscall whilst within a chaining. This means that using the 'DO' command in Zen will permit a chain file to be activated at any time so as to provide a standard sequence of keyboard input. In its simplest form imagine creating a program called Scripsit/Cmd

where the source files are in three parts, Scrip1/Zen, Sctip2/Zen and Scrip3/Zen. After modifying any or all of these files then a standard sequence of commands would be used such as (spaces added after prompts for clarity):

```
Z> A<cr>
    Source file>   SCRIP1/ZEN, SCRIP2/ZEN, SCRIP3/ZEN<cr>
    Option>        D<cr>
    Filespec>      SCRIPSIT/CMD<cr>

Z>  O<cr>
    Dos ready      SCRIPSIT/CMD<cr>
```

Whilst the entering of the list of source files can be replaced by using a linking file the rest still has to be entered every time unless a chain file can be triggered to provide this input. This can be done under Newdos80 by using Zen's 'DO' command to carry out a Dos command to start chaining a file as follows:

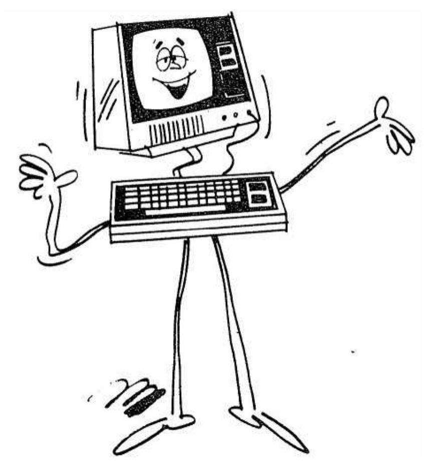
DO DO SCRIP

or

Z>DO CHAIN SCRIP where the file SCRIP/JCL, written with Zen of course, is:

/./5Y A SCRIP1/ZEN,SCRIP2/ZEN,SCRIP3/ZEN D SCRIPSIT/CMD O SCRIPSIT

(Don Bannister uses this technique and calls the chainfile DO/JCL, thus from Zen he keys DO DO DO, known as the Sinatra option). A similar technique can be used in Newdos80 Basic to activate a chain file to replace keyboard input in mid-program via CMD"DO CHAINFIL" or even CMD"DO "+X$. Thus allowing selective files to be activated by the logic determining the value of X$



HAPPINESS IS A NOT
TOO-WARM M1 TANDY!

IN MAV'S WORKSHOP

by

IAN MAVRIC

PROJECT 1: MOUNTING FreHD IN A MODEL I DISK DRIVE CASE

A few weeks ago I found on eBay a Dragon dual disk drive unit. Dragon was a UK company which made a computer compatible with the TRS-80 Colour Computer called the Dragon 32 and Dragon 64. Like all well supported computers there was a dual disk drive unit available for the Dragon 32/64 and this is what I had found. Luckily it was only $20 and just as well... both disk drives were damaged beyond repair. What I had bought was a $20 case and power supply. Although the original plan was to buy the Dragon dual drive to use with a Model 1, it occurred to me that by installing a FreHD inside it, and a single disk drive, one could considerably cut down the number of power-outlets needed to run a Model 1 system.

Photo: Dragon Dual Disk Drive for Dragon 32 or 64 Computer - it didn't work.

In its standard form the Model 1, by virtue of its modular design, can take an incredibly large number of power outlets just to make it work... One for the monitor, one each for the computer and Expansion Interface, one for each disk drive. Then if you expand the system further with a speech synthesiser, printer, modem and hard drive, one extra power outlet for each device. You soon end up with daisy-chained power-strips.

Hence my interest in dual disk drive boxes and any idea which will cut down the number of power strips needed to get systems up and running. Putting a FreHD inside a disk drive box and powered by the same power supply seemed like a great idea.

Not long ago I made some Model 1 hard disk adaptor boards which allowed connection of a TRS-80 hard disk drive or a FreHD to a Model 1. These little devices have two resistors and two transistors, and is connected to the Expansion Interface via a short cable. With only four parts it is powered by the Expansion Interface. There is a lot of electronic "traffic" on the short connecting cable, but I still planned to mount both the FreHD and M1 HD adaptor into the disk drive case for a neater installation. How long could I make the "short" cable before it would run into problems of signal degradation and propagation irregularities causing timing dramas?

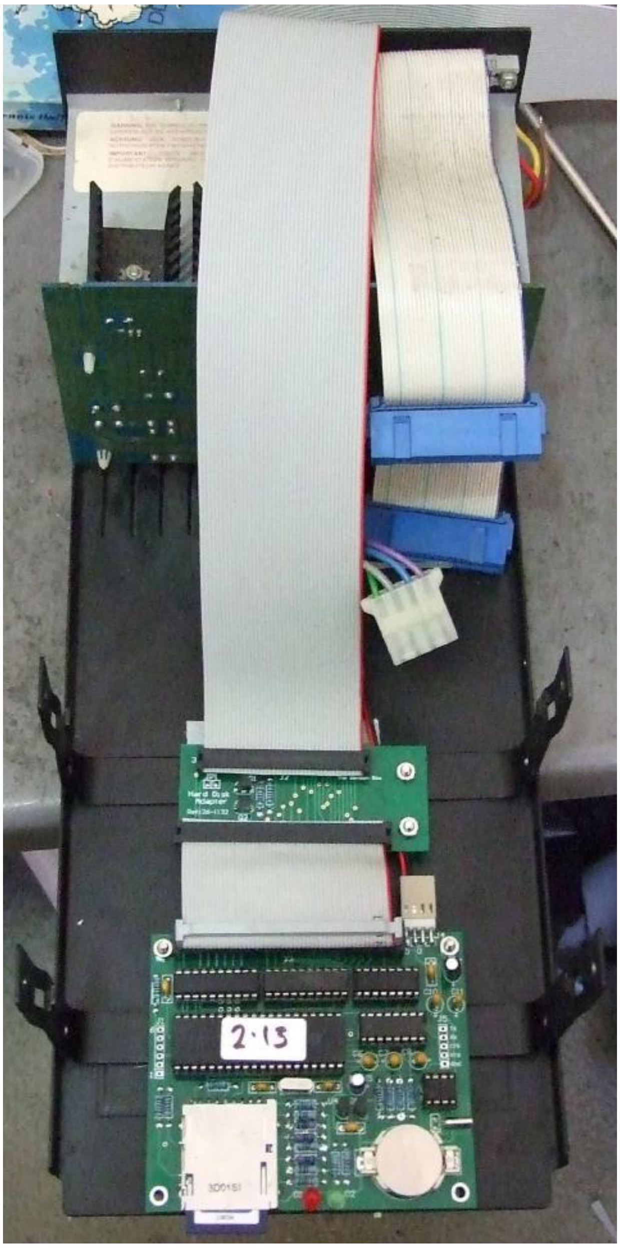


Photo: FreHD and M1 HD adaptor mounted into the Dragon drive case

I tried with a 90cm long cable and it worked a treat. I put an AM radio near the FreHD and found there was considerable electronic noise, but in the tests I ran I found no problems. I didn't feel like testing with any longer cable since 90cm is probably more than anyone would want to conceivably use.

After some time I decided upon a disk drive to mount in the Dragon case, a 40 track double sided double density SA-455 from Shugart Associates. While dual drives are nice on a Model 1, let's face it if you work off a FreHD or hard drive, you only need one disk drive to boot the system, and for a method of loading programs from diskette onto your FreHD. One drive would be adequate.

As you can see from the photos I mounted the FreHD up front in the case and the M1 HD adaptor right behind it joined by a short 5cm long cable. I then made up a small power cable to bring 5V from the disk drive power supply to the FreHD. The data cables for the FreHD and the disk drive come out at the same place, and were shortened to about 60cm.

Photo: FreHD mounted and the disk drive allows the system to boot

As a side project for this one I also made a modification to the Australian TRS-80 power supply brick. You see in most countries the power supply for the Model 1 is a small black unit which provides enough current to power either the Model 1 keyboard unit or the Expansion Interface, but not both. In Australia though, Tandy Electronics contracted Furguson Transformers to make the 240V power supply for the Model 1. It's twice the size of the small black ones used in the USA and UK, and fortuitously due to its over-engineered design has plenty of current to drive both the Model 1 and Expansion

Interface. Al you need to do is open it up and install a second power lead to the Expansion Interface. I've done this several times over the years and always got a good result. Thanks Furguson - the supply may be too big to fit in the EI but otherwise is a great design.

So my nicely restored TRS-80 Model 1 system now has a green-screen monitor with anti-glare panel, a M1 and EI powered from one power supply, and a disk drive box with both a disk drive and FreHD in it. With 48K Ram, lowercase modification, ALPS keyboard, and megabytes of storage online thanks to FreHD, the system only takes up 3 power outlets, and is both neat and reliable, and isn't that what everyone wants from their Model 1 systems?

Photo: Neat little FreHD-based TRS-80 Model 1 system

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.

Business Time with Kev
Kevin Parker

Hi everyone!!! Let me say that it’s a great opportunity that has come my way to write for Dusty’s TRS8BIT mag.
So the first order of business is probably a little bit about me and how I came to be writing this column. Kevin Parker is my name and like another columnist, Ian Mavric, I live in Victoria (Australia) but I’m not a native of Victoria. I actually grew up in South Australia and moved to Victoria about eight or nine years ago. I live in a little country town near the regional coastal city of Warrnambool in the south west of Victoria. I’m not that far from some of the greatest coastal scenery in the world such as The Great Ocean Road and The Twelve Apostles for those who have toured down under. Despite that I still have my day job in Melbourne (the capital city of Victoria and about 250kms away for our non-Australian readers) as the Senior Web Analyst for an insurance company. I’ll talk a little bit about work-life too in this inaugural article because my work history has some bearing on where I’m at today as a classic computer collector.

So the story goes, a little while ago I was approached to write a column about the 8″ TRS-80’s of which I have two but I’ll save the machine details for later articles. I actually have three 8″ machines in my classic computer collection but I’ll save the detail of #3 for later as well.

Now I actually only know two things about the 8″ machines - they have at least one 8″ floppy drive and they're awfully heavy. If you’ve ever tried to pick one up you’ll feel the quality! So why am I writing this column then? When I was initially approached I said no but then it occurred to me that I’m actually going to try and get these machines up and running and restored to their former glory so that journey in itself is probably of interest to others. You might say that this column will almost write itself as we go along.
So let me take you back (briefly) about 30+ years!

My tertiary qualifications are actually in Human Resource Management and Business Management but interestingly I actually found myself working in business facing IT and then web related positions. It’s surprising (or maybe not to some) how many people work in areas that are not necessarily related to their tertiary education but that’s how things work out some times - for me that outcome has not been a bad thing at all.

A previous employer (several employers ago now) decided that to facilitate the entry of computers into the workplace (that's how far back I go) they would buy a few laptops and lend them to staff to generate familiarity. 40 lines in monochrome LCD brings back memories - it might have been something like a Toshiba T1100 but the memory around that detail is now quite vague.

Now, more out of curiosity at the time, I actually already had my first computer, a TRS-80 Model 1 with an EI and two floppy disk drives (which in hindsight knowing what I know today I should not have sold of course), which I was dabbling with but dabbling with the laptop got me truly hooked. That produced the segue into more IT related employment positions primarily working with IT as the business representative performing roles more like a Business Analyst. Then, to cut a very long story short, I found my way, through some natural progression and some accidents of tenure, into the web area eventually becoming the Web Services Manager for a statutory corporation in South Australia.

I also run a small web hosting and development business on the side so you can see that my life for many years has been comprised of computer technology in various forms and ages almost 24x7.

Despite the changes in fortune in my working life and accelerating technological advances one thing has never changed - my interest in old computer technology. Even with taking the logical path forward for most computer users to using PC's, I've retained a very keen interest in classic computer technology - someone has to preserve the history is what I often say to those curious about my hobby.

My interests are more eclectic and I have quite a range of classic computers in my collection but my two key areas of interest are TRS-80's and black, business to enterprise grade IBMs (my current employer finds it most curious that I own more AS/400's than they do). From time to time I might mention other machines I have or use (or even don't use) in this column because it's all part of the history and the story and we know for certain that while they did a dam good job given the times, Radio Shack did not have a monopoly on good ideas, solutions or what makes classic computers interesting.

For many years I was also a member of the South Australian based AMUG (Adelaide Micro Users Group <http://www.trs-80.com/wordpress/magazine-user-group/>) which was a club of like-minded users back in those days using both TRS-80 and System-80 machines (but interestingly I can't recall anyone in the club actually having a System-80). I can't recall how many

years I was a member but I might try to work it out one day. I think my name may have even appeared in an issue of the club Newsletter when I wrote a very open, but constructive, article about some aspects of how the club was being run or not being run depending on your outlook at the time.
I also have very fond memories of accessing the Tan80 Connection BBS during those years which was run by Eric Rasmussen, a member of AMUG, and if memory serves me correct he was also the President for some time and a very good President I seem to recall or at least a very good member.
In those times connecting to another computer got the very big WOW factor but who would have thought then that connectivity would become what it is today.
For the curious I found this listing from 1994

System	TAN80 BBS
Sysop	Erik Rasmussen
Phone	(08) 326-1132 Multi-Line
FidoNet	3:800/822
Baud	V.21 V.22 V.22bis V.23 V.32 V.32bis
Access	Mem LVA
Hours	24 Hours
Computer	IBM 386/33 Clone
DOS	MS-DOS
BBSoftware	Opus
Established	Jun 1989 (4 Years 9 Months)
Info	Supporting IBM-PC, Soundblaster
Tandy	TRS-80 Models 1, 3 and 4
	CP/M, CD-ROM online.

Contact point for Adelaide Micro Users Group.

(Source: http://slash.dotat.org/~newton/bbslist.txt)

I can say that in those days I had absolutely no awareness that 8″ machines even existed.

So where did the column name came from? Well the 8″ TRS-80’s primarily had a place in business back in those days. They were probably over-engineered for home use and were awfully expensive for those days. Their feature set was more focused on business solutions so it seemed pertinent that my column title should reflect that. Other names were canvassed but not knowing the TRS8BIT audience too well at this stage, references to eight inches and columns may not have been well received!

Now despite being a little light hearted about it, I have neither overstated nor understated the extent of my knowledge about the 8″ TRS-80 machines - I came into possession of these machines more by accident (the story about the rotary hoe will make an interesting paragraph or two a bit later) and good fortune rather than a desire to actually add them to my collection (but I'm most certainly not unhappy about that) so anything I say in this column you can definitely take as interesting but please don't take it as gospel. This is going to be more about the journey in getting these machines up and running and other aspects of my collecting life that kind of fit into and around that. If you see anything that I've got wrong or misquoted or that needs correcting or enhancement your feedback is very welcome - it might make good fodder for the next article.

Kevin

PLAY THE DOCTOR WHO ADVENTURE ON YOUR TRS-80 MODEL I / III BY IAN MAVRIC

With 2013 being the 50th anniversary of Dr Who I thought I'd write a little belatedly about an adventure game based on the show which appeared in Micro-80 magazine Volume 3, Issue 8, July 1982, pages 21,31-34. It's a cassette or disk based game which runs on a 16K cassette system or a 32K disk system. Its a long and complex program, in 2 parts (an initialiser and the main program) which I'd always meant to type in out of the magazine but it always seemed daunting.

Fortuitously, Ira from www.trs-80.com was contacted by the author of the program, Jim Smith, who gave permission to place the game in the public domain and be made available for download from Ira's site. You can find the download at: http://www.trs-80.com/cgi-bin/down-ok-model1.pl?drwhoadv.zip

I emailed Australian Dr Who aficionado Aaron Brockbank about how the game had been turned over to public domain, and he immediately recognised the game from his time at Newcastle University, where the original of this game ran on an old PDP-11/70 and was written in DEC-BASIC. The similarities of DEC-BASIC to TRS-80 Model II Basic enabled Aaron to "port" the original game to the TRS-80 Model II in approximately 1981. I'm trying to organise a program listing from Aaron for posterity.

At around the same time I emailed Aaron I obtained a contact email address for Jim Smith and sent some questions about his version of the Dr Who Adventure, which I'm pleased to present the answers here:

Ian: May I send you some questions to answer about the program?

Jim: I wrote the game under Newdos/80, as a single program. To get it published I had to make it fit into 16K, which is why there is a data file.

The published version was written for the tape system. There were instructions for modifying it for a DOS, which amounted to replacing two lines (720 and 730) with:

```
720 OPEN"I",1,"DRWHO/DAT
730 INPUT#1,SP:INPUT#1,SD:INPUT#1,SO:INPUT#1,S1
:S1=LEFT$(S1,60)+CHR$(34)+MID$(S1,62,4)+CHR$(34)
+RIGHT$(S1,12):CLOSE1
```

which reads from a disk file rather than tape. A better solution would be to merge the two programs back into one, but I don't have the means to do that. :-)

I'd also recommend changing TL=20+RND(20) on line 760 to something like TL=200+RND(20). :-) Otherwise it's ridiculously hard.

Ian: What is the objective of the game?

Jim: To collect the six parts of the key of time and bring them back to the throne room on Gallifrey (which I misspelled as Galafry).

Ian: How did you get started on the project?

Jim: Someone at Newcastle University (I can't tell you who) had written a Dr Who game which was floating around on their PDP 11/70 computer and which I played, and that inspired me to write the program.

I nicked a couple of their ideas, but all the code and most of the story are my invention.

Ian: What programming difficulties / limitations did you two find?

Jim: I wrote it on a 48K disk system. The magazine wanted a 16K cassette version, and it was too big. So I had to compress it down. That's why there's a data file. :-) The code itself is full of byte saving tricks.
One element I was proud of at the time was the command decoding system. My original version was far to slow, so I wrote a new version using a binary search algorithm, and it flew. I was the cat who got the cream!

IAN: THANKS JIM.

To read the original article and see the program listing as published, you can download the Micro-80 issue in question from Terry Stewart's System 80 site: http://www.classic-computers.org.nz/system-80/micro80/Micro-80-Volume-03-Issue-08-(1982)opt.pdf

CREATING IMAGE AND DSK FILES
GAZZA

Transferring files to and from the TRS-80 line of computers and a Windows PC is reasonably easy using the RS232 and if you own a FreHD, then the supplied utilities - IMPORT2, EXPORT2 and DSK/DCT make life even easier. While DSK/DCT and DISKCOPY/CMD will allow you to make working copies of most DSK files it will not allow you to make copies of disks such as Percom's Microdos (single density) or Newdos80/86/90.
What to do if you don't own a FreHD or if you want to email someone a copy of the above mentioned disks or even make copies of your existing floppies so you can run them on an emulator.
Most methods involve using an old MSDOS based PC with a 5.25" drive that can handle single density or some piece of wiz-bang hardware, but most of us already have the tools to do the job. Kim Watt's masterpiece, Super Utility, Matthew Reed's TRS32, TRSTOOLS, an RS232 or a FreHD and a couple of blank floppies are all we need.

As it's impossible to cover all the possible combinations of software, operating systems and hardware, in the following text I'll just describe the method I used to make both IMAGE and DSK files from copies of my Newdos/80/86 hard drive boot disks using a TRS80 4P.

I've also used this method for making image files and working floppies from DSK files of various disks including Multidos Models 1 and 3, Newdos80/86/90, Hardos, Dosplus Models 1,3 and 4 and Rapidos and I have also used a System80 to make image and DSK files of various Dick Smith's programmes such as WORP1, WORP9 etc that ran on the Microdos/OS80 operating system.

The reader will need to alter the details to suit their own situation. This process is definitely not PnP and the reader will need to have a working knowledge of the structure of the disks they are going to copy.

An IMAGE file or what I call an IMAGE file is just a file that contains a complete image of a floppy disk. It contains none of the header information that a DSK file has and as such may not run on emulator.

Creating Newdos/86/90 image files using a 4P

I recently received a request for copies of some of my Model 3 Newdos/Hardos boot disks**.**
I could have run off some copies, stuck a stamp on them and entrusted them to Aussie Snail Mail but as I had previously made images and DSK files of some of Dick Smith's MicroDos programmes I thought I would try the same process on the 4P and then email the image files with instructions on how to convert them into working floppies. Pleased to say that for once all went according to plan.

The process involves extracting all the sectors from the original disk and then saving them to an IMAGE file. That IMAGE file can then be emailed to another user and converted back to a working floppy or converted to a DSK file that will run on an emulator. Of course the reverse of all of this is also possible and a DSK file can be converted back to a floppy.

The following instructions detail creating an image file from a Newdos86 system disk and assume standard 40 track SSDD drives.

1) Run Super Utility and from the main menu select option 8 and set the configuration of drive 0 to suit the disk being copied. In this case it was a 40 track Newdos disk.
Drive 0 - N3DR',40,17 - This will change depending on the type of DOS and configuration.
Set drive 1 to -
Drive 1 - T4D',42,01

You will notice that I have set drive 1 to 42 tracks. The standard single sided drives are not large enough to hold a 40 track image file but we can work a swifty to get around the problem. Most drives will step an extra 2 tracks and this will give us enough space to hold the file. Thanks to Ray Whitehurst (Audronic) for coming up with this trick.

Remove the Super Utility disk and insert the WRITE PROTECTED copy of the disk to be copied into drive 0 and a blank floppy into drive 1.

2) From the main menu select
3 - Format
1 - Standard Format
Drive 1
This disk will be the IMAGE DISK and will hold the image file.

3) To stop Super Utility from saving part of the image file in last half of track 0 we create a small file that will fill the remainder of track 0.
From the main menu select
7 - File Utilities
10 - Build File
File to Create - XYZ/FIL:1
Grans to Allocate - 2

Next we will build a file to hold the image.

10 - Build File
File to Create - xxxxxxxx/IMG:1
Grans to Allocate - 120

4) From the main menu select
1 - Zap Utilities
4 - Copy Sectors
Source - 0,0,0
Sector Count - 720
Destination - 1,2,0
Mount Prompts - N

If the gods are happy, **xxxxxxxx/IMG:1** should now be an exact copy of the disk in drive 0 and should start at track 2, sector 0 of drive 1 and run for a contiguous 720 sectors.
It will not contain any of the header information that a DSK file has and as such may not run

on an emulator, but can be made to do so.

We can now boot LSDOS and transfer **xxxxxxxx/IMG:1** to the PC via an RS232 or the FreHD's SD card using EXPORT2 and then email the image file to the lucky recipient.
As I had a number of disks to convert I reused the IMAGE DISK by just renaming the image file to whatever was appropriate.
Having received the image file via the magic of email the lucky recipient will want to transfer it to floppy, which is explained in the next chapter.

SAVING THE DISK IMAGE TO FLOPPY

Firstly we need to make a disk to hold the IMAGE file. As above we will need to make a 42 track disk.

1) Boot LSDOS 6.3.1 and format a disk in drive 1 with 42 tracks and the directory on track one.
FORMAT :1 (D=1,C=42)
This will be the IMAGE disk.

2) Connect the 4/4P to the PC via the RS232, fire-up your favorite terminal programme and transfer the image file to the IMAGE disk or if you own a FreHD, you can use IMPORT2 to transfer the file to floppy. With any luck it should save at track 2, sector 0 and in the case of a 40 track SSDD disk, run for a contiguous 720 sectors. If it doesn't then the command **MEMORY (A="A",B=2)** should force DOS to save it at track 2, sector 0.

3) Run Super Utility and set the configuration to
Drive 0 - N3DR',40,17 - This will change depending on the type of DOS and configuration.
Drive 1 - T4D',42,01

4) Insert a blank disk into Drive 0 and from the main menu select
3 - Format
1 - Standard Format
Drive 0

5) When formatting has completed return to the main menu and select
1 - Zap Utilities
4 - Copy Sectors
Source - 1,2,0
Sector Count - 720
Destination - 0,0,0
Mount Prompts - N

6) When Super Utility has finished copying return to the main menu and select
5 - Repair Utilities
4 - Read Protect Directory
Drive 0

Super Utility will respond with

Cannot Locate Directory
Track,Sector Count

Enter 17,10 - This will change depending on the type of DOS and configuration.

Job done, the disk "should" now boot.

Creating a DSK file.

While an IMAGE file is OK for emailing and making a working floppy it may not run on an emulator without a bit more work.
The following text will explain how I created a DSK file from an image of my NEWDOS80 system disk. Super Utility's configuration will need to be altered to suit other operating systems and disk formats.

1) To make a DSK file we need TRSTOOLS to create 2 virtual floppies. Firstly a blank 40 track **DSDD** disk and as this will be used to hold the .IMG file we can call this the IMAGEDISK and another floppy in the same size as the original floppy. In this case it was a 40 track SSDD 180k disk. Name it to suit. This disk will be mounted in drive 2 of TRS32.

2) Run TRS32, boot LSDOS, mount the IMAGEDISK in drive 1 and format it with the following parameters -

FORMAT :1(DDEN,SIDES=2,C=40,D=1)

3) Exit TRS32 and use TRSTOOLS to copy the .IMG file to the IMAGEDISK. TRSTOOLS hopefully will save the image file starting at track 2 sector 0.
Run TRS32. The dos command **FREE :dn** should show if it has been saved in the correct place. If all is OK then go to **4)**
If the file has not saved starting at track 2 sector 0 then we need to create a small file that will fill the remainder of track 0.

From DOS enter the commands
FORMAT :1(DDEN,SIDES=2,C=40,D=1)
MEMORY (A="A",B=0)
CREATE XYZ/FIL:1(REC=30) - The size may change depending on the type of disk**.**
Run TRSTOOLS and copy the .IMG file to the IMAGEDISK.

4) Start TRS32 and make sure we have a copy of Super Utility in drive 0, the IMAGEDISK in drive 1 and the blank 40 track SSDD floppy in drive 2.
Run Super Utility and set the configuration to
Drive 1 - T4D",40,01
Drive 2 - N3DR',40,17 - This will change depending on the type of DOS and configuration.

5) Return to the main menu and select
1 - Zap Utilities
4 - Copy Sectors
Source - 1,2,0

Sector Count - 720
Destination - 2,0,0
Mount Prompts - N

6) When Super Utility has finished copying return to the main menu and select
5 - Repair Utilities
4 - Read Protect Directory
Drive 2

Super Utility will respond with
Cannot Locate Directory
Track,Sector Count

Enter 17,10 - This may change depending on the type of DOS and configuration.

If all has gone to plan the disk in drive 2 should now be a valid DSK file that will, in the case of a system disk, boot in drive 0 of TRS32.
The IMAGEDISK can be reused by deleting the old IMG file and copying any new file using TRSTOOLS.

Creating a working floppy from a DSK file.

DSK/DCT supplied with the FreHD and DISKCOPY/CMD will allow you to make working copies of most DSK files that will run on the original hardware. The process is incredibly simple but for those occasional files that it can't handle we need another less simple approach.
The process involves turning a DSK file into an image file that we can then download to the computer either by the RS232 or the FreHD's IMPORT2 utility.
Having created the image file we can then use the process described above in **"SAVING THE DISK IMAGE TO FLOPPY"** to make a floppy that will run on the original computer.
The following procedure assumes that the DSK file that we are extracting the image from is 40 track SSDD. The settings will need to be changed to suit other formats.

1) Use TRSTOOLS to create blank 40 track **DSDD** disk to hold the .IMG file, we can call this the IMAGEDISK. A 40 track DSDD disk will give us more than enough space to hold a 40 track SSDD image.

2) Start TRS32 and mount a disk with Super Utility in drive 0, the DSK virtual disk to be copied, in this case it was a 40 track SSDD virtual disk, in drive 1 and the IMAGEDISK in drive 2.
Run Super Utility and set the configuration of drive 1 to suit the format of the DSK disk. In the case of an LSDOS SSDD DSK file we should have something like this
Drive 1 - T4D',40,20

Set drive 2 to suit the IMAGEDISK
Drive 2 - T4D",40,01

3) From the main menu select
3 - Format
1 - Standard Format
Drive 2

4) When formatting has completed return to the main menu and select
7 - File Utilities
10 - Build File
File to Create - XYZ/FIL:2
Grans to Allocate - 5

Next we will build a file to hold the image.

10 - Build File
File to Create - xxxxxxxx/IMG:2
Grans to Allocate - 120

5) From the main menu select
1 - Zap Utilities
4 - Copy Sectors
Source - 1,0,0
Sector Count - 720
Destination - 2,2,0
Mount Prompts - N

The file **xxxxxxxx/IMG:2** should now be an exact copy of the disk in drive 1 and should start at track 2, sector 0 and run for a contiguous 720 sectors.
We can now extract this file using TRSTOOLS, download it to the TRS80 and create a working disk using the methods described above in **"SAVING THE DISK IMAGE TO FLOPPY"**

All of the above may seem unnecessarily complicated, I wrote it and I'm not sure I understand it, but it does work flawlessly or it has done so far.

Finally a big pat on the back to Kim Watt for Super Utility and Matthew Reed for TRS32 and TRSTOOLS.

--== oØo ==--

EMAILS TO THE EDITOR

Subject :- 64k and MISE

From: Malcom Ramey
To: dustym@fabsitesuk.com
Cc: Peter Bartlett

Dusty, I just finished reading the Dec/13 issue and I wanted to says thanks to all you guys for your continued effort to support the TRS-80 community.

A couple of notes that might be helpful:

Model I 64K upgrade:
Ira - TRS-80.com also has information on a 64k upgrade on his web site. One difference that caught my eye was that the M1 needs a 128 cycle refresh and some of the chips are 256 cycle refresh.
http://www.trs-80.com/cgi-bin/down-software-patch.pl?

Hardware_-_Upgrading_Model_I_to_64K_DRAM.zip

I think some clarification on type and speeds of chips would be useful. For example I would like to know if the *KM4164B-15 *chip, which seems to fairly inexpensive (500/$85), would be a good fit.

MISE - Model I System Expander:

I recently purchased a MISE board from Peter Bartlett. (http://home.comcast.net/~bartlett.p/MISE/) While Ian and his crew are getting a lot of attention for the well done FreHD project (I have 2), we shouldn't overlook this little wonder. Like the FreHD, you get HD support via inexpensive flash memory (CF in the MISE) and RTC. But for a only $100 more, you get VGA, Hires graphics support, Joy stick port and a LAN which has DNS, SNTP along with FTP and TFTP servers. The ability to move files to and from my M1 over the Lan well lets just say the FreHD will only be used on my M4 now on.

While you could keep your M1 setup complete and just hide the MISE off to the side for nostalgia reasons/vintage feel while keeping the advantages of the MISE, you can also keep the vintage feel and connect it directly to the keyboard and along with the 64k keyboard upgrade.
You can have a very compact space saving system without the expansion interface.

The 64k upgrade isn't required for the MISE to be used connected directly to the KB, but the lack of 64K will limited you to using the TFTP server instead of the FTP server. Of course for those without an expansion interface, it's even a better option.

I will cc Pete as I believe he will be able to answer the question on the RAM compatibility. In addition, you will have his email and maybe you can coax him into writing an article about the MISE in the Mar/14 issue.

From the little I know about Pete in recent conversations, he was a chip designer and is focused on the Model I. Once you see the quality of the MISE and his custom drivers and software he developed for it, you will appreciate his efforts.

Pete has other interesting M1 projects that many would find interesting too. As I too am looking forward to learning more.
We'll have to leave the mystery with Pete to reveal when he has time and motivation.

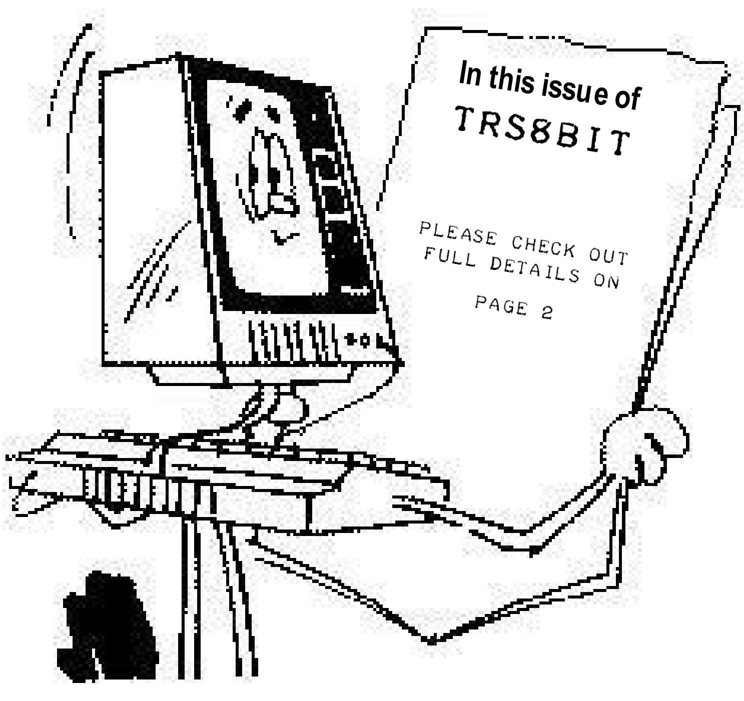
Malcom

TRS8BIT

HI EVERYONE AND WELCOME TO THE 30TH EDITION OF TRS8BIT. ONCE AGAIN, I FEEL WE'VE MANAGED TO PUT TOGETHER A GREAT SELECTION OF ARTICLES WHICH I HOPE YOU WILL FIND OF INTEREST. ONCE AGAIN, MANY THANKS TO ALL OF YOU WHO HAVE SENT IN SUCH A VARIED AND WELL-RESEARCHED SET OF ARTICLES.

THERE ARE GREAT FUN AND GAMES HERE AT 'MILLER TOWERS' AT THE MOMENT. DEE'S GONE OFF AND BOUGHT A 'LENOVO' WINDOWS 8.1 COMPUTER. I MUST ADMIT THOUGH, AT THE MOMENT, ALTHOUGH IT'S ALL VERY STRANGE (AND QUICK!) I'M QUITE IMPRESSED WITH IT.

ANYWAY, DOWN TO BUSINESS. FIRST I MUST ANNOUNCE THAT THE TRAVELLING FREHD IS HERE FOR ANYONE IN THE UK WHO'D LIKE TO TRY IT OUT. IT'S ISSUED ON A FIRST COME, FIRST SERVED BASIS FOR A 30 DAY TRIAL. ONCE YOUR 30 DAYS ARE UP, EITHER JUST POST IT ON TO THE NEXT PERSON OR RETURN IT TO ME. YOU JUST NEED TO EMAIL ME, IF YOU'D LIKE TO HAVE A PLAY WITH THIS FANTASTIC PIECE OF 21ST CENTURY HARDWARE.



I ALSO WANT TO REMIND EVERYONE THAT MAV HAS PLACED VARIOUS VIDEO'S ON UTUBE SHOWING HOW TO BOOT UP THE FREHD, TO ALL MODELS 1,3,4 & 4P WITH OR WITHOUT A DISK SYSTEM.

THERE HAVE BEEN, ONCE AGAIN, QUITE A BIT OF TANDY RELATED ITEMS ON EBAY OVER THE LAST THREE MONTHS. AS I'M SAT HERE WRITING THESE NOTES, I'VE NOTICED THERE'S A VIDEO GENIE ON OFFER TOO!

I WOULD LIKE TO THANK EVERYONE ON THE TRS-80 FORUM FOR THE QUICK REPLIES TO MY REQUEST FOR HELP, TRYING TO ACCESS THE CHARACTER MATRIX. I WAS JUST HAVING A BIT OF FUN, TRYING TO WRITE A 'ONE-LINER' BANNER PRODUCING PROGRAM. I THOUGHT OF TRYING TO ACCESS THIS INFORMATION TO SAVE USING DATA STATEMENTS WITHIN THE PROGRAM.

ENOUGH OF MY RAMBLINGS, READ AND ENJOY EVERYONE! SEE YOU ALL IN SEPTEMBER ..

DUSTY

CONTENTS

The TRS-IDE versus the FreHD

by
Hans Rietveld

As owner of a TRS-80 Model 4P my love for old computers has been completely revitalized. I want this computer to keep as much as possible in his original condition but on the other hand, I do want maximum ease of use and solve its shortcomings from past.

At the moment there is a unique opportunity to solve one of these known shortcomings in two ways.

Unfortunately, the floppy disk system barely meets the modern requirements for ease of use, due to the absence of a reliable hard disk system. There are now two excellent solutions available, namely the FreHD from the highly active duo Ian Mavric and Frederic Vecoven and the TRS-IDE interface of the trio Matt (FireBox), Frank Galphin (Chromedome45) and James Pearce. Sorry for all the names of the people who have done a lot for these projects and I did not mention here, I don't know them.

With one of these two systems you can make your TRS-80 completely up to date and useful for modern applications.

I bought a sets of the TRS-80 adapter and Ian was so kind to send me a FreHD set to evaluate it to convince me of the qualities of this interface. This set is going on tour in Europe for other users to test it for a few weeks. Just ask Ian about it.

First something about the technical aspects. As I have available only a Model 4P, I can only involve my findings on this model. For the other models (Model III and Model I) the possibilities are about the same, but mostly with an extra interface or only minor limitations.

The hardware consists of a board of about 10cm x 8 cm. The quality of the print and the components is excellent. The interfaces in both cases are available as a kit, loose components and as a readymade version.

The self-build of the interface, using a component package is for some experienced hobbyist with some experience and a good soldering iron and solder certainly feasible and cost saving properly.

I have three TRS-IDEs built without any problems and they worked immediately and were completely interchangeable.

The manual and support for the FreHD is excellent and for the TRS-IDE, less obvious and requires some detective work of the builder.

The result is a perfectly functioning modern disk system with a capacity of about 26 megabytes. This is the maximum for the TRS-IDE and intentionally limited to the FreHD (FreHD can actually be configured for up to 65Mb, and standard images are around 42Mb - Ian). This space can be divided into up to eight partitions. This is more than enough, I think it fits all the available software for the TRS80 5 to 10 times over.

The hard disk can be used for LS-DOS 6; LDOS 5 and CP/M.

After this general information I would like to mention the advantages and disadvantages of the two interfaces. The pro's in green the against in red..

The FreHD

Pro:

Very small PCB that is rather easy to build.
Really FAT32 file system, so that the images can be edited and saved on an SD-Card, which can be modified on a PC. The images can be managed and used with most popular emulators for building and maintenance.
No need for modified drivers, just use the default TRS-80 drivers.
The software sees the FreHD as an ordinary TRS-80 hard drive.
Extremely useful support software for file transfer between the hard disk image and the PC, TRS-80 and floppies .
A built-in clock chip for date and time with battery backup, which can be fully integrated into the software.
By using different SD-cards you can easily switch between the various operating systems.
If you use the modified TRS-80 boot-rom, you get a menu to select one of the operating systems which exist on the SD card.
Automatically boot without a floppy in the drive once boot-rom installed in TRS-80.
Very active development of supporting software

Against:

Data storage is only possible on a SD card. (No real hard disk or CF-card supported)

Due to the technical expensive components and advanced file system and the RTC is the FreHD considerably more expensive than the TRS-IDE.
Think also about the transport from Australia and customs clearance fees at customs.
Due to extra build in security checking the response and transfer time is a little bit longer than the with the TRS-IDE controller.

The TRS-IDE

Pro:

The ability to use either an IDE hard disk, CF or SD makes the system attractive
With a IDE hard drive is the system super fast.
With multiple partitions and smart boot floppies you have way to build a powerful system that supports more operating systems on one storing device.
Built-into a computer is certainly possible, but requires some ingenuity.
If you build it yourself, you have a real cheap hard disk for your TRS-80.

Against:

Besides the small PCB with the controller there is need for a real IDE hard disk, CF to IDE or SD to IDE adapter.
The system depends on modified drivers for every operating system, so you are dependent of the good will of the developer.
The supporting software is minimal and you will need to make use of utilities that are available in various libraries on the internet
Auto Boot is not available, but may be solved in the future by modifying the existing software and the system rom.
The development of supporting software is at a low level or has completely stopped.
The data on the CF or SD card can not be maintained on the PC, but needs through floppies to happen.

Conclusion

Both interfaces do well where they are made for. They provide your TRS-80 with an extremely powerful hard disk system. It makes working with the elderly computers again a pleasure.

Are you going for a cheap solution then the TRS-IDE is a good choice. If you want more comfort and support, and you are willing to pay a reasonable, but justified price, then go for the FreHD.

A final point to consider is the availability and support in the future. Both projects rely entirely on highly motivated volunteers, but for how long they can and will continue to do so.

But whatever you chose, just do it, you won't regret it.

Hans

hans@rietveldcomputers.nl

Business Time with Kev

Well firstly can I say thank you for all the positive comments I received about my inaugural article. It's greatly appreciated!
As I indicated in my first article, this would be more about the journey in resurrecting two Model 16's and that journey very much continues. Unfortunately my Model 16's are still mothballed – a few unfortunate house moves has meant that stuff, in fact my entire collection of TRS-80's and others, has been stored away and not to be enjoyed. But all that is changing as I write.

So with that in mind I had to cast my mind to what I might write about for this issue of TRS8BIT and then I had an epiphany – as collectors we often talk about our collections and rightly so, but at the moment I'm totally consumed with the facilities I have for housing, managing and enjoying my collection - and therein lay the answer – we don't talk much about where we do what we do.

For me I've just started moving into the new home for my collection after many months of going through the sometimes onerous building cycle. Please keep in mind that I didn't purpose build a facility but a few things occurred along the way that might be useful to others who are considering renovations, building or improving your facilities. I hope these pointers are useful to some of you even though I might be stating the obvious on occasions.

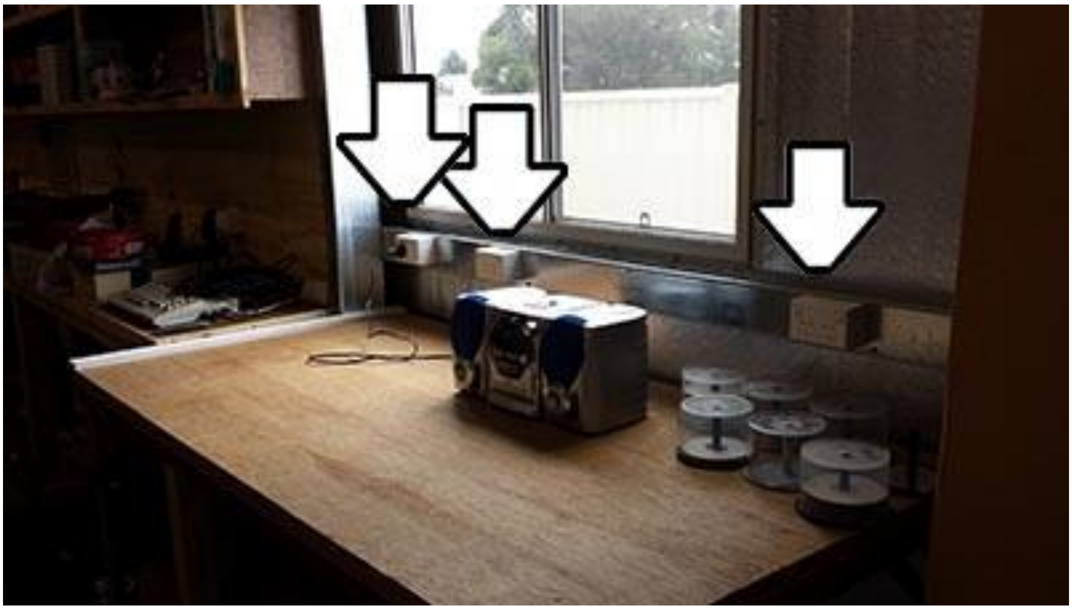
Now in the first instance I did not build a standard shed – one of the notable features is the height of the shed. This served two purposes and one was cosmetic to align with the gutter line of the house (see below).

The other was to provide height in storage which in turn increases my workable floor space which even at this stage of getting sorted out is shaping up to be an issue. Now surprisingly adding some more height to the shed only added a few hundred dollars to the cost. I have some shelves that are 2.4m high, 2.4m long and 1.2m deep so having a shed with 3.3m sides gives me plenty of space to move the big shelves in and also store stuff up and on top.
Now before the building permit was issued the building surveyor signing off on permits required us to put an angled bracket under the bottom of

the shed sides. Now with all due respect to any government readers that we have, my immediate reaction was that this was the work of some bureaucrat who had never seen a country town or even a blade of grass. I was also a bit dark about the $400 it added to the build. Apparently this was part of fire ratings because we live in a rural area. In the photos below (outside the shed is the left photo and inside is the right photo) you can see this structure – it bends down on the outside and up on the inside.

But once done I noticed something. This provided a seal for the corrugations on the outside of the shed (see the left photo). The idea was that it should stop a fire finding its way into the building via the gaps at the bottom of the corrugations on the sides of the shed. If you saw where I lived you'd probably laugh at that notion. But what I have noticed is that by doing this the shed is a lot cleaner, is easier to keep clean, the temperature is more moderate and it's a lot dryer. If I was to build a shed again for this purpose I'd have this done even if I wasn't required to by the building regulators.

Now powering the shed was a bit of challenge given what I was intending to do and the potential demands on the electricity supply. Planning showed that we needed 25 double power points (I don't want a lot of extension cords and power boards running around cluttering the workspace and catching fire) and 14 double fluros. I was deliberately generous with lighting as I wanted no shadows and I didn't want to have to roll in additional lighting for any specific spot in the shed – with the shiny sisalation on all sides and the roof, light saturation in the shed is exceptionally good and I'm quite pleased with the outcome (with one exception which I'll explain later). I also specified double fluros because if that was too much light you can always remove a tube but if you don't have enough light its expensive to retrofit more lights – and remarkably double fluros are usually only a few dollars more than singles. The initial plan by the electrician was to run two circuits in the shed i.e. one down the left side and one down the right side. I wasn't happy with this for several reasons. One was that it meant that the total safe draw down was about 20 amps. I'm one for over engineering things like this because the last thing I wanted is power lines to the shed melting. After discussions with the electrician we asked for four circuits in the shed (excluding lights and overhead fans of course) – one for the rumpus room at the front, one down the left side, one down the right side and one exclusively for my work bench (also on the right side). This provided for plenty of capacity allowing a drawdown of the full 32 amps (not that I intended to do that but it gave adequate capacity if I had a lot of stuff fired up at once). The reason I isolated the work bench was that powering up old machines can be a bit risky and if something was going to go all smoky with lot of sparks I didn't want that taking out the whole shed and house. The three power points over my work bench are on their own circuit, RCD and circuit breaker to minimise any issues like this.



Now my workbench, where I rescue old machines, is probably worthy of some discussion as well although I've had it a lot longer than the new shed. With the exception of the rubber feet I purchased it was made entirely out of salvaged material including a very large and very serious 1½" chipboard top I was given but I made a few improvements. I've applied a sheet of thin plywood to the top – this gives a much nicer finish to work on and move stuff around on and it's much more forgiving if you drop something. You can see it in the photo above (and my apologies for the Commodore in the photo but it's not allowed to live in or go to the other side of the shed where the TRS-80's are) but at one end I've fixed a strip of timber. This gives me something to "push" the object against if I need two hands free to work on it. The bench height is 1m and I'll discuss the significance of that a little later.

Now as to infrastructure, unfortunately I wasn't able to snap some photos before I started unpacking so please excuse the clutter in these photos.

This next photo is looking at the bench work down the left side. You'll note that there are no exposed bolts fixing the posts to the floor, they are concealed inside the post. This was done for four reasons – it's cosmetically pleasing, you don't get those awkward moments where you can't quite fit something under a bench properly because bolts and brackets are in the way, you can't get something out without a fight because its snagged on a bolt or bracket in the way, and it's easier to clean.

Now in some of the photos you'll also notice a few small pallets sitting under benches. I grabbed these a while ago when my wife was working for a mobile phone repair company – they come under boxes of mobile phone spares and are small and easy to handle keeping stuff up off the floor and making thing tidier and easy to clean. In the photo above you'll also notice a row of slats under the bench – these are the slats from an IKEA bed – again I want to keep storage tidy and easier to maintain. (You may have noticed that I'm a self-confessed "scab" and if I see anything that's on its way to the dump I usually see a way I can put it to use.) There's a really small risk of flooding in most places (I don't mean just mine) so if that were to occur everything is off the floor as well. I've also found that with stuff not sitting on the concrete and allowing air to move around it there are less issues with insects living under boxes – notwithstanding the edging placed around the bottom of the shed corrugations crawly insects do still get in, and as I've found, like living under my boxes. Here's a larger view down the left side.

Now the noticeable feature in this photo is the shelves – you can see here that I was able to take advantage of the additional height of the shed – yes I need a small step ladder to reach the top shelves but that's for infrequently accessed stuff. Now an issue a little aside from this – the bench tops are made from flooring material – the recommended spacing of the underside supports is 450mm, however, we set them at 400mm (I like over-engineered) – I can quite safely stand on the benches and I'm about 135kg. It's not quite so obvious here but the bench height is 1m – the standard height for benches in your house may be around 900mm. The reason I set it higher in the shed was because 1m is my standing non-bent-over working height. If I'm working for long periods at 900mm that's going to play havoc with my aging back. Yes, this may be problematic for height challenged visitors to my shed but as I'm going to be the one most often in there it had to suit me. I have an idea how to build small platforms for our smaller folk. But it's the height of the underside of the shelves that is significant here as you can see in the next photo.

You'll notice in the photo above that the back shelves are set higher than the shelves running down the right side of the shed. The reason for this is not TRS-80 related but it does help. While the Model 16's are big machines compared to their Model 4, 3 and 1 cousins I actually have some other machines in my collection that have 19" RGB CRT monitors (Apollo Workstations for those who may be interested). If anyone has one of these you'll understand this – they are very big (and extremely heavy – they're a serious bit of glass) and I wanted them to fit comfortably under the shelves so I had plenty of working space around them including in front of them.

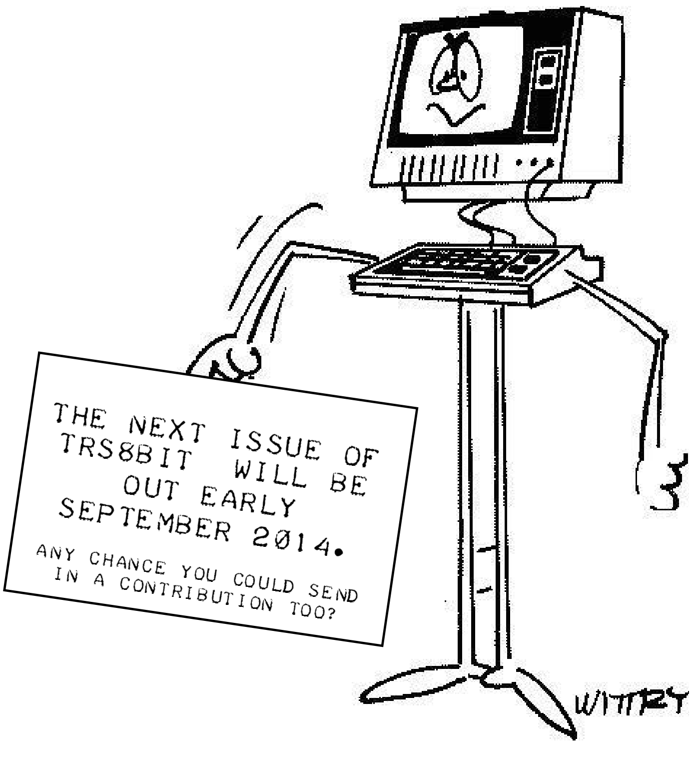
For the observant you'll also notice that the bench width in the photo at the back is deeper – its 1m deep whereas the bench tops running down the right and left side of the shed are narrower. Again this is to accommodate the larger machines to ensure there is plenty of space around them for working and peripherals such as printers, external drives etc. There's nothing worse than trying to use a machine where the front or the keyboard is hanging over the edge.
For the interested here is a larger shot (below) of the work currently underway. You also get a better shot of the floor here. The floor is painted with Berger Jet Dry. There were two reasons for this – cosmetic of course and naked concrete has lots of pits in it and these become dust traps. Painting the floor has not only made the shed cleaner but also easier to clean.

On the right you can see my work bench and above and to the left is a fluro. This fluro is on its own switch. There were two reasons for this – one as that I only needed to turn on a single light to go into the shed to grab something rather than the whole 14. The other reason was to provide light over the bench for working and that was a bit of a fail as it's behind me and too far back. In hindsight I should have had an additional light or two installed directly over the bench as we knew where it was going (lighting and power was done when the shed was empty). I can still fix this but at a cost of course.

It's a bit hard to see here but there are two whirly-gigs in the roof. While the intent was good (temperature moderation) it's proved problematic. The issue to look out for here is what happens in a storm. Under normal conditions they're fine but in extreme weather I've found water on the floor under them. In a highly electrified environment that is an issue for me for which I have a plan to resolve but as yet it's untested.

Well I hope there were some useful tips in this for readers and hopefully by the next issue I'll have the 16's out on the bench.

Kevin


THE NEXT ISSUE OF TRS8BIT WILL BE OUT EARLY SEPTEMBER 2014.
ANY CHANCE YOU COULD SEND IN A CONTRIBUTION TOO?
WITTRY

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

IN MAV'S WORKSHOP
by
IAN MAVRIC

MAKING A COMPOSITE VIDEO CABLE FOR THE TRS-80 MODEL I

Finding a TRS-80 Model I is not particularly hard - there are still good numbers of them up for grabs on eBay, Gumtree, CraigsList etc. The computer itself is compact and so easy to store until you wish to use it. Oftentimes they show up for sale without the monitor or power supply. Some time ago in TRS8Bit Volume 7, Issue 3, September 2013, pp. 23-28, I wrote about how to build a power supply. This time I talk about what you can do if the monitor is missing.

TRS-80 MODEL I MONITORS

The Model I monitor (or Video Display as it used to be called) was a stripped out RCA television set with a B&W CRT, or a stripped out Sanyo television set with either a B&W or green CRT. They were both bulky and really only the Sanyo with the green CRT was really anything approaching "good". If you can't find one for sale it's no biggie. A simple cable can be made in about 15 minutes, and it connects to most monitors with a composite input. Many people who collect computers other than TRS-80s have suitable monitors on hand - this cable works with the Apple IIe/IIc monitor, Commodore 1084S, as well as many modern-day flat screen TVs which have a yellow RCA video input.

PARTS NEEDED

RCA cable such as an audio patch cable

5-Pin DIN connector eg. Jaycar part number PP0304

ASSEMBLY

Start by cutting one end off the RCA patch cable. Cut the insulation away and you will find one wire runs through the core, and the other forms the shielding. Tin the ends into two nice thin tips. The ground is the external shielding on the cable, and the actual video signal travels down the core strand. The purpose is to not leave excess solder on the tips, otherwise they will not go into the holes in the 5-pin DIN connector.

The DIN connector comprises of four parts... the pin array (has the 5 pins in it), the upper and lower clam halves, and the rubber plastic cover shield. Before proceeding any further, put the cover shield down the RCA patch cable. If you forget, it's impossible to put it on without undoing the soldered connections you are about to make.

Looking at the pin array, the rear side is where the solder connections are made. They are numbered as so:

Using a multimeter, confirm that the core strand connects to the centre-post on the RCA plug. Solder the ground to pin 4 and the video signal line to pin 5.

Assemble the two clam halves together, press the cable into the retainer at the back of the clamshell, move the black rubber plastic cover shield over it. It's still a good idea to test the cable before assembling the clamshell etc., but I have made so many of these cables now that I don't need to double check.

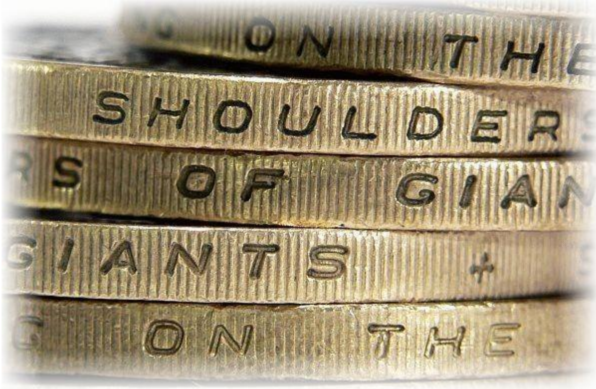
Picture: TRS-80 Model I connected to an Atari RGB monitor

Connect it to your monochrome monitor or LCD TV with composite colour input (normally a yellow RCA female connector) and you should be all set.

Ian.

ianm@trs-80.com

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.



I WAS REMINDED, THE OTHER DAY OF NEWTON'S QUOTE ABOUT STANDING ON THE SHOULDERS OF GIANTS, SO I THOUGH IT MIGHT BE OF INTEREST TO RECALL SOME OF THE KNOWN, AND LESSER-KNOWN NAMES WHO INSPIRED MUCH OF WHAT WE, TODAY, TAKE FOR GRANTED. I START OFF THIS SERIES WITH **GOTTFRIED LEIBNIZ**

Gottfried Leibniz

1646
Born on July 1 in Leipzig
1661
Enrols at University of Leipzig and awarded degree at 17
1660's
Works as lawyer and diplomat. Publishes paper on 'The Art of Combination'
1672
In Paris, he develops the principle of Sufficient Reason
1673
Calculating machine presented to Royal Society in England
1675
Invents calculus independently of Newton
1676
Considers dynamics through the concept of kinetic energy
1678
Appointed librarian and adviser to the Duke of Hanover
1679
Develops binary mathematics
1683
Publishes pamphlet, 'The Most Christian War God', an attack on Louis XIV
1690's
His genealogy of the House of Hanover expands into a History of the World. Develops an interest in linguistics and the origin of languages
1700
Organises Berlin Academy of Sciences
1714
Responsible for establishing the right of succession of George I to the vacant English throne after the death of Queen Anne
1716
Dies in Hanover November 14

COURTESY OF THE SCIENCE MUSEUM

Scientists involved in the fifth generation computers are taking an interest in the work of this 17th century thinker

Gottfried Wilhelm Leibniz was the leading scientific light of his time — the period known as The Age of Reason. He was born in the central European city of Leipzig in 1646 and died in Hanover in 1716. During his life of three score years and ten (the sort of exact figure you might expect from a mathematician), he invented calculus, worked on dynamics, and made contributions to geology, theology, history, linguistics and philosophy. Most important of all, he developed ideas that would be fundamental to the creation of the computer.

He began his travels at the age of 20, after the University at Leipzig refused to confer a doctorate of law on him because of his youth. Throughout his life, without any private means to support him, Leibniz was forced to take up work that hampered his scientific research. In his early twenties he worked as a lawyer and diplomat; later in life he was a librarian and adviser to royalty.

His interests were wide-ranging, and his cosmopolitan nature led to extensive travel in Europe talking with all the great thinkers of his time. Leibniz was a prolific letter writer, as well — engaging in correspondence with over 600 people.

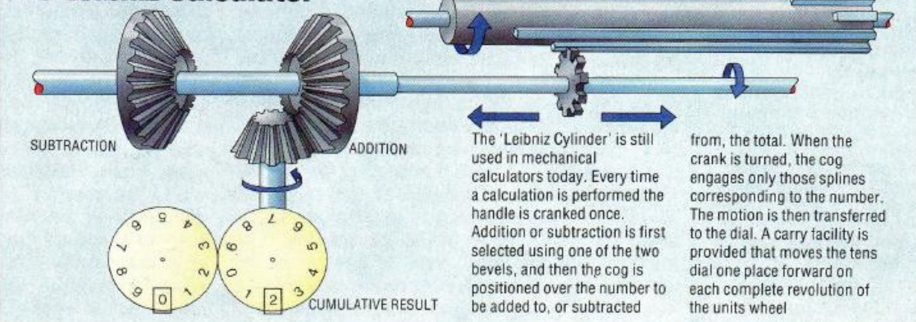
His first important contribution to philosophy came in 1672 when he formulated the principle of Sufficient Reason. This held, simply, that there must be a reason for everything, and 'everything is for the best in the best of all possible worlds'.

Turning his attention to mathematics, he then set to work to perfect the Pascaline adding machine invented by Blaise Pascal in 1642 (see page 86). Leibniz sought to upgrade it so that it would be capable of both multiplication and division. He did so by designing a mechanical device called the Leibniz Cylinder (see below). Leibniz's device was a major breakthrough for its time. Previously, because of the complexity of manipulating Roman numerals, multiplication had been taught only in the higher institutes of learning. A machine that could multiply mechanically made arithmetic more accessible. Once Leibniz had perfected this device, he moved on from base ten arithmetic to consider and formalise binary mathematics.

Leibniz's greatest ambition was to devise a universal language that could use the clarity and precision of mathematics to solve any problem that mankind faced. His language was to use abstract symbols to represent the fundamental 'atoms' of understanding, with a set of rules to manipulate these symbols. His attempt failed; but his ideas were taken up in a more modest way in the early 20th century by Bertrand Russell, who tried to explain mathematics in terms of a formal logical 'language'.

In the last few years, interest has been rekindled in the work of Leibniz by the scientists involved in the long-term project to create the fifth generation of computers. These machines, it is hoped, will be able to solve any problems of human endeavour with the same speed and certainty that computers of today execute mathematical calculations. To do this they will require a new sort of language altogether.

The Leibniz Calculator



The 'Leibniz Cylinder' is still used in mechanical calculators today. Every time a calculation is performed the handle is cranked once. Addition or subtraction is first selected using one of the two bevels, and then the cog is positioned over the number to be added to, or subtracted from, the total. When the crank is turned, the cog engages only those splines corresponding to the number. The motion is then transferred to the dial. A carry facility is provided that moves the tens dial one place forward on each complete revolution of the units wheel

KEVIN JONES

TRS-80 Emulators . com

TRS32: A Model I/III/4/4P Emulator For Windows

written by Matthew Reed

Unregistered Shareware Version:

- Works under all current versions of Windows
- Full Windows application — no low-level hardware conflicts!
- Model I, Model III, Model 4, and Model 4P emulation
- Four floppy disk drives (with optional realistic disk drive sound)
- Cassette tape drive with graphical on-screen controls
- Exatron Stringy Floppy emulation
- Printer support
- Serial port for RS-232 communications
- Joystick support (using a Windows joystick — TRISSTICK and Alpha Products joysticks are emulated)

Registered Version:

- All features included in the shareware version
- Built-in emulation of an Epson FX-80 dot matrix printer (including graphics and control codes)
- High resolution graphics (Radio Shack and Micro-Labs)
- Up to 1 megabyte of additional memory in Model 4 and 4P modes
- Hard disk support
- Orchestra 85/90 music generation

Actual Dotwriter output — emulated FX-80
click for larger sample

Interested?

- Read the TRS32 emulator documentation
- Download the shareware version
- Register online

HOW TO INSTALL 3.5" STANDARD IBM FLOPPY DRIVES IN A RADIO SHACK MODEL III OR MODEL IV

Larry Kraemer

My goal is to get two 3.5" Standard IBM Style Floppy drives working on my Model IV, without modifying the actual 3.5" drive. I have decided that it would be acceptable to modify the existing Floppy Interface cable, or make a new one. Temporarily, I have modified an old Edge Connector to 34 Pin DIN Adapter for my initial testing. (Basically I just lifted the following Pins on the Adapter : 2, 10, 12, and soldered a 1K OHM Resistor from Pin 1 to Pin 2, making the perfect Drive 1 test adapter. Drive 0 will require further modifications as shown later in a detailed image.)

The Radio Shack Model III & IV Floppy Disk Controller (FDC) has the following Signals routed to Connectors J1 (EXTERNAL) and J4 (INTERNAL), each capable of supporting two INTERNAL and two EXTERNAL 5.25" Floppy's at 40 Track, 80 Track, Single Density, Double Density, or Aerocomp FLIPPY Drives (Allowing Storage on the Back Side of TRS-80 Model 1 & 3 Drives).

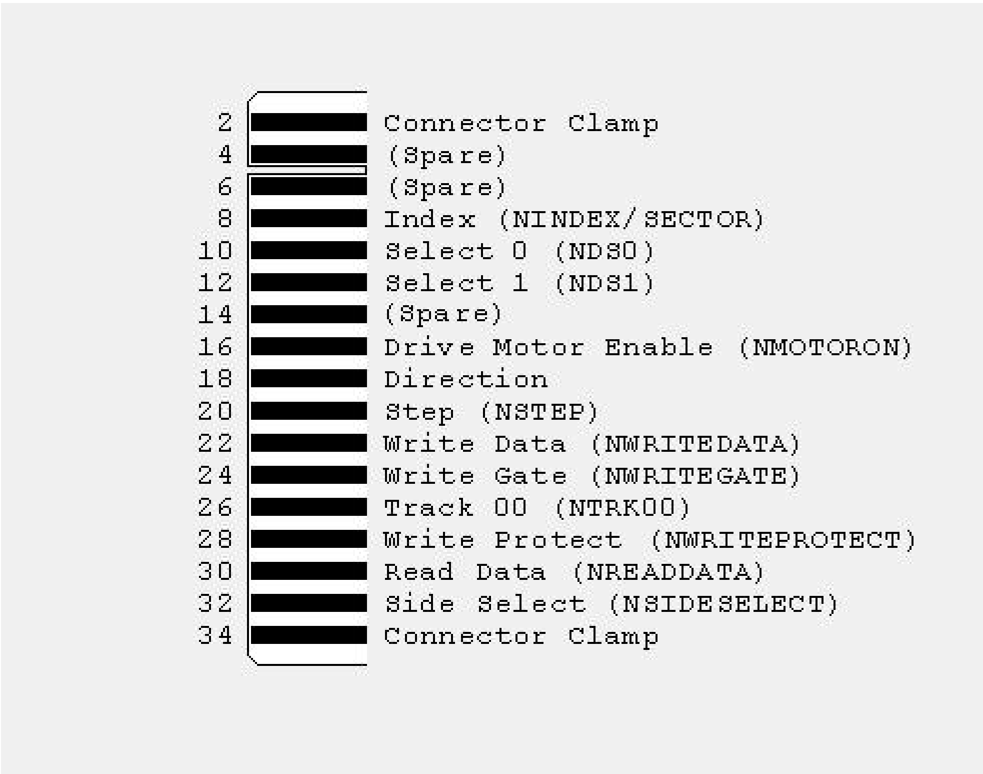


Image referenced above is from: http://pinouts.ru/Storage/5_12_floppy_pinout.shtml

NOTES:

Some Drives use External Terminator Resistors to PULLUP the signals to the + Power Supply Rail. If you are going to be installing (Model 1) drives that have the terminators installed, they must be removed for use with the Model 3 or Model 4 as the FDC already has the Pullup Resistors installed.

The 3.5" Standard IBM Style Floppy Drive uses Pin 2 - Reduced Write Current (/REDWC) from the controller to the drive to control the Density. The Model III & IV do not support this functionality, and the Drive's electronics are Pulled LOW via a 1K Resistor to Common (Pin 1) forcing LOW Density.

TRS-80 Model 1 Floppy Disk Controller Signals to Radio Shack Drives

Pin No.	Signal Name	Description
1	Ground	--
2	N/C	Connector Clamp (Not used on TRS-80 Model 1 Floppy Disk Controller FDC)
3	Ground	--
4	N/C	Reserved (Not used on TRS-80 Model 1 Floppy Disk Controller FDC)
5	Ground	--
6	N/C	Reserved (Not used on TRS-80 Model 1 Floppy Disk Controller FDC)
7	Ground	--
8<--	/Index	NINDEX/SECTOR (0=Index)
9	Ground	--
10-->	/NDS1	0=Drive Select 0 (Physical Drive 1 = External)
11	Ground	--
12-->	/NDS2	0=Drive Select 1 (Physical Drive 2 = External)
13	Ground	--
14-->	/NDS3	0=Drive Select 2 (Physical Drive 3 = External)
15	Ground	--
16-->	/MOTEB	0=Motor Enable Drive 1
17	Ground	--
18-->	/DIR	0=Direction Select
19	Ground	--
20-->	/Step	0=Head Step
21	Ground	--
22-->	/WDATA	Write Data
23	Ground	--
24-->	/WGATE	Floppy Write Enable, 0=Write Gate
25	Ground	--
26<--	/TRK00	0=Track 00
27	Ground	--
28<--	/WPT	0=Write Protect
29	Ground	--
30<--	/RDATA	Read Data
31	Ground	--
32-->	/NDS4	0=Drive Select3 (Physical Drive 4 = External)
33	Ground	--
34	N/C	Connector Clamp (Not used on TRS-80 Model 1)

Pins 2, 4, 6, 34 SPARE in J5 Floppy Disk Bus

TRS-80 Model III Floppy Disk Controller Signals to Radio Shack Drives

Pin No.	Signal Name	Description
1	Ground	--
2	N/C	Connector Clamp (Not used on TRS-80 Model III/IV Floppy Disk Controller FDC)
3	Ground	--
4	N/C	Reserved (Not used on TRS-80 Model III/IV Floppy Disk Controller FDC)
5	Ground	--
6	N/C	Reserved (Not used on TRS-80 Model III/IV Floppy Disk Controller FDC)
7	Ground	--
8<--	/Index	NINDEX/SECTOR (0=Index)
9	Ground	--
10-->	/NDS1	0=Drive Select 0 (Physical Drive 1 = Internal)
11	Ground	--
12-->	/NDS2	0=Drive Select 1 (Physical Drive 2 = Internal)
13	Ground	--
14-->	N/C	(Not used on Model III FDC)
15	Ground	--
16-->	/MOTEB	0=Motor Enable Drive 1
17	Ground	--
18-->	/DIR	0=Direction Select
19	Ground	--
20-->	/Step	0=Head Step
21	Ground	--
22-->	/WDATA	Write Data
23	Ground	--
24-->	/WGATE	Floppy Write Enable, 0=Write Gate
25	Ground	--
26<--	/TRK00	0=Track 00
27	Ground	--
28<--	/WPT	0=Write Protect
29	Ground	--
30<--	/RDATA	Read Data
31	Ground	--
32-->	/SDSEL1	TRS-80 Model III & IV Floppy Disk Controller this is Side Select for the Read Head
33	Ground	--
34	N/C	Connector Clamp (Not used on TRS-80 Model III/IV)

Pins 2, 4, 6, 14, 34 SPARE in J1 (EXTERNAL) & J4 (INTERNAL)

TRS-80 Model IV Floppy Disk Controller Signals to Radio Shack Drives

Pin No.	Signal Name	Description
1	Ground	--
2	N/C	Connector Clamp (Not used on TRS-80 Model IV Floppy Disk Controller FDC)
3	Ground	--
4	N/C	Reserved (Not used on TRS-80 Model IV Floppy Disk Controller FDC)
5	Ground	--
6	N/C	Reserved (Not used on TRS-80 Model IV Floppy Disk Controller FDC)
7	Ground	--
8<--	/Index	NINDEX/SECTOR (0=Index)
9	Ground	--
10-->	/NDS1	0=Drive Select 0 (Physical Drive 1 = Internal)
11	Ground	--
12-->	/NDS2	0=Drive Select 1 (Physical Drive 2 = Internal)
13	Ground	--
14	N/C	(Not used on TRS-80 Model IV Floppy Disk Controller)
15	Ground	--
16-->	/MOTEB	0=Motor Enable Drive 1
17	Ground	--
18-->	/DIR	0=Direction Select
19	Ground	--
20-->	/Step	0=Head Step
21	Ground	--
22-->	/WDATA	Write Data
23	Ground	--
24-->	/WGATE	Floppy Write Enable, 0=Write Gate
25	Ground	--
26<--	/TRK00	0=Track 00
27	Ground	--
28<--	/WPT	0=Write Protect
29	Ground	--
30<--	/RDATA	Read Data
31	Ground	--
32-->	/SDSEL1	TRS-80 Model III & IV Floppy Disk Controller this is Side Select for the Read Head
33	Ground	--
34	N/C	Connector Clamp (Not used on TRS-80 Model III/IV)

Pins 2, 4, 6, 14, 34 SPARE in J1 (EXTERNAL) & J4 (INTERNAL)

3.5" Standard IBM Style Floppy Drive A Pin Out (On Twisted Cable - IBM Style Drives shipped Strapped for Drive 1)

Pin No.	Signal Name	Description
1	Ground	--
2--->	/REDWC	Density Select 0=Low/1=High
3	Ground	--
4	N/C	Reserved
5	Ground	--
6	N/C	Reserved
7	Ground	--
8<---	/Index	0=Index
9	Ground	--
10-->	/MOTEA	0=Motor Enable Drive 0
11	Ground	--
12-->	/DRVSB	Drive Select 1
13	Ground	--
14-->	/DRVSA	Drive Select 0
15	Ground	--
16-->	/MOTEB	0=Motor Enable Drive 1
17	Ground	--
18-->	/DIR	0=Direction Select
19	Ground	--
20-->	/Step	0=Head Step
21	Ground	--
22-->	/WDATE	Write Data
23	Ground	--
24-->	/WGATE	Floppy Write Enable, 0=Write Gate
25	Ground	--
26<--	/TRK00	0=Track 00
27	Ground	--
28<--	/WPT	0=Write Protect
29	Ground	--
30<--	/RDATA	Read Data
31	Ground	--
32-->	/SIDE1	0=Head Select
33	Ground	--
34<--	/DSKCHG	1=Disk Change/0=Ready

Standard IBM Style Floppy Drive A/B Twist Pinout

--	Controller	Drive A	Drive B	Description
Wire 1-9	1-9	1-9	1-9	No Change
Wire 10	10	16	10	Motor Enable Drive 0/1
Wire 11	11	15	11	Ground, No Change
Wire 12	12	14	12	Drive Select 0/1
Wire 13	13	13	13	Ground, No Change
Wire 14	14	12	14	Drive Select 0/1
Wire 15	15	11	15	Ground, No Change
Wire 16	16	10	16	Motor Enable Drive 0/1
Wire 17-34	17-34	17-34	17-34	No Change

So, the conductors 10 through 16 are twisted for IBM, but what you don't know is that pin 10, AND Pin 16 are ASSERTED (HIGH) for the Motor ON signal. Your older computer doesn't follow this IBM standard, so we need to do one small modification to make the signals proper for older computers.

Instead of twisting conductors 10 thru 16, just cut off the last connector and now twist just conductors 10 thru 12. Now replace the connector accordingly. This keeps conductor 16 MOTOR ON for ALL Drives in the proper position. (Only Drives 0 & 1 can now be used......but if you route the conductor from Pin 10 to 14 (for D2) or Pin 6 (for D3) those drives can also be used, assuming your Drive has the appropriate jumpers that can route the Drives Electronics to the proper pin on the connector.) I hope this is clear enough text to not be confusing. Maybe reading it a couple of times will help it sink in. There is a chart below showing the same information.

ACTUAL WIRING CONVENTION THAT IBM USES by also duplicating the Signal on Pin 10 that Pin 16 has using a Twist Cable.

6-->	/DRVSD	Drive Select 3 **
8<---	/Index	0=Index
10-->	/DRVSA	Drive Select 0 ***
12-->	/DRVSB	Drive Select 1 ***
14-->	/DRVSC	Drive Select 2 **
16-->	/MOTEA & B	0=Motor Enable for All Drives

NOTES:
** - Some Model Drives (TEAC FD-235HF) Have jumper positions for Drives 2 & 3 along with 0 & 1.
*** - Drive Select Wiring will be determined from the Actual Drive and if it has jumpers or not. If no jumpers are on the Drive, then it's Default is to be Strapped for Drive 1. Otherwise wire the Proper Drive Select from the Controller to the associated Pin shown for the Drive, and set the Jumper for that Drive.

DRIVE	PIN	DRIVE STRAPPED FOR
0	10	D0
1	**12**	**D1 DEFAULT** (Jumper - connects the (D1) Pin 12 or (D0) Pin 10 Signals to the Drives Electronics)
2	14	D2
3	06	D3

TYPICAL INTERCONNECT WIRING

TRS-80 FDC Controller PC-AT Style Controller — 3.5" disk drive Based on TEAC FD235HF

Sig Name	Sig	Sig. Name	Sig Name	Sig	Ground
Connector Clamp * #	02	0=Double/1=High Density ----->>--------------	REDWC	02	01
Spare *	04	N/C ->>-		04	03
Spare *	06	Drive Select D -------------------->>---------------	Drive Select 3	06	05
Index **	08	Index ---------------------------------<<--------------	Index	08	07
Select 0	10	Drive Select A -------------------->>--------------	Drive Select 0	10	09
Select 1	12	Drive Select B -------------------->>--------------	Drive Select 1	12	11
Spare *	14	Drive Select C -------------------->>---------------	Drive Select 2	14	13
Motor Enable	16	Motor Enable A&B --------------->>--------------	Motor Enable	6	15
Direction	18	Direction Select ------------------>>--------------	Direction Select	18	17
Step	20	Head Step ------------------------->>---------------	Head Step	20	19
Write Data	22	Write Data ------------------------->>---------------	Write Data	22	21
Write Gate	24	Write Gate ------------------------>>-------------	Write Gate	24	23
Track 00 **	26	Track 00 ----------------------------<<-------------	Track 00	26	25
Write Protect **	28	Write Protect ----------------------<<-------------	Write Protect	28	27
Read Data **	30	Read Data -------------------------<<-------------	Read Data	30	29
Side Select	32	Side Select ------------------------>>--------------	Side Select	32	31
Connector Clamp * ##	34	Disk Change/Ready -------------<<-------------	DC/Ready	34	33

Notes:
Signals denoted in BLUE need to be modified for Drive 0 & Drive 1 (Internal & External)
* - This Signal is not used on the Model III & IV
** - These Signals have Pullup Resistors Installed on the FDC for Internal and External Connectors, so drives that have Terminator resistors installed need them removed.
\# - This Signal needs to be Pulled LOW through a 1K Resistor for Double Density (2D) versus High Density (HD).
\## - This Signal doesn't need to be connected since it isn't used.

This Chart depicts what needs to be done to interconnect all the signals properly, but gets confusing about exactly what is required. So, I've drawn out each interconnecting cable showing the exact wiring for each drive.

Floppy Disk Controller Pin Out for TRS-80 Model III & IV Drive 0 INTERNAL or EXTERNAL

TRS-80 Signals Pin No.	Signal Name	Description		IBM Style Drive Signals Pin No.	Signal Name	Description
			X 1K Resistor to Common	2-->	/REDWC	Density Select 0=Low/1=High
				4	N/C	Reserved
				6	/DRVSD	Drive Select 3
8<--	/Index	NINDEX/SECTOR (0=Index)	-----------	8<--	/Index	0=Index
10-->	/NDS1	0=Drive Select 0	-------|	10-->	/DRVSA	Drive Select 0
12-->	/NDS2	0=Drive Select 1	X | ---	12-->	/DRVSB	Drive Select 1
14	N/C	Not used on TRS-80 Model IV FDC	X	14-->	/DRVSC	Drive Select 2
16-->	/MOTEB	0=Motor Enable Drive 1	-----------	16-->	/MOTEA & B	0=Motor Enable Drives
18-->	/DIR	0=Direction Select	-----------	18-->	/DIR	0=Direction Select
20-->	/Step	0=Head Step	-----------	20-->	/Step	0=Head Step
22-->	/WDATA	Write Data	-----------	22-->	/WDATE	Write Data
24-->	/WGATE	Floppy Write Enable, 0=Write Gate	--------	24-->	/WGATE	Floppy Write Enable, 0=Write Gate
26<--	/TRK00	0=Track 00	-----------	26<--	/TRK00	0=Track 00
28<--	/WPT	0=Write Protect	-----------	28<--	/WPT	0=Write Protect
30<--	/RDATA	Read Data	-----------	30<--	/RDATA	Read Data
32-->	/SDSEL1	Side Select for the Read Head	----------	32-->	/SIDE1	0=Head Select
			X	34-->	/DSKCHG	1=Disk Change/0=Ready

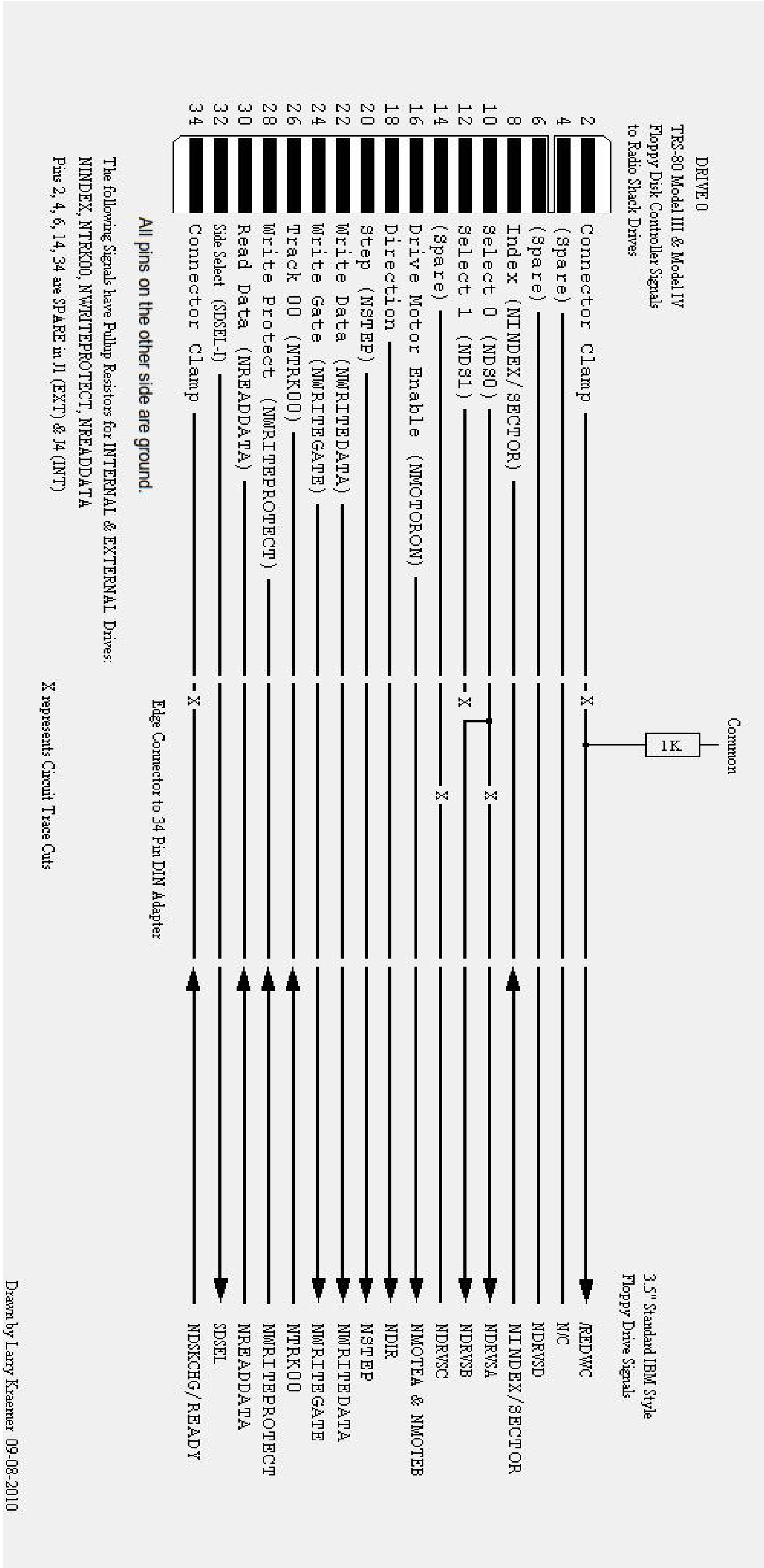
Pins 2, 4, 6, 14, 34 SPARE in J1 (EXTERNAL) & J4 (INTERNAL)
X represents Circuit Trace Cut

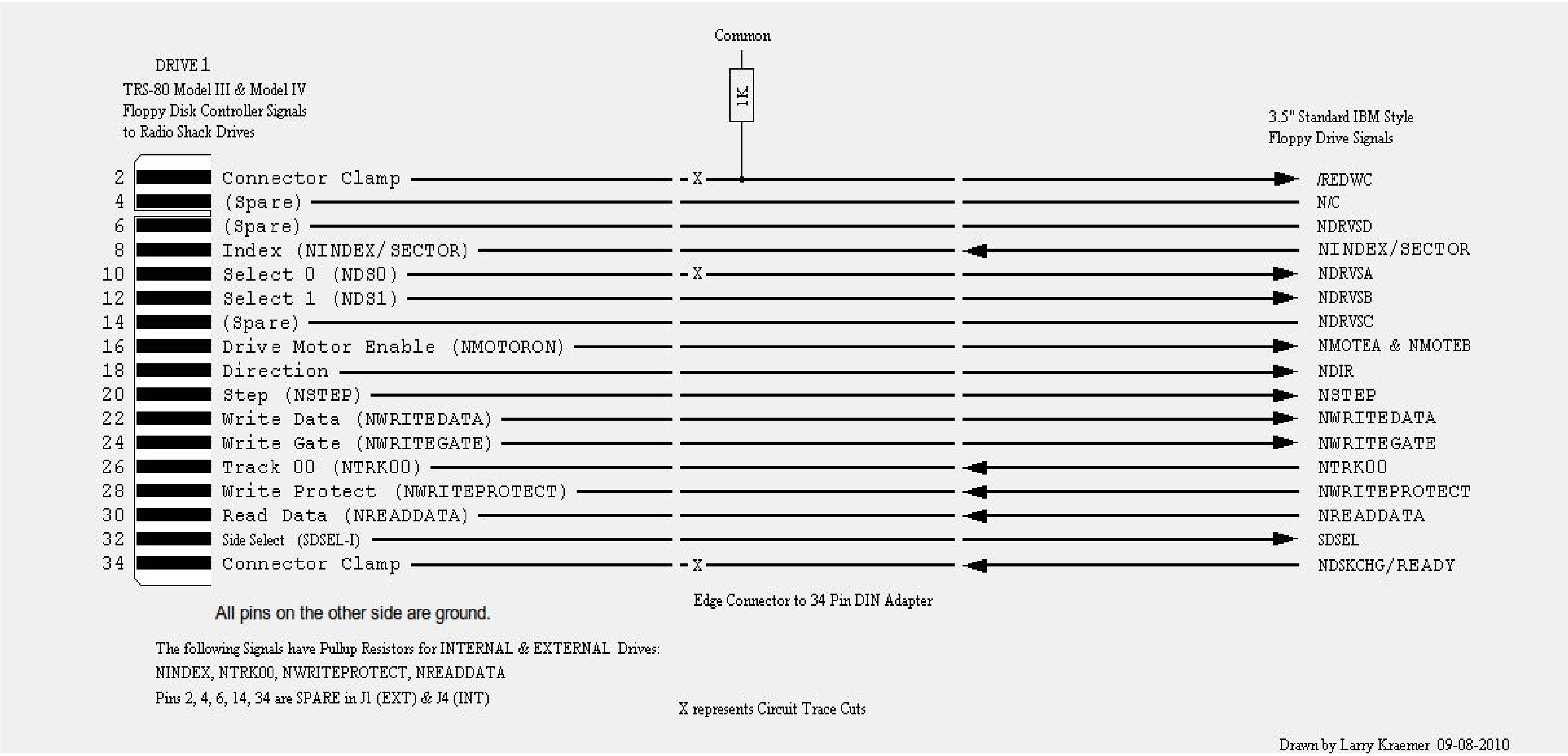
Floppy Disk Controller Pin Out for TRS-80 Model III & IV Drive 1 INTERNAL or EXTERNAL

TRS-80 Signals Pin No.	Signal Name	Description		IBM Style Drive Signals Pin No.	Signal Name	Description
			X 1K Resistor to Common	2-->	/REDWC	Density Select 0=Low/1=High
				4	N/C	Reserved
				6	/DRVSD	Drive Select 3
8<--	/Index	NINDEX/SECTOR (0=Index)	-----------	8<--	/Index	0=Index
10-->	/NDS1	0=Drive Select 0	X	10-->	/DRVSA	Drive Select 0
12-->	/NDS2	0=Drive Select 1	-----------	12-->	/DRVSB	Drive Select 1
14	N/C	Not used on TRS-80 Model IV FDC		14-->	/DRVSC	Drive Select 2
16-->	/MOTE A &	0=Motor Enable Drive	-----------	16-->	/MOTEA & B	0=Motor Enable Drives
18-->	/DIR	0=Direction Select	-----------	18-->	/DIR	0=Direction Select
20-->	/Step	0=Head Step	-----------	20-->	/Step	0=Head Step
22-->	/WDATA	Write Data	-----------	22-->	/WDATE	Write Data
24-->	/WGATE	Floppy Write Enable, 0=Write Gate	--------	24-->	/WGATE	Floppy Write Enable, 0=Write Gate
26<--	/TRK00	0=Track 00	-----------	26<--	/TRK00	0=Track 00
28<--	/WPT	0=Write Protect	-----------	28<--	/WPT	0=Write Protect
30<--	/RDATA	Read Data	-----------	30<--	/RDATA	Read Data
32-->	/SDSEL1	Side Select for the Read Head	-----------	32-->	/SIDE1	0=Head Select
			X	34-->	/DSKCHG	1=Disk Change/0=Ready

Pins 2, 4, 6, 14, 34 SPARE in J1 (EXTERNAL) & J4 (INTERNAL)
X represents Circuit Trace Cut

REF:
http://pinouts.ru/pin_Storage.shtml
http://pinouts.ru/Storage/InternalDisk_pinout.shtml
http://pinouts.ru/Storage/5_12_floppy_pinout.shtml


DRIVE 0
TRS-80 Model III & Model IV
Floppy Disk Controller Signals
to Radio Shack Drives
2 Connector Clamp
4 (Spare)
6 (Spare)
8 Index (NINDEX/SECTOR)
10 Select 0 (NDS0)
12 Select 1 (NDS1)
14 (Spare)
16 Drive Motor Enable (NMOTORON)
18 Direction
20 Step (NSTEP)
22 Write Data (NWRITEDATA)
24 Write Gate (NWRITEGATE)
26 Track 00 (NTRK00)
28 Write Protect (NWRITEPROTECT)
30 Read Data (NREADDATA)
32 Side Select (SDSEL-1)
34 Connector Clamp
All pins on the other side are ground.
The following Signals have Pullup Resistors for INTERNAL & EXTERNAL Drives:
NINDEX, NTRK00, NWRITEPROTECT, NREADDATA
Pins 2, 4, 6, 14, 34 are SPARE in J1 (EXT) & J4 (INT)
Common
1K
X
Edge Connector to 34 Pin DIN Adapter
X represents Circuit Trace Cuts
3.5" Standard IBM Style
Floppy Drive Signals
/REDWC
N/C
NDRVSD
NINDEX/SECTOR
NDRVSA
NDRVSB
NDRVSC
NMOTEA & NMOTEB
NDIR
NSTEP
NWRITEDATA
NWRITEGATE
NTRK00
NWRITEPROTECT
NREADDATA
SDSEL
NDSKCHG/READY
Drawn by Larry Kraemer 09-08-2010


DRIVE 1
TRS-80 Model III & Model IV
Floppy Disk Controller Signals
to Radio Shack Drives
Common
1K
3.5" Standard IBM Style
Floppy Drive Signals
2 Connector Clamp -X- /REDWC
4 (Spare) N/C
6 (Spare) NDRVSD
8 Index (NINDEX/SECTOR) NINDEX/SECTOR
10 Select 0 (NDS0) -X- NDRVSA
12 Select 1 (NDS1) NDRVSB
14 (Spare) NDRVSC
16 Drive Motor Enable (NMOTORON) NMOTEA & NMOTEB
18 Direction NDIR
20 Step (NSTEP) NSTEP
22 Write Data (NWRITEDATA) NWRITEDATA
24 Write Gate (NWRITEGATE) NWRITEGATE
26 Track 00 (NTRK00) NTRK00
28 Write Protect (NWRITEPROTECT) NWRITEPROTECT
30 Read Data (NREADDATA) NREADDATA
32 Side Select (SDSEL-I) SDSEL
34 Connector Clamp -X- NDSKCHG/READY
All pins on the other side are ground.
Edge Connector to 34 Pin DIN Adapter
The following Signals have Pullup Resistors for INTERNAL & EXTERNAL Drives:
NINDEX, NTRK00, NWRITEPROTECT, NREADDATA
Pins 2, 4, 6, 14, 34 are SPARE in J1 (EXT) & J4 (INT)
X represents Circuit Trace Cuts
Drawn by Larry Kraemer 09-08-2010

At this point the FDC to Drive hardware wiring setup should be complete. The FDC Signals to the 3.5" Drive Connector are wired correctly, but some drives allow multiple jumper settings to route these signals to different pins or allow changing signal states. If the STANDARD jumper settings for an IBM Style drive are used, the 3.5" Drive should function properly. There is a possibility of using a 3.5" 720K Drive, 3.5" 1.44K Drive, 720K Diskettes, 1.44K Diskettes (with the Density Hole covered for LOW Density), 40 Track configuration settings, or 80 Track Configuration Settings. I used Montezuma Micro CP/M ver 2.31, and configured my EXTERNAL 3.5" Drive for 80 Tracks, 1 head, and Double Density in the **CONFIG** software. That gave me an 80 Track, SS, DD, 400K Formatted Floppy. (I left the Drive size at 5.25" since a 3.5" drive selection isn't available and a 3.5" drive will function properly when connected.)

I have tested my wiring on three different drives. TEAC FD 235HF-117-U 720K Drive with a 720K Floppy, TEAC FD-235HF-B291-U5 Drive with a 720K Floppy, and a NEC FD1231H 1.44 Drive with a HD floppy with the Density Hole Covered to make a 720K Floppy.

I also used Montezuma Micro CP/M ver 2.31, and configured my EXTERNAL 3.5" Drives for 80 Tracks, 2 heads, and Double Density in the **CONFIG** software. That gave me an 80 Track, DS, DD, 710K Formatted Floppy. (I left the Drive size at 5.25" since a 3.5" drive selection isn't available and a 3.5" drive will function properly when connected.) PIP does copy the files from A: to C: or D: properly.

Using NEWDOS-80 (Model 3 version) on my Model 4, I have set the PDRIVE according to the chart below and everything works accordingly.

Drive Type	Drive Size	Density	Sides	SPT	TI	GRANS	GPL
TD=A	5	Single	Single	10	A	77	2
TD=B	8	Single	Single	17	A or D	157	2
TD=C	5	Single	Double	20	A	141	2
TD=D	8	Single	Double	34	A or D	285	2
TD=E	5	Double	Single	18	A	157	2
TD=F	8	Double	Single	26	A or D	317	2
TD=G	5	Double	Double	36	A	285	8
TD=H	8	Double	Double	52	A or D	573	8

TD = Type Drive
TI = Type Interface (Standard or Apparat Disk Controller)
SPT = Sectors per Track
GPL = Grans per Lump
GRANS = Storage size on the formatted Floppy Disk

Pdrive,0 ……………Shows the Disk Parameters for the 10 drives stored in Memory.

Usage:
PDRIVE,SOURCE,DESTINATION,OPTIONS,,OPTIONS,OPTIONS,……,A

You can change a drives parameters, and read a new disk without a reboot. Use the following command ,A appended at the end:
PDRIVE,0,3,SPT=20,TC=40,A
PDRIVE,0,3=1,A

PDRIVE,0,3,TD=A,SPT=10,TC=40,A
PDRIVE,0,3,TD=G,SPT=36,TC=80,A
(If you don't add the ,A you will need to reboot after changing the PDRIVE Parameters.) PDRIVE Parameters are stored in RAM and will be RESET in the system on a reboot.

AMPRO Little Board Z80 CP/M Computer

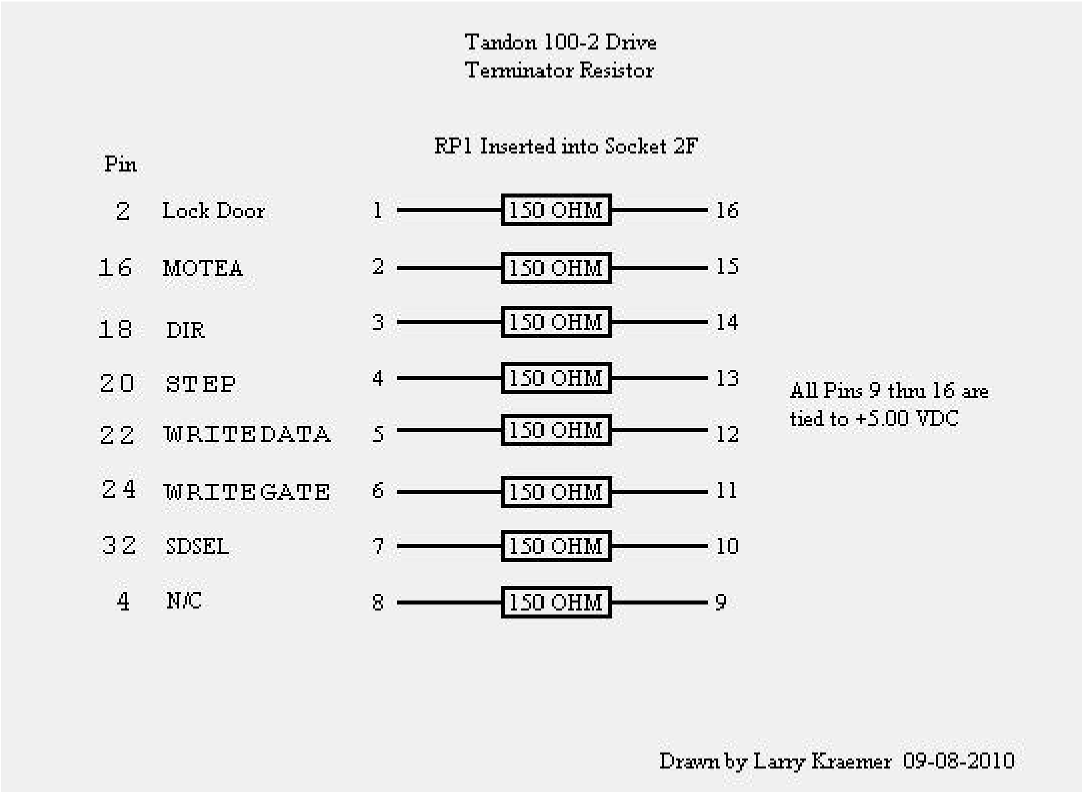
The Ampro Little Board Computer supports four Drives. If you are using Drives such as the Tandon 100-2 with the Ampro, the manual states that the Terminator resistor (RP1 in Socket 2F) needs to be REMOVED in all Drives except for the last Drive on the Floppy Cable.

I am including a Photo of what Signals are associated with these PULLUP Resistors. Drive Pins 2, 4, 16, 18, 20, 22, 24 and 32 are PULLED HIGH through a 150 OHM DIP Resistor. So, you should also be able to use a 3.5" Floppy Drive with the Ampro Computer.

I now have two TEAC FD-235 HF Drives connected to my Ampro Little Board Z80 Computer working correctly.

ENJOY!

Larry Kraemer 09-08-2010
373 CR 250
Chaffee, MO. 63740
ldkraemer@gmail.com
10-18-2010 - Updated 10-09-2012 for Model 1 FDC Interface.
05-01-2014 – Updated address



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.

IN MAV'S WORKSHOP

by

IAN MAVRIC

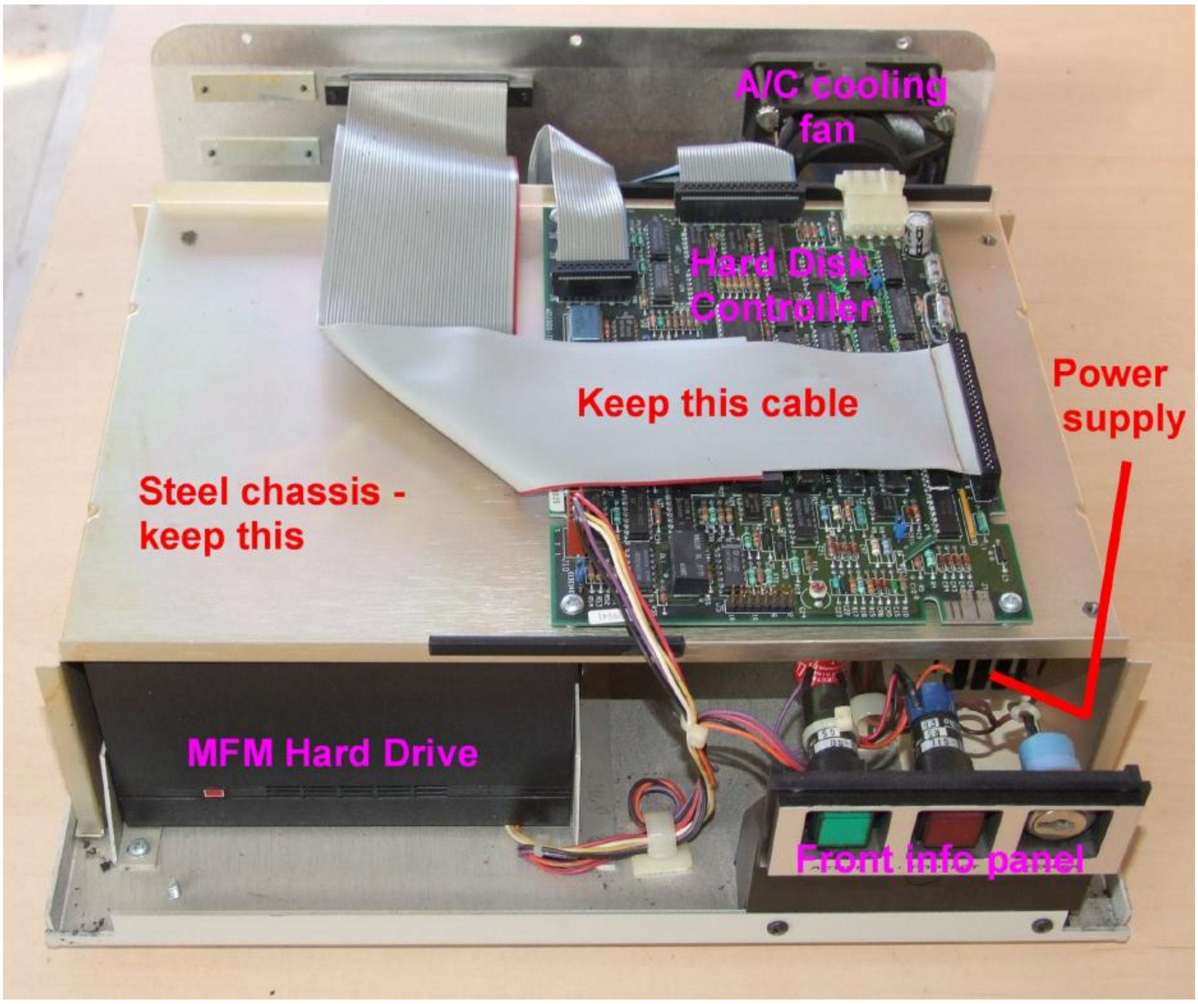
PROJECT 2: MOUNTING FreHD IN A TRS-80 HARD DRIVE CASE

Last time I talked about installing a FreHD and M1 HD adapter in to an external disk drive case, along with a disk drive, to have a single unit connected to the Model I computer. It was a neat elegant solution that worked well, and also proved you could place the Model I hard disk adapter some 90cm down the cable away from the EI. It worked perfectly.

This time, I'm going to explain how to install a FreHD into a TRS-80 Hard drive enclosure. In 2013 I wrote a 4 part series on restoring, connecting, formatting and using the Radio Shack hard drives. In 2013 at the same time the FreHD team ironed out the last of it's few bugs and started marketing the device. FreHD is sold as either a partially for fully populated PCB, it's up the buyer to determine their casing and power supply requirements. It didn't take me long to realise the Radio Shack hard drive has all the requirements to power and house FreHD. Most Radio Shack hard drives sold on eBay are in a poor state and beyond economical repair, so retro-fitting a FreHD makes even more sense.

INSIDE THE RADIO SHACK PRIMARY HARD DRIVE

You open it up by undoing the three machine screws on the back, and sliding the whole top forwards and up. What you are then looking at is a hard disk controller on top of a steel cage assembly. Under it are the MFM hard drive itself, and a power supply. As well as the outer case, we will be re-using the pressed steel cage assembly and power supply, the power key-switch/activity light/protect light assembly, and the internal 50-way cable.

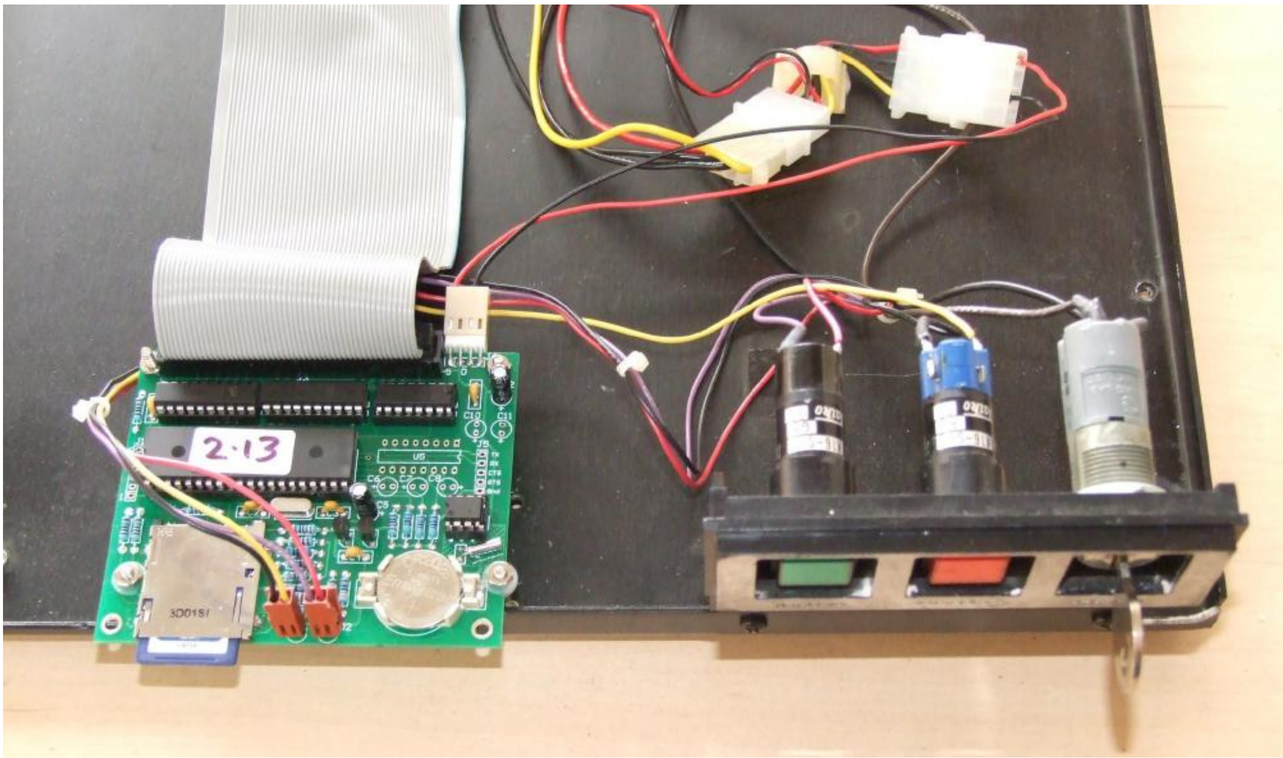


Unplug everything you can see and undo the 4 machine screws which hold down the steel cage. Lift it up about 10cm and disconnect the cables for AC and power distribution from the power supply. Remove the cooling fan and the hard drive. Clip or unsolder the wires from the hard drive and the Activity and Protect lights. Leave enough length on the wires from the lights to reach the FreHD green and red LEDs. If you need to recap the power supply, now is a good time to do it, refer to TRS8Bit Vol 6, Issue 4, page 10.

The reason we remove the AC cooling fan is to replace it with an 80mm 12V DC cooling fan. This is important because the power supply is a switching supply which won't work properly without a load on the 12V and 5V outputs. The 12V powers the cooling fan and the 5V powers the FreHD.

MOUNTING THE FreHD

You could mount FreHD anywhere inside the enclosure, and leave one SD card in it with one image on it and what do you have? Basically a replacement that offers no more functionality than the original TRS-80 hard drive. It will work fine. But since one of the nicer features of FreHD is the ability to change SD cards for ones with other software and images on them, it makes sense to mount the FreHD in a manner where the SD card can be easily removed. Fortunately for us the air grille on the front of the hard drive enclosure has sufficient room to slide an SD card through, so if you mount FreHD close enough to the front, at the correct height, inserting the SD card works quite well.



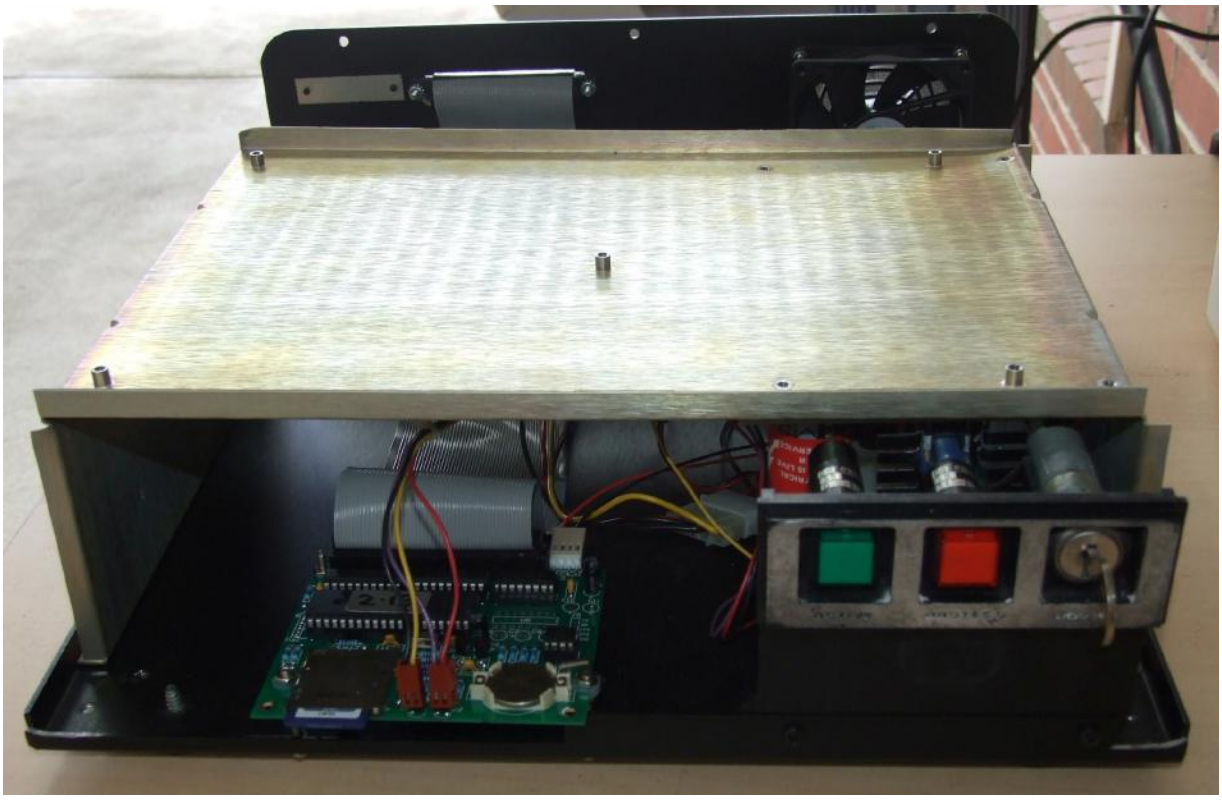
By removing the front plastic facia from the top cover, you can determine where the best place for the FreHD is. Two holes will need to be drilled, one next to the SD card socket and one next to the battery clip. Look carefully - there is space with no electronic connections which you can drill through. You place the leading edge of the SD card socket against the plastic facia, and by using long screws with multiple nuts and washers, you can position the FreHD at the correct height so that the SD card socket lines up with the bottom cooling vent on the plastic facia. This part of the process took the longest amount of time, but once done correctly the SD card inserts and ejects without contacting the plastic of the cooling vent.

WIRES AND POWER

The two LEDs on the front of the FreHD lend themselves to being connected up front on the information panel of the hard drive, and even the colours match. By removing the incandescent globes from the Activity and Protect light sockets and replacing them with LEDs, then run some wires to the FreHD LEDs. What I did was install 2-pin sockets where the LEDs were on the FreHD, and ran wires up to the front panel. You could also hard wire them to the FreHD.

For power, we have the convenient power loom left over from the old hard drive application. One end plugs into the power supply, another end into the hard drive, and the other end to the HDC. For our new application the hard drive power connector goes to the 12V 80mm cooling fan, and we make an adaptor to connect two wires of the HDC end (5V and Gnd) to the FreHD. Use a multimeter to make sure you have the correct wires and polarity.

Finally the large 50-way cable from inside the hard drive can be connected directly to FreHD. Once completed, the steel chassis holding the power supply and be put back in place and if you haven't already done so, remove the HDC.



FINISHING OFF

You may notice that the front doesn't fit on properly without notching out the base where the front screws which hold the FreHD to the chassis foul the plastic facia. Cutting a couple of small notches underneath won't be visible and will allow the lid to be re-fitted to the hard drive unit.

Oh and don't forget to put a CR-2032 battery in before closing your upgraded TRS-80 hard drive.

Ian.

ianm@trs-80.com

SOME HELPFUL HINTS ON DEVELOPING A DISK DEFINITION FROM SCRATCH

Larry Kraemer

If possible, run programs on the CP/M system to help get the necessary information for building a new definition. The following files may be transferred by RS-232C if a Terminal Program is available,
or by PIPMODEM.COM (PIPMODEM.DOC & PIPMODEM.ASM)

1. CPMADR.COM - Finds addresses used by the CP/M system.

2. DPB.COM - Finds Disk Parameter Block information.
   DPB.BAS - Same, but runs under MBASIC. (Located in 22DISK.TXT)

3. PROBE.COM - Probes CP/M System and Drive for information.
   Note: The previous four programs may not locate and display accurate information for some CP/M systems.

4. SKEW.COM - Finds disk SKEW information.

5. Execute the following "STAT" commands on system in question.
   Note: "Control P" will allow a hardcopy of the information.

   STAT DEV: - Displays current DEVICE information.
   STAT VAL: - Displays current STATUS information.
   STAT USR: - Displays current USR information.
   STAT A*.* $S - Displays detailed file information.
   STAT DSK: - Gives some detailed disk information for logged drive.
   STAT A: DSK: - Gives some detailed disk information for A:.
   STAT B: DSK: - Gives some detailed disk information for B:.

If you have a functional CP/M system, you can use DDT to find the Disk Parameter Block and the
information is stored there for each drive.

1. Log the drive you wish to locate the DPB data for. I will use A; for this example
2. Run DDT and enter the code below:

```
DDT
-f0100,0200,00
-A0100
0100   mvi c,1f
0102   call 5
0105   rst 7
0106   <Carriage Return>
-d0100
 0100  0e1fcd0500FF
-g0100
*105
-x
```

The contents of register HL is the address of Drive A: DPB. If you display that address (mine was F5E3) you will see F5E3: 2400040F0154007F00C00020000200.

RPT = 0024 = The number of 128 byte records per track.
BSH = 04 = The block shift count.
BLM = 0F = The block mask.
EXM = 01 = The extent mask.
DSM = 0054 = Disk storage maximum (the largest block number).
DRM = 007F = Directory maximum (the largest directory entry).
DAB = 000C= Directory Allocation Block AL0: = 00 & AL1: = 0C
CKS = 00020 = Directory check size.
OFF = 0002 = Track offset (number of reserved tracks).

SPT: Number of sectors per track. May differ from RPT if physical sectors are other than 128 bytes in size.

SSZ: Sector size code:
- 0 = 128 bytes per sector
- 1 = 256 bytes per sector
- 2 = 512 bytes per sector
- 3 = 1024 bytes per sector

NTK: Number of tracks (35, 40, 77, or 80). The number of tracks can be determined by formatting a Floppy, or with Anadisk. Anadisk will also tell you the sector size and SKEW.

Quite frequently, the only information available when developing a definition for 22DISK is a diskette with a few files and nothing more.

Here's a guide on what to do if you'd like to try working up a definition yourself:

1. Get a copy of Sydex's ANADISK. This utility will furnish far more information than will any other program. A typical disk definition follows:

```
BEGIN AMP4  Ampro - DSDD 96 tpi 5.25"
DENSITY MFM,LOW
CYLINDERS 80 SIDES 2 SECTORS 5,1024 SKEW 2
SIDE1 0 17,18,19,20,21
SIDE2 1 17,18,19,20,21
ORDER SIDES
BSH 4  BLM 15  EXM 0  DSM 394  DRM 255  AL0 0F0H  AL1 0  OFS 2
END
```

New 22DISK definitions require the following information:

DENSITY xx,xx
CYLINDERS x
SIDES xx
SECTORS x,xxxx
SKEW x
SIDE1 0 x,x1,x2,x3, , ,xn
SIDE2 x x,x1,x2,x3, , ,xn
ORDER x
BSH x
BLM x
EXM x
DSM x
DRM x
AL0 x
AL1 x
OFS x (or SOFS x)
COMPLEMENT (unlikely)

New LIBDSK definitions require the following information:

[title]
description = DESC The description of the format as shown by (for example) dskform–help.
sidedness =TREATMENT How a double-sided disk is handled. This can either be alt (sides alternate – used by most PC-hosted operating systems), outback (use side 0 tracks 0-79, then side 1 tracks 79-0 – used by 144FEAT CP/M disks), or outout (use side 0 tracks 0-79, then side 1 tracks 0-79 – used by some Acorn formats). If the disk is single-sided, this parameter can be omitted.
cylinders = COUNT Sets the number of cylinders (usually 40 or 80).
heads = COUNT Sets the number of heads (usually 1 or 2 for single- or double- sided).
sectors = COUNT Sets the number of sectors per track.
secbase = NUMBER Sets the first sector number on a track. Usually 1; some Acorn formats use 0.
secsize = COUNT Sets the size of a sector in bytes. This should be a power of 2.
datarate = VALUE Sets the rate at which the disk should be accessed. This is: HD, DD, SD or ED.
rwgap = VALUE Sets the read/write gap.
fmtgap = VALUE Sets the format gap.
fm = Y or N Sets the recording mode - Y for FM, N for MFM.
multitrack = Y or N Sets multitrack mode.
skipdeleted = Y or N Sets whether to skip deleted data.

The LIBDSK Data rate will be one of:
RATE_HD, /* Rate for High-density disc - 1.2Mb in 5.25" 96 tpi drive, or 1.44Mb in 3.5" 96 tpi drive */
RATE_DD, /* Rate for Double-density disc - 360k in 5.25" 48 tpi drive, or 720K in 3.5" 48 tpi drive */

RATE_SD, /* Rate for Double-density disc - 180k in 5.25" 48 tpi drive, or 360k in 3.5" 48 tpi drive */
RATE_ED /* Data rate for 2.8Mb 3.5" in 3.5" 96 tpi drive */

New CPMTOOLS definitions require the following information:

```
diskdef title
  seclen xxx      #= Sectors xx,1024
  tracks xx       #= (Cylinders * Sides) = 80*2 = 160
  sectrk xx       #= Sectors 5,xxx
  blocksize xxxx  #= (128*(BLM+1)) = 2048
  maxdir xxx      #= (DRM+1) = 256
  skew x          #= may be 1 thru 6, or so
  boottrk x       #= OFS = 2
  os x.x          #= 2.2, or 2, or 3
end
```

So, if you know the 22DISK parameters, you can easily fill in the details for LIBDSK & CPMTOOLS. As an example, here is an AMPRO LITTLE BOARD DSDD 96 tpi 5.25" Definition for all three software packages.

```
BEGIN AMP4  Ampro - DSDD 96 tpi 5.25"
DENSITY MFM,LOW
CYLINDERS 80 SIDES 2 SECTORS 5,1024 SKEW 2
SIDE1 0 17,18,19,20,21
SIDE2 1 17,18,19,20,21
ORDER SIDES
BSH 4  BLM 15  EXM 0  DSM 394  DRM 255  AL0 0F0H  AL1 0  OFS 2
END

[amp4]
description = Ampro - DSDD 96 tpi 5.25"
sides=alt          #=/* XXX Provisional depending on 22DISK Definition */
cylinders = 80     #= Cylinders = 80
heads = 2          #= Sides = 2
sectors = 5        #= Sectors 5,xxx
secbase = 17       #= First Sectors from (SIDE1 0 17)
secsize = 1024     #= Sectors xx,1024
datarate = DD      #= Data Rate = SD, DD, HD, ED
rwgap = 12         #=/* XXX Provisional */
fmtgap = 23        #=/* XXX Provisional */
fm = N             #= FM or MFM (Likely MFM)
multitrack = N     #= Most likely N
skipdeleted = Y    #= Most likely Y
```

```
diskdef AMP4
    seclen 1024      #= Sectors xx,1024
    tracks 160       #= (Cylinders * Sides) = 80*2 = 160
    sectrk 5         #= Sectors 5,xxx
    blocksize 2048   #= (128*(BLM+1)) = 2048
    maxdir 256       #= (DRM+1) = 256
    skew 0           #= may be 1 thru 6, or so
    boottrk 2        #= OFS = 2
    os 2.2           #= 2.2, or 2, or 3
end
```

2. Run ANADISK on the diskette in SECTOR EDIT mode with the display set for "HEX".

Note that the interleave (or SKEW) is given, as well as the number, size and addresses of sectors. You now have data for: DENSITY, CYLINDERS, SIDES, SKEW SIDE1 and SIDE2.

3. Locate the directory on the diskette. Almost invariably, it starts on the first sector of a track. If for some reason it can't be found on the first sector of a track, try using the SEARCH feature to find a "FILE" name that exists on the disk. The number of tracks or sectors up the point where the directory begins will give the figure for OFS or SOFS. The directory stands out because it shows file names every 32 bytes, or every other line on the ANADISK display.

Directory was located at Cylinder 0, Side 0, Sector 2:

```
0000  00 32 32 44 49 53 4b 20 20 44 4f 43 00 00 00 80  .22DISK  DOC....
0010  02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00  ................
0020  00 32 32 44 49 53 4b 20 20 44 4f 43 01 00 00 80  .22DISK  DOC....
0030  0a 00 0b 00 0c 00 0d 00 0e 00 0f 00 10 00 11 00  ................
0040  00 32 32 44 49 53 4b 20 20 44 4f 43 02 00 00 80  .22DISK  DOC....
0050  12 00 13 00 14 00 15 00 16 00 17 00 18 00 19 00  ................
0060  00 32 32 44 49 53 4b 20 20 44 4f 43 03 00 00 80  .22DISK  DOC....
0070  1a 00 1b 00 1c 00 1d 00 1e 00 1f 00 20 00 21 00  ............ .!.
0080  00 32 32 44 49 53 4b 20 20 44 4f 43 04 00 00 08  .22DISK  DOC....
0090  22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  "...............
00a0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00b0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00c0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00d0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00e0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00f0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0100  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0110  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0120  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0130  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0140  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0150  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0160  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0170  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0180  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0190  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
```

```
01a0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01b0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01c0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01d0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01e0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01f0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
```

Fig 1

Directory continues on Cylinder 0, Side 0, Sector 3:

```
0000  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0010  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0020  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0030  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0040  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0050  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0060  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0070  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0080  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0090  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00a0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00b0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00c0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00d0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00e0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
00f0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0100  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0110  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0120  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0130  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0140  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0150  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0160  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0170  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0180  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
0190  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01a0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01b0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01c0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01d0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01e0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
01f0  e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5  ................
```

Fig 2

4. Examine the first directory entry. The second 16 bytes (the second line of the entry) gives a list of allocation unit numbers (AUN) belonging to the file. The first one of these will tell how many allocation units have been reserved to hold the directory. Thus, if the first allocation byte is 02, two allocation units have been allocated to the directory. Another thing to note is whether the allocation unit number appears to be an 8 bit or 16 bit quantity. For example, if an entry has the following form:

```
00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 10  .PIP     COM....
04 00 05 00 06 00 00 00 00 00 00 00 00 00 00 00  ................
```

the allocation unit designators are 4, 5 and 6 and are 16 bit quantities. On the other hand, the following item:

```
00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 10  .PIP     COM....
04 05 06 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

shows the same allocation units, but as 8 bit quantities. If 16 bit quantities are involved, more than 255 allocation nits are defined for the diskette, if an 8 bit quantity, less than 256 units. This serves as a check for later computations. In any case, the first allocation number of the first directory entry gives the number of bits to be set in AL0 and AL1.

```
Allocation Unit Number (AUN)
  02          AL0 11000000B All 0
  04          AL0 11110000B All 0
```

The following information shows a detailed breakdown of all the information in the the first and second directory entries.

```
00 - Valid file (E5 for Erased File)
32 32 44 49 53 4b 20 20 44 4f 43 - File name 22DISK  DOC (11 bytes)
00 - Extent (used for large files)
00 - S1 (Digital Research Reserved)
00 - S2 (Digital Research Reserved)
80 - Number of records in this extent
02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 - allocation units
```

Note: 02 is the first AUN.

```
00 - (Continuation of above file)
32 32 44 49 53 4b 20 20 44 4f 43 - File name 22DISK  DOC (11 bytes)
01 - Extent (used for large files)
00 - S1 (Digital Research Reserved)
00 - S2 (Digital Research Reserved)
80 - Number of records used
0a 00 0b 00 0c 00 0d 00 0e 00 0f 00 10 00 11 00 - allocation units
```

Note: Unused allocation units will be 00.

5. Starting with the first directory sector, advance the sector (and track, if necessary) until you find the data belonging to the first file. Unused directory entries are almost always filled out with a pattern of E5's. The number of sectors dedicated to the directory will give an indication of both the number of directory entries and the allocation unit size.

First Program located at Cylinder 0, Side 1, Sector 2:
NOTE: (This was the 9th sector displayed, so 8 are reserved for directory)

```
0000  0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 20 20 20  ............
0010  20 20 53 79 64 65 78 0d 0a 0d 0a 0d 0a 0d 0a 0d    Sydex.........
0020  0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d  ................
0030  0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20  ...............
0040  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0050  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0060  20 20 20 20 20 32 32 44 49 53 4b 0d 0a 0d 0a 0d       22DISK.....
0070  0a 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .
0080  20 20 20 20 20 41 20 43 50 2f 4d 2d 74 6f 2d 44       A CP/M-to-D
0090  4f 53 20 44 69 73 6b 65 74 74 65 20 49 6e 74 65  OS Diskette Inte
00a0  72 63 68 61 6e 67 65 20 55 74 69 6c 69 74 79 0d  rchange Utility.
00b0  0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d  ................
00c0  0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d  ................
00d0  0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20  ...............
00e0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00f0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0100  20 20 20 20 20 20 20 20 20 20 20 20 20 53 79 64               Syd
0110  65 78 0d 0a 20 20 20 20 20 20 20 20 20 20 20 20  ex..
0120  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0130  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0140  20 20 31 35 33 20 4e 6f 72 74 68 20 4d 75 72 70    153 North Murp
0150  68 79 20 41 76 65 2e 0d 0a 20 20 20 20 20 20 20  hy Ave...
0160  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0170  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0180  20 20 20 20 20 20 20 53 75 6e 6e 79 76 61 6c 65         Sunnyvale
0190  2c 20 43 41 20 20 39 34 30 38 36 0d 0a 20 20 20  , CA  94086..
01a0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01b0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01c0  20 20 20 20 20 20 20 20 20 20 20 28 34 30 38 29             (408)
01d0  20 37 33 39 2d 34 38 36 36 0d 0a 0d 0a 0d 0a 0d   739-4866.......
01e0  0a 0c 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a  ................
01f0  0d 0a 0d 0a 0d 0a 0d 0a 20 20 20 20 20 20 20 20  ........
```

Fig 3

Suppose that a our hypothetical diskette shows 8 sectors of 512 bytes allocated to the directory. Since a directory entry is 32 bytes long, the number of directory entries possible for this format is:

16 entries/sector X 8 sectors = 128 entries total

or

((Sector Size in Bytes/32) X Sectors to DIR) -1 = DRM

Thus, DRM is one less than this or 127. Since we know that these 8 sectors represent 2 allocation units or 4,096 bytes, the Allocation Unit Size (AUS) is:

4,096 / 2 = 2,048 bytes

or

((Sector Size in Bytes) X Sectors to DIR) / AUN = AUS

Given this, we can use the following table to compute the values for BSH and BLM:

Unit Size	BSH	BLM	
1,024	3	7	
2,048	4	15	←This one
4,096	5	31	
8,192	6	63	
16,384	7	127	

7. We still need a value for EXM and DSM. DSM is simply the total number of allocation units on the diskette, not counting the system OFS area. Let's assume that our diskette has an offset of 2 tracks and is single-sided with 40 cylinders with 8 sectors per track. The value of DSM is then:

((40 cylinders X 1 side) - 2 tracks) X 8 sectors X 512 bytes) =
155,648 bytes / 2,048 = 76 Allocation units, total

To get EXM, we then use the following table:

Unit Size	DSM < 256	DSM > 255
1,024	0	na
2,048	1	0
4,096	3	1
8,192	7	3
16,384	15	7

8. That's almost the complete picture, but for the SIDE1 and SIDE2 sector ordering. This is the hardest to determine of any of the parameters. Find a readable text file on the diskette and piece the text back together by observing how words break across sector boundaries. After a bit, a pattern in the ordering will become apparent, such as:

1, 3, 5, 7, 2, 4, 6, 8

9. The ORDER value can be determined on double-sided diskettes by observing whether file data appears to move from one track to the next on the same side (CYLINDERS), or appears to go from one side to the other and then advance to the next cylinder (SIDES).

10. COMPLEMENT is rare. Generally, if a diskette appears to be formatted with values of 1A instead of E5, the data on the diskette is complemented.

11. Last resort for information will require a memory dump from F200 hex through FFFF to provide all the DBP information as hex bytes. It will be necessary to dig through 3584 (E00) bytes of data to locate the DPB information. Most systems follow the layout shown below.

(Hex values starting in memory at Fxxx)
RPT: (word) total # of logical 128-byte sectors/track
BSH: (byte) data allocation block shift factor,
determined by data block allocation size
1k=3 2k=4 4k=5 8k=6 16k=7
BLM: (byte) data allocation block mask (2**[BSH-1])
1k=7 2k=15 4k=31 8k=63 16k=127
EXM: (byte) extent mask, determined by the data block
allocation size and the number of disk blocks
1k=0 2k=1 4k=3 8k=7 16k=15 if DSM < 256
1k=n 2k=0 4k=1 8k=3 16k=7 if DSM > 256
DSM: (word) disk size in blocks-1; determines the total
storage capacity of the disk size
DRM: (word) directory size-1; determines the total number
directory entries that can be stored on this
drive. DRM: = ((# dir entries) / 4) -1
AL0: (byte) alloc0 (1100 0000) 2 blocks/dir
(1000 0000) 1 block/dir
AL1: (byte) alloc1 (0000 0000)
Note that AL0: and AL1: are taken together
as a single sixteen-bit mask for the number
of blocks of directory allocation.
(1100 0000 0000 0000) 2 blocks/dir
(1000 0000 0000 0000) 1 block/dir
CKS: (word) directory check vector size
fixed media: CKS = 0
removable media: CKS = (DRM + 1)/4
OFF/SOFF: (word) # of reserved system tracks or sectors (offset)
at the beginning of the (logical) disk.
Usually the directory begins at the first
sector of the track defined by this number.

By formatting a disk the number of tracks can be determined, which will give the first byte to look for. The SPT will be followed by 00 and the value for BSH and BLM will be the third and fourth bytes.

BSH and BLM will be one of the following Hex values:

BSH 3 or 4 or 5 or 6 or 7
BLM 7 or 0F or 1F or 3F or 7F

The above information should be fairly easily to locate.

READER'S EMAILS, TEXTS & LETTERS

DUSTY,

PETE BARTLETT IS BUILDING 4 MORE 'MISE' FOR THE MODEL I. IF YOU OR ANYONE YOU KNOW WANTS ONE THEY MIGHT WANT TO GET ON THE ORDER LIST BEFORE THEY ARE GONE.

PRETTY COOL TO BE ABLE TO USE THE MODEL I ON THE NETWORK. I USE THE FTP SERVER HE DEVELOPED FOR THE M1 ALL THE TIME TO TRANSFER STUFF BACK AND FORTH. IT HAS HD VIA COMPACT FLASH, RTC, JOYSTICK, VGA AND BOTH HIRES OPTIONS BUILT IN.

IT BOOTS DIRECTLY FORM THE HD WITHOUT ANY MODS TO THE M1, EVEN IAN OWNS ONE :)

I THINK THESE WILL BE HIS LAST BATCH. HE'S CURRENTLY DESIGNING A E/I REPLACEMENT.

I'M TRYING TO CONVENIENCE MY WIFE I NEED ANOTHER ONE, I HAVE 4 MODEL 1'S :)

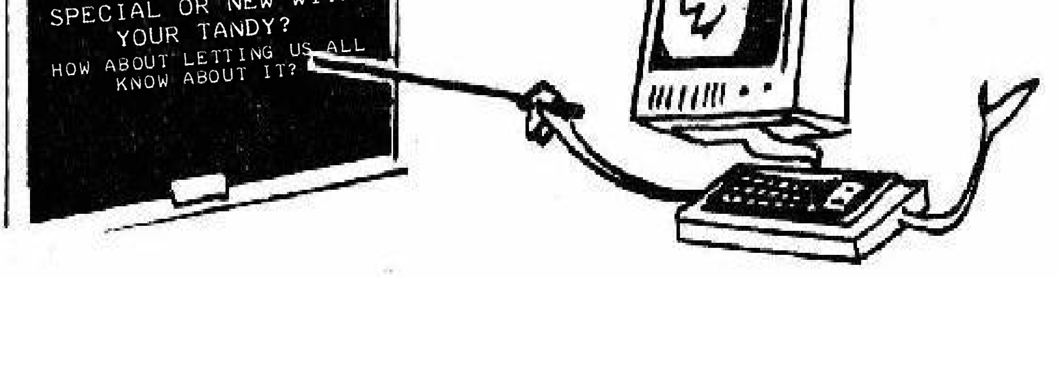
THANKS FOR MENTIONING PETE'S MISE, HE'S NOT VERY GOOD AT PROMOTING HIS PROJECTS. ALSO I ENJOYED THE ISSUE WITH THE MODEL I 1.3 ROMS. I HAVE ONE 1.3 SYSTEM, I'M PLANNING TO GET SOME OF THOSE EPROMS AND UPGRADING THE OTHER 3!

KEEP UP THE GOOD WORK ON THE NEWSLETTER,

http://home.comcast.net/~bartlett.p/MISE

MALCOM RAMEY

THANKS FOR THE EMAIL MALCOM, I'VE PLACED A LINK TO PETE'S MISE WEBSITE, AS ABOVE, ON TRS-80.ORG.UK FOR EVERYONE TO USE.



ASK MAV
ABOUT THE TRS-80 MODEL I MICROCOMPUTER
by Ian Mavric

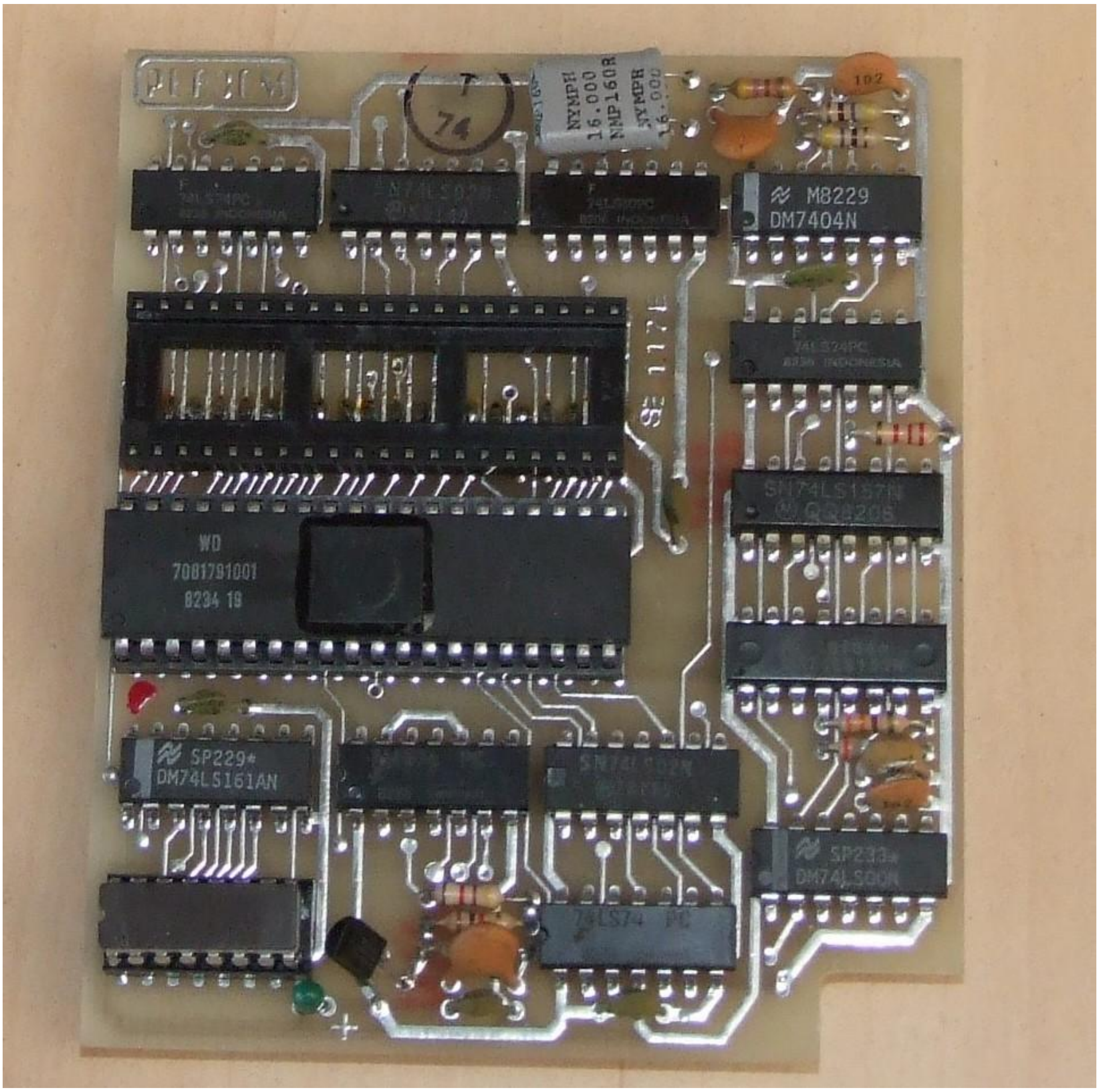
1.
Q. How do I upgrade a Model I to a disk system?
A. The Model I is a modular system so you need to plug in additional hardware to upgrade the machine to a disk system. Before you start you need to make sure your keyboard unit is 16K and Level II Basic. You can't add disks to a 4K or Level I system. You need an Expansion Interface, which has the floppy disk controller inside it, and an external disk drive. When buying these things from eBay, CraigsList etc., bear in mind you need the interconnect cables from the keyboard to the Expansion Interface, and the disk drive cable. Also remember the Expansion Interface needs it's own power supply so make sure it comes with one. The Expansion interface holds up to 32K extra memory, and it's advisable to make sure you get one of these, or the additional 16K chips to upgrade a 16K or 0K EI to 32K. You then need to choose a DOS, see below.

Picture: a typical Model I disk system

2.
Q. Why are there so many DOSes, and which is the best?
A. When Radio Shack first released TRSDOS for the Model I it was full of bugs. This drove other enterprising programmers to write new DOSes which worked properly, and many more functions and fewer bugs. Although Radio Shack did fix up TRSDOS by version 2.3, other more powerful DOSes like Multidos, Newdos, and LDOS had a foothold. By the 1980s the ability to control a hard drive became a necessity, and the main two DOSes for hard drive usage are Newdos/80 2.5 and LDOS 5.3.1 These therefore make the best choice for DOS for floppy disk and hard disk based Model Is (this includes FreHD).

3.
Q. I have a 4K Level 1 Model I - how do I upgrade it to 16K and Level II BASIC?
A. Although not as rare as the Level I Model III, the Level I Model I is still pretty hard to find. If you find one I would keep it that way for posterity. At the time of writing good numbers of 16K Level II Model Is are still easily found for sale for reasonable prices. Ask around the internet forums or TRS-80 resellers.

4.
Q. What is a double-density adapter and how do I install one?
A. In it's standard form a Model I diskette is single density and holds about 89K of data per disk. Radio Shack designed the system to be able to address up to 4 disk drives, and 356K of online storage was considered good for 1978 - as capable as other similar systems of the time. By 1980 disk drives became more sophisticated, diskettes more reliable and double density operation was perfected, and implemented no newer computers like the Model III. Companies made adapters which installed inside the Expansion Interface which added the necessary circuitry and double density FDC to allow the Model I to operate in double density. Even Radio Shack eventually made their own doubler. The main benefit of a doubler is to be able to store more programs on each disk, and as new programs got bigger and more complex into the 1980s, larger disk capacity was a desirable feature. A single density disk may hold the DOS plus three 16K games, but a double density disk will typically hold 10-12 games in addition to the DOS. To install a doubler you open your EI and remove the FD1771 disk controller chip and install it on the empty socket on the doubler. You then install the doubler into the vacated FDC socket in the EI, and you are done.



Picture: a Percom double-density adapter

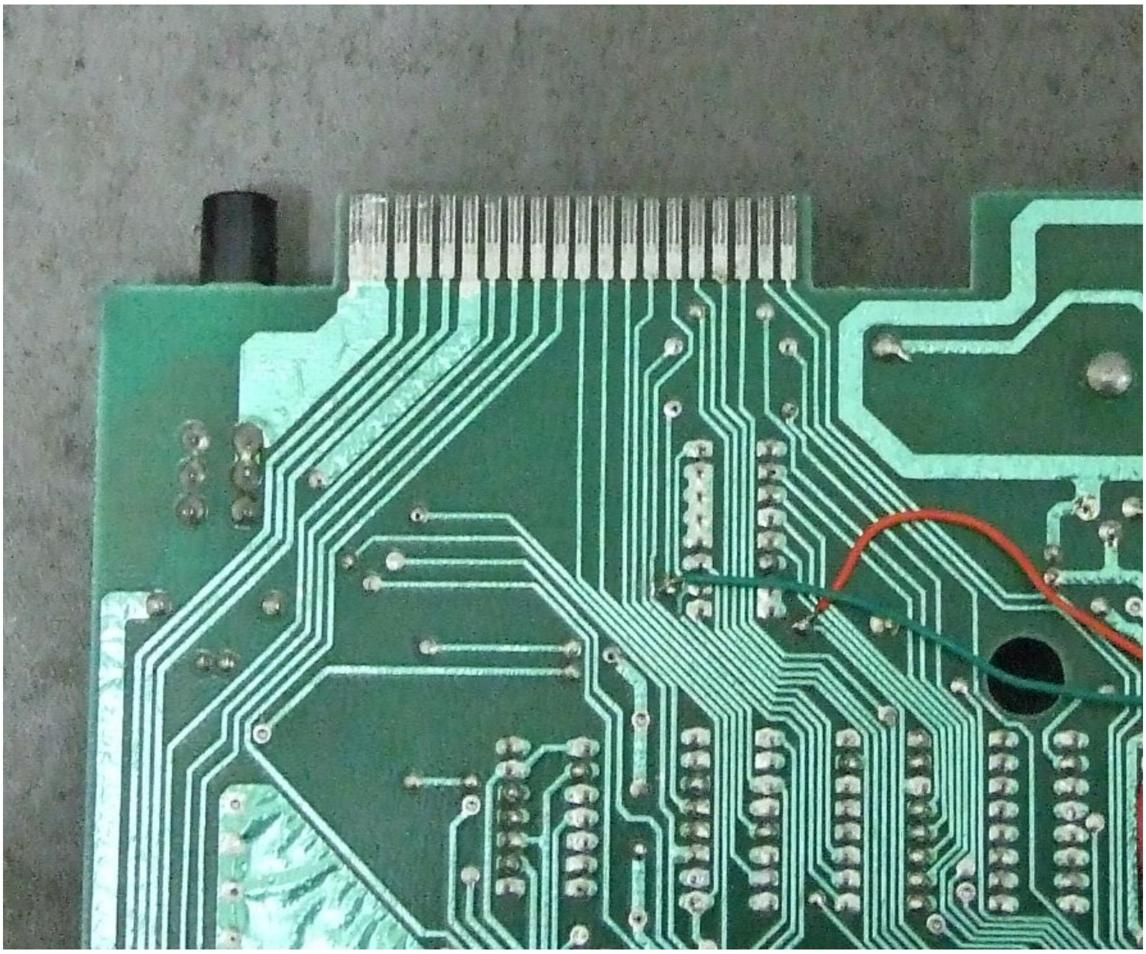
5.
Q. My Model I is mostly reliable but it crashes from time to time.
A. The Model I was not particularly reliable but people who bought it initially didn't know any difference, it was about as reliable as every other machine on the market in its price range, and random re-boots was just part of the life of using a Model I, so you saved and backed up your disks regularly. The main culprits are underwhelming design, software bugs, and solder covered edge connectors.

1. Underwhelming design of the Expansion Interface - you probably noticed the Model I is a very robust design for a computer motherboard in 1977. In fact can and will run for days without

crashing. All this changes when you add an Expansion Interface. Tandy cut a few corners design-wise and the asked the EI to live under a very electrically noisy monitor. That cable connecting the EI to the Model I carries a lot of information every millisecond which requires very precise timing. When one of these goes pear shaped the system crashes. Tandy's solution was the buffered EI cable, a bulge between the M1 and EI, and it helps a bit but the system needs to be treated very respectfully. Around December 1979 Tandy re-designed the Expansion Interface to make it more robust internally and it worked quite well in this form. The buffer cable was gone and the high-speed signals running back and forth between the two units were reduced and replaced with a DDU inside the EI. This is the EI to look for if you are buying one, but like I say to people, the early buffer-box ones are just as reliable 35 down the track, all other things being equal.

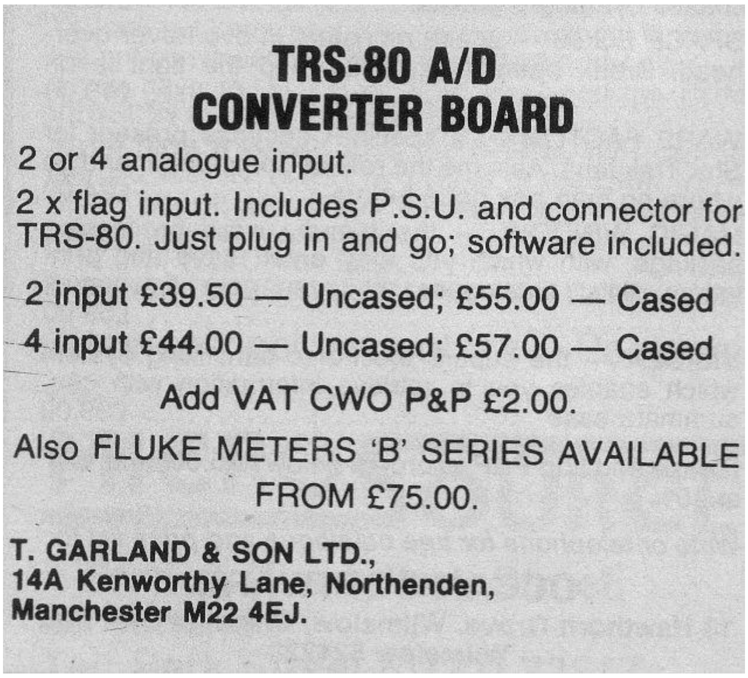
2. Software bugs - early versions of TRSDOS had bugs, and as mentioned earlier, TRSDOS 2.3 was pretty reliable, but still has its bugs. Most of the early versions of the alternative operating systems had bugs as well, so make sure you use the most up-to-date versions. Also did you know there was 4 versions of the Level II BASIC Rom (1.0, 1.1, 1.2 and 1.3) which fixed various problems which caused various crashes. You can tell version 1.3 because it asks MEM SIZE? on boot up. All previous versions ask MEMORY SIZE? and it can be hard to determine which version you have.

3. Solder covered connectors - in the computer world, aside from Tandy's, it's almost impossible to find edge connectors which are not gold plated. Tandy, in an economy move which they stuck with right through their computer producing years, decided it was acceptable to solder coat the edge connectors. Solder, in contact with the metal teeth on the headers of the cable, oxidises over time. (Gold, although expensive, fixes this problem, which is why all computer manufacturers use it these days). Oxide is easy to observe, the edge connectors no longer look silvery-grey, but has black lines - that's the oxide. You need to rub this away periodically with a pink pearl eraser. Although Tandy applied this cheap production philosophy to all their computers, it's particularly dire on the Model I with the traffic going between the M1 and EI, even if you don't clean the others, it's important that these be cleaned regularly. The best EI in the world will still crash if the edge connectors for the I/O bus joining them become too oxidised.



Picture: Oxidised edge-connector

ianm@trs-80.com

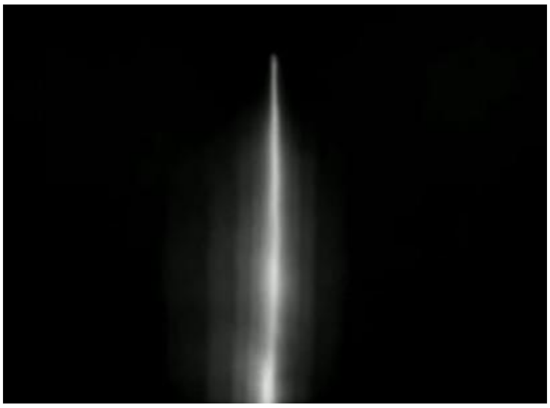


PLAY THE DOCTOR WHO ADVENTURE ON YOUR TRS-80 MODEL I / III
by Ian Mavric

PART TWO - NEW TITLE GRAPHICS

Last issue I wrote about the Dr Who adventure, a program written in the early 1980s by a gentleman named Jim Smith from Newcastle in Australia. The program can be found on Ira's web site if you want to play it, the download is at http://www.trs-80.com/cgi-bin/down-ok-model1.pl?drwhoadv.zip

An interesting and fun game, I recommend anyone to try it out. Because the game is large (it uses most of the memory on a 16K Model I) Jim would have left out a proper title sequence of graphics. Since Dr Who without the intro graphics just isn't the same, I decided to create my own. Once, my son was playing around with BASIC set/reset graphics on my Model I and wrote a simple routine which ran a straight line straight up the middle of the screen and said "Hey Dad, what does this remind you of?". I dunno? Pre-cooked spaghetti noodle? He then explained it was like the start of Dr Who...



Typical of most teenagers he lost interest at this point and I remembered how back in the day I'd spend a lot of time on the primitive TRS-80 128x48 graphics creating representations of things which I found mildly amusing, even if no one else did. Thinking it's been a while since I'd working with TRS-80 graphics to any extent, I set myself a goal to create a proper intro to Jim's program, using only BASIC, and gave myself 5 hours to see what I could come up with. I'm quite pleased with the results myself at the photos below show:


Radio Shack
TRS-80
VIDEO DISPLAY


DOCTOR
WHO
Radio Shack
TRS-80


DOCTOR
WHO

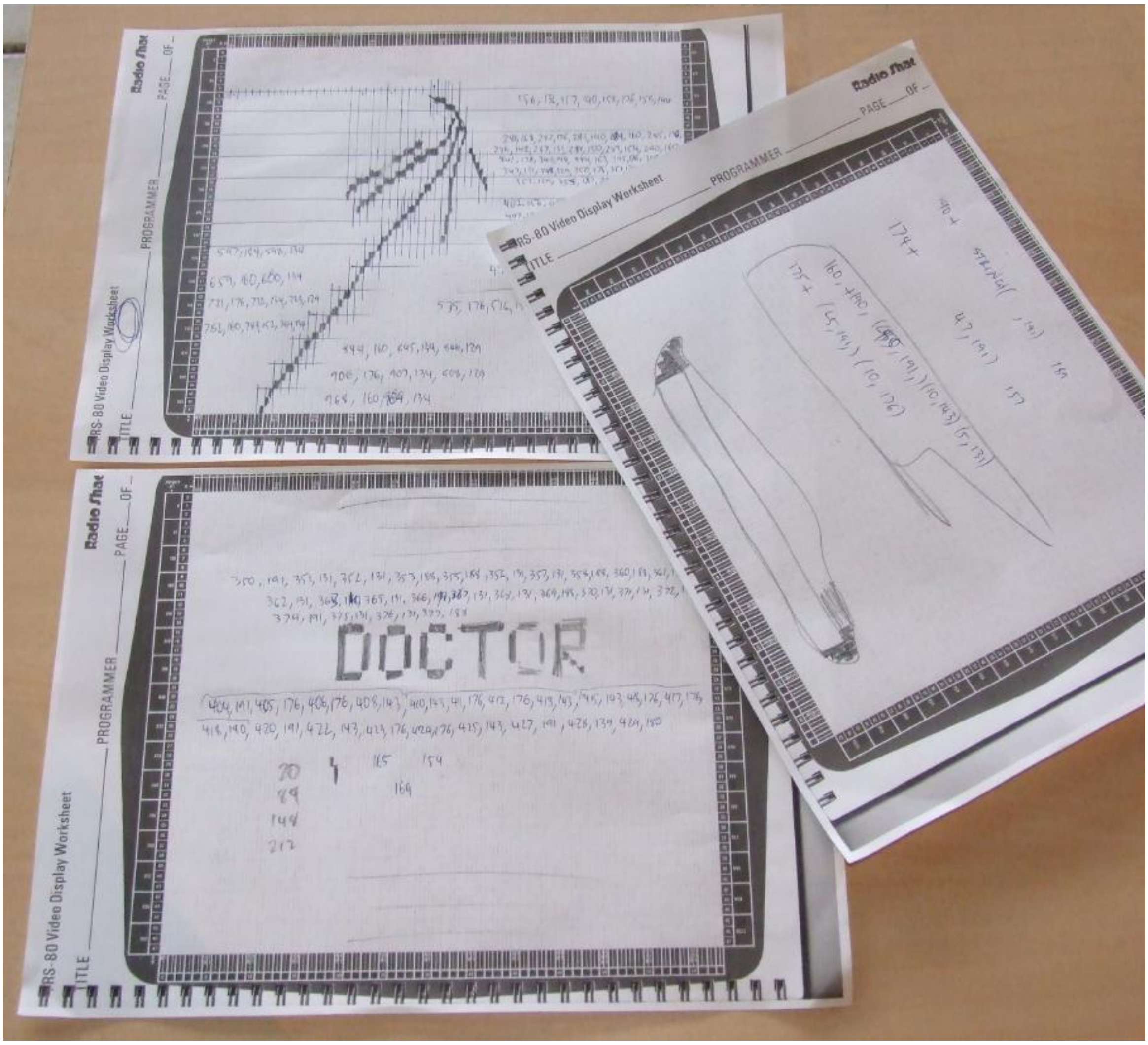

WHO
Radio Shack
TRS-80


DOCTOR
WHO

Animation in BASIC on the Model I leaves a bit to be desired even at the best of time, but I still resisted the temptation to include fast assembly language routines to make more realistic moving images. Even so, my program came in at under 4Kb in size. I employed every string and graphical manipulation I could think of that was fast enough to draw each screen and animate elements where possible. After some time I came to the conclusion that clouds of nondescript blobs of video feedback are difficult to render and animate convincingly. So I chose two clouds and animated them in a sequence which amused me. I used lots of artistic licence and while I think the program does some justice to the original, if William Hartnell were here he'd tsk tsk and comment about the dearth of blobs... and in hind sight if he'd seen a Model I he'd probably say "You call that a computer? It's no WOTAN"

PROGRAMMING TECHNIQUE

Those video display worksheets come in handy when designing graphics displays (that's what they are for) because I'm not smart enough to do screen co-ordinates in my head anymore. Tandy used to sell packs of 100 video display worksheets, unavailable to buy since the late 1980s. Not to worry, each Level II Basic manual and Model III/4 Basic manual has one in it - just photocopy it like I did.



Another thing to remember is that the coding in the Level II Basic interpreter to PRINT@ is much more efficient than the coding of the SET/ RESET routine. In other words for the purposes of fast moving animation, PRINTing the graphics characters to the screen is much faster than SETing/RESETing the individual pixels. And in fact most Basic language games which use animation use this technique rather than SET/ RESET. Chapter 12 of Basic Faster and Better & Other Mysteries book expands greatly on ways to speed up animations and graphical manipulations for those who wish to continue working with the TRS-80 graphics.

PROGRAM LISTING

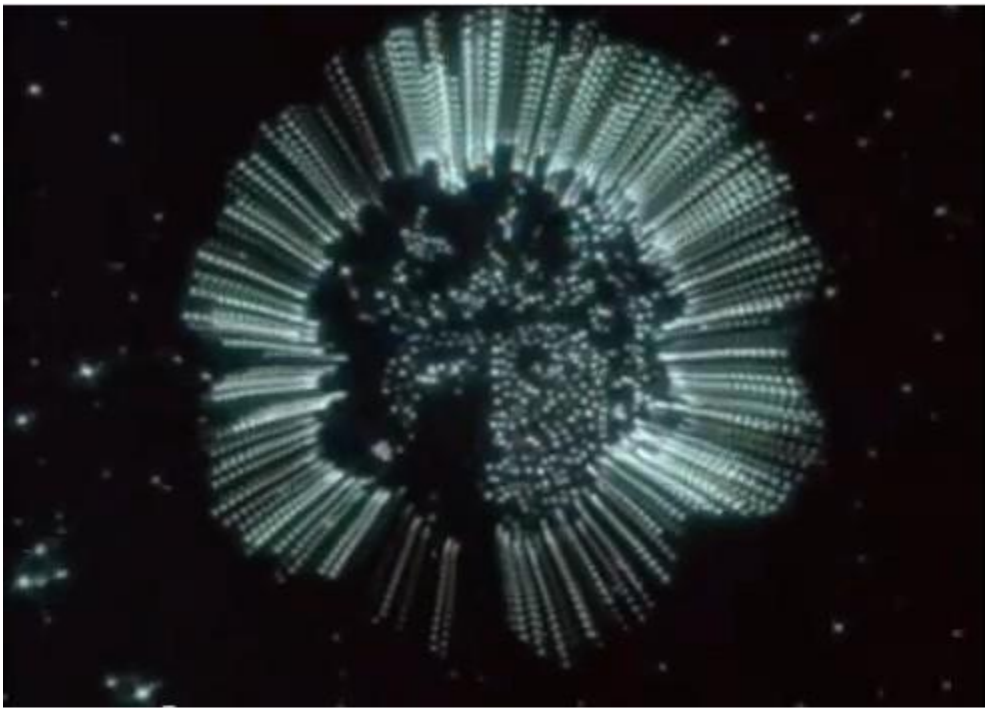
```
10 REM dr who intro

11 CLS:CLEAR500

12 V1$=CHR$(170):V2$=CHR$(149)

20 REM line runs up the screen

22 FORL=992TO224STEP-64:PRINT@L,V1$;:GOSUB 1000:NEXT

30 REM line bends

36 FORL=990TO798STEP-64:PRINT@L,V2$,"  ";:GOSUB 1000:NEXT

37 FORL=798TO606STEP-64:PRINT@L,V1$,"  ";:GOSUB 1000:NEXT

38 FORL=543TO415STEP-64:PRINT@L,V2$,"  ";:GOSUB 1000:NEXT

39 FORL=351TO223STEP-64:PRINT@L,V1$,"  ";:GOSUB 1000:NEXT

40 REM splash screen

41 FORL=1TO70:READA,B:PRINT@A,CHR$(B);:NEXT

49 DATA 156,136,157,140,158,176,159,144,222,160,223,166,224,180

60 DATA
281,160,282,176,283,140,284,160,285,176,286,140,287,131,288,150,2
89,154,290,160,341,176,342,144,344,163,345,176,346,140,347,131,34
8,129,350,176,351,134,352,168,353,129,355,137,356,144

61 DATA
402,156,403,134,404,131,405,176,406,140,407,131,408,129,413,152,4
14,134,417,149,421,139,467,136,468,131,473,160,474,152,475,134,47
9,160,480,133,535,176,536,134,537,129,543,154

62 DATA
597,184,598,134,659,160,660,134,721,176,722,134,723,129,782,160,7
83,152,784,134,844,160,845,134,846,129,906,176,907,134,908,129,96
8,160,969,134

63 GOTO 71
```

```
70 CLS:CLEAR 500:REM upper cloud

71 C1$=CHR$(190)+STRING$(45,191)+CHR$(189):C2$=CHR$(174)
+STRING$(47,191)+CHR$(157):C3$=CHR$(139)+STRING$(45,143)
+CHR$(135):GOTO 91

80 PRINT@8,C1$:PRINT@71,C2$:PRINT@136,C3$:RETURN

90 REM lower cloud

91 C4$=CHR$(160)+CHR$(190)+STRING$(45,191)+STRING$(10,143)
+STRING$(5,131):C5$=CHR$(174)+STRING$(40,191):C6$=CHR$(175)
+STRING$(45,191)+STRING$(5,176):GOTO 100

95 PRINT

96 PRINT@833,C4$:PRINT@896,C5$:PRINT@961,C6$;:RETURN

100 FORT=1TO5:CLS:GOSUB 80:GOSUB96:NEXT

110 REM doctor who large

111 B=0

120 DATA
191,131,131,188,128,188,131,131,188,128,188,131,131,140,128,131,1
91,131,128,188,131,131,188,128,191,131,131,188

121 DATA
191,176,176,143,128,143,176,176,143,128,143,176,176,140,128,128,1
91,128,128,143,176,176,143,128,191,131,139,180

130 FORL=71TO98:READA:PRINT@338+B,CHR$(A);:B=B+1:NEXT

132 B=0:FORL=1TO28:READA:PRINT@402+B,CHR$(A);:B=B+1:NEXT

140 REM who

141 B$=CHR$(191)+CHR$(191)

145 DATA
0,66,130,196,134,70,8,74,138,204,142,78,16,20,84,148,212,86,88,26,9
0,154,218,94,158,32,34,224,226,100,164

149 FORL=1TO31:READA:PRINT@A+525,B$;:NEXT

150 REM flicker clouds
```

```
151 C$="                                                    "

155
FORT=1TO5:PRINT@0,C$;C$;C$;:GOSUB80:GOSUB1000:PRINT@832,C$
;C$;C$;:GOSUB 96:GOSUB 1000:NEXT

160 REM smaller who

165 PRINT@320,C$;C$;C$;C$;C$;C$;C$;:PRINT@475,"DOCTOR";

170 DATA
535,165,600,169,537,154,538,160,602,169,539,154,541,181,605,149,5
42,176,543,186,607,170,545,150,546,131,547,169,609,165,610,176,61
1,154

171 FORL=1TO17:READA,B:PRINT@A,CHR$(B);:NEXT
175
FORT=1TO5:PRINT@0,C$;C$;C$;:GOSUB80:GOSUB1000:PRINT@832,C$
;C$;C$;:GOSUB96:GOSUB1000:NEXT

176 PRINT@460,C$;C$;C$;

180
FORT=1TO5:PRINT@0,C$;C$;C$;:GOSUB80:GOSUB1000:PRINT@832,C$
;C$;C$;:GOSUB96:GOSUB1000:NEXT

998 GOTO 998

999 STOP

1000 REM short pause

1005 FORP=1TO60:NEXT

1010 RETURN
```

A COMPETITION, ANYONE?

Dusty and I like the idea of a competition to see who can design the best Dr Who graphical re-creation of its iconic title sequence on a TRS-80 Model I/III, II/4/12/16/6000, or Coco 1-2-3. Chose any Dr Who title sequence you like, and upload a video to Youtube and email your entry to Dusty. Judging will be by Dee, in December, and the prize will be a FreHD, delivered in January 2015. So you have 6 months to complete and submit your masterpiece. Language and memory requirements are open, as are hardware and memory requirements. Do you have a Hi-Res Model 4 with 128K memory and can program in Z80 assembly language? Then a killer Tom Baker intro would look incredible:



ianm@trs-80.com

TESTING THE Z80 ADDRESS LINES

Larry Kraemer

With a Z80 address line tester, you can test all address lines on any logic board that uses a Z80 chip.

When the Z80 receives instruction hex 00 (a NOP), it does nothing except increment the program counter (PC) and reads the next instruction. If the next instruction is also a NOP, the microprocessor is forced to count through all 65,536 (2^16) possible addresses on its l6-bit address bus.

If you then monitor the address bus lines with an oscilloscope, each address line will display a square wave, with a period twice that of the next lower address line. This gives you a predictable set of signals to trace.

TOOLS REQUIRED

1. Wire cutters
2. Soldering iron
3. Solder

PARTS REQUIRED

1. 40-pin wire wrap socket
2. Z80 Microprocessor
3. Jumper wire

PROCEDURE

Construction of test assembly:

1. Cut pins 7 through 10, and 12 through 15, on the bottom of the wire wrap socket by about 1/4 inch. This will prevent them from making contact with the pins in the socket of the pcb (see Figure 24-1).

2. Wire pins 7 through 10, and 12 through 15 to pin 29 (ground) to force the hex 00 instruction on the data bus (see Figure 24-1).

3. Plug the Z80 into the wire wrap socket. Make sure that its notch faces in the same direction as the notch on the wire wrap socket (see Figure 24-1).

Operation of test assembly:

1. Remove the Z80 CPU from the board to be tested, and set it aside.
2. Install the test assembly (with a known good Z80) in the wire wrap socket. Make sure that the notch faces in the same direction as that on the other chips.
3. Apply power to the logic board.
4. Inspect the address lines for the wave forms listed in Table 24-1.
5. If you do not detect any pulses on the address lines, check for the following signals:

Pin 11 +5V
6 Phase 0 clock
18 HALT (should be high)
24 WAIT (should be high)
26 RESET (should be high)

If any of these inputs are held in an incorrect state, the microprocessor will not work. Before continuing, correct any problems with these lines.

Figure 24-1 shows a top and a bottom view of the wire wrap socket.

Table 24-1
Expected Wave forms

Pin#	**NAME**	**WAVE FORM**
1	A11	xx second period square wave
2	A12	xx second period square wave
3	A13	xx second period square wave
4	A14	xx second period square wave
5	A15	xx second period square wave
6	*CLK	
7	D4	Tied Low
8	D3	Tied Low
9	D5	Tied Low
10	D6	Tied Low
11	+5VDC	
12	D2	Tied Low
13	D7	Tied Low
14	D0	Tied Low
15	D1	Tied Low
16	*INT	
17	*NMI	

18	*HALT	
19	*MREQ	
20	*IORQ	
21	*RD	
22	*WR	
23	*BUSACK	
24	*WAIT	
25	*BUSRQ	
26	*RESET	
27	*M1	
28	*RFSH	
29	GND	
30	A0	xx second period square wave
31	A1	xx second period square wave
32	A2	xx second period square wave
33	A3	xx second period square wave
34	A4	xx second period square wave
35	A5	xx second period square wave
36	A6	xx second period square wave
37	A7	xx second period square wave
38	A8	xx second period square wave
39	A9	xx second period square wave
40	A10	xx second period square wave

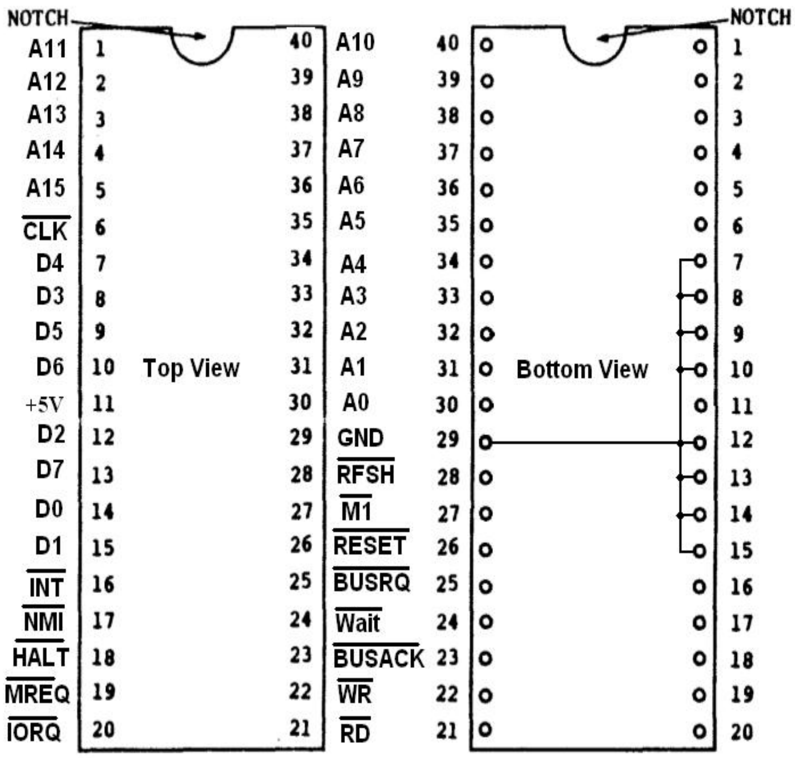
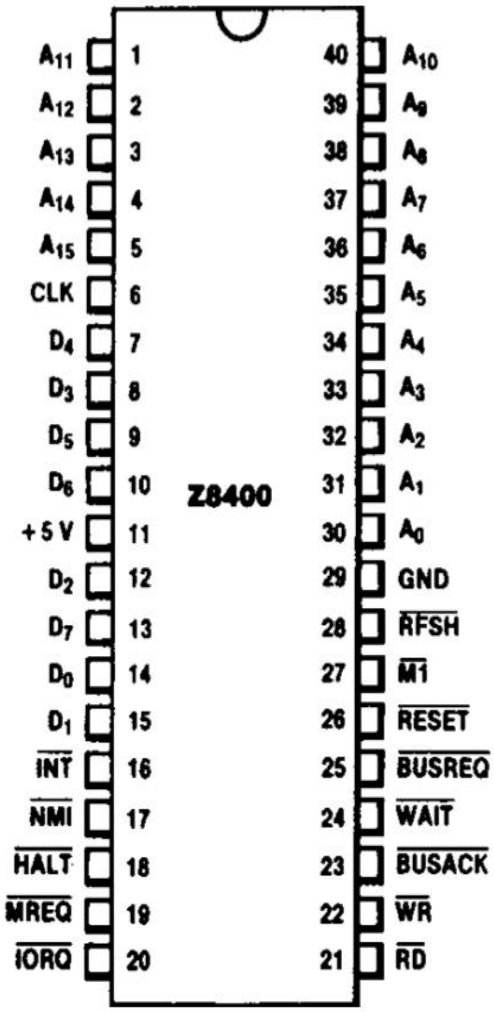
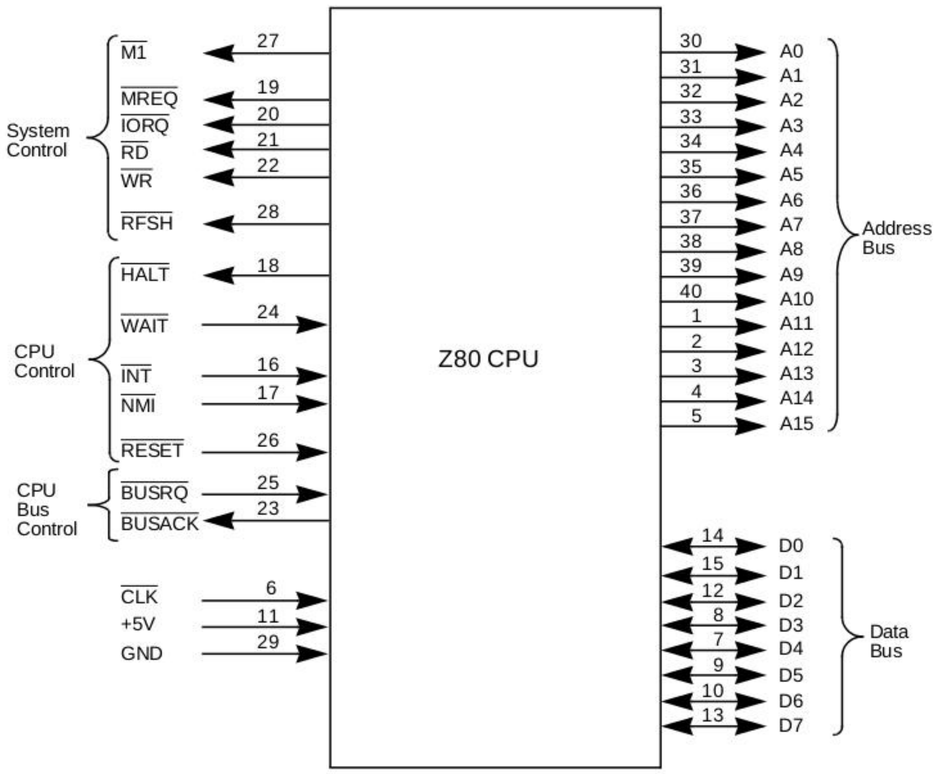


Fig 24-1



Z8400
A_{11} 1
A_{12} 2
A_{13} 3
A_{14} 4
A_{15} 5
CLK 6
D_4 7
D_3 8
D_5 9
D_6 10
+5V 11
D_2 12
D_7 13
D_0 14
D_1 15
INT 16
NMI 17
HALT 18
MREQ 19
IORQ 20
40 A_{10}
39 A_9
38 A_8
37 A_7
36 A_6
35 A_5
34 A_4
33 A_3
32 A_2
31 A_1
30 A_0
29 GND
28 RFSH
27 M1
26 RESET
25 BUSREQ
24 WAIT
23 BUSACK
22 WR
21 RD



Z80 CPU
System Control
M1 27
MREQ 19
IORQ 20
RD 21
WR 22
RFSH 28
CPU Control
HALT 18
WAIT 24
INT 16
NMI 17
RESET 26
CPU Bus Control
BUSRQ 25
BUSACK 23
CLK 6
+5V 11
GND 29
Address Bus
30 A0
31 A1
32 A2
33 A3
34 A4
35 A5
36 A6
37 A7
38 A8
39 A9
40 A10
1 A11
2 A12
3 A13
4 A14
5 A15
Data Bus
14 D0
15 D1
12 D2
8 D3
7 D4
9 D5
10 D6
13 D7


The
Right
Stuff



TRS8BIT

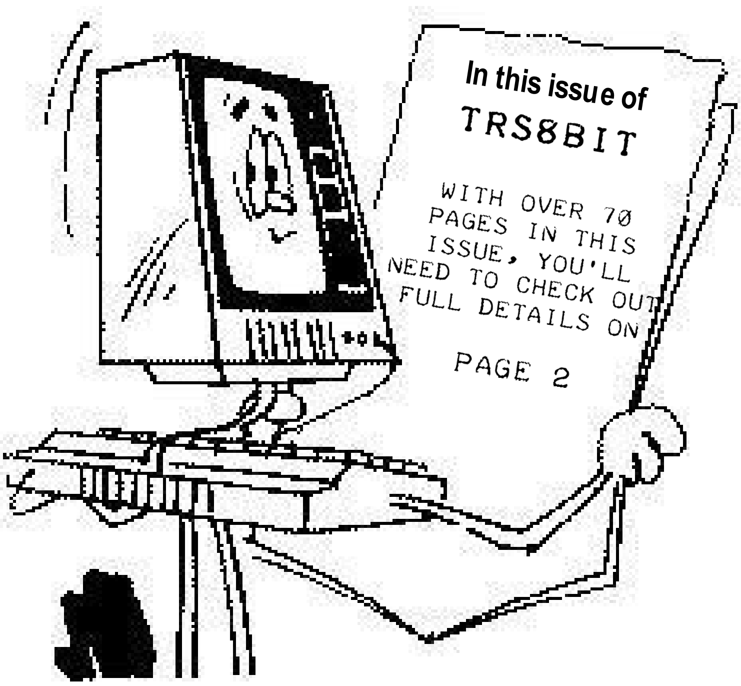
HELLO EVERYONE AND WELCOME TO THE SEPTEMBER EDITION.

THE FIRST THING I WANT TO MENTION IS THE 8-BIT COMPUTER 'EVENT' ON

SATURDAY 1ST NOVEMBER 2014

WHICH IS BEING HELD AT MY LOCAL LIBRARY, IN LOUTH, LINCOLNSHIRE.
FULL DETAILS ARE PUBLISHED HERE AND ARE ALSO ON THE WEBSITE. I DO HOPE THAT SOME (ANY) OF YOU CAN MANAGE TO MAKE THE JOURNEY, IT WOULD BE SO NICE TO MEET FACE TO FACE. IF YOU NEED DETAILS OF TRAVEL OR ACCOMMODATION PLEASE EMAIL ME AT THE USUAL ADDRESS.

DID ANYONE ELSE SEE THE 'RAIR DECWRITER 2' FOR SALE ON EBAY RECENTLY. DID YOU BUY IT AND HAVE YOU MANAGED TO GET IT UP AND RUNNING? WE WOULD ALL LIKE TO HEAR ABOUT IT IF YOU DID. WITH ITS SERIAL INTERFACE, IT WOULD CONNECT EASILY TO A M1 16K L2!!



I'M NOT GOING TO 'RABBIT ON' TOO MUCH, BECAUSE, WITH OVER 70 PAGES OF ARTICLES IN THIS ISSUE, I KNOW YOU JUST CAN'T WAIT TO GET READING.
HOWEVER, I'D LIKE TO THANK EVERYONE WHO SENT IN ARTICLES AND TO GIVE SPECIAL MENTION TO MAV'S ARTICLE ON AUTO BOOTING WITH THE FREHD AND HANS'S ARTICLE ON FIXING A 35 YEAR -OLD BUG ON THE M2. BOTH TRULY AMAZING.

LARRY KREAMER SHOWS US HOW HE WENT ABOUT SORTING OUT HIS M4 WHICH JUST DIED ON HIM.

KEVIN PARKER EXPLAINS JUST HOW BAKING CREATED PROBLEMS FOR HIS UNIQUE M16

I'VE ALSO FOUND A COUPLE OF ARTICLES FROM THE 1980'S WHICH I HOPE YOU ENJOY.

I WOULD ALSO LIKE TO REMIND YOU ALL OF THE **DR WHO COMPETITION**, FULL DETAILS ON PAGE 69
AND
THE TRAVELLING FREHD IS STILL AVAILABLE FOR A FREE 30-DAY TRIAL. PLEASE EMAIL ME AT THE USUAL ADDRESS FOR DETAILS.

BYE FOR NOW

DUSTY

CONTENTS


8 - BIT
COMPUTER EVENT
Saturday 1st November 2014
10:30 am - 3:00 pm
Louth Library-Conoco Room
Northgate, Louth, Lincs. LN11 0LY
FREE ADMISSION
See and use some of the early micro-computer hardware and software, that started the information technology revolution, which has taken us from Space Invaders to Call of Duty and bulletin-boards to the World Wide Web.
SPONSORED BY WWW.TRS-80.ORG.UK

Exchanging my model 4P keyboard.

Hans Reitveld

A year ago I purchased a model 4P which was in a very poor condition. There was also an additional problem because, as it was a French M4P, it came with an AZERTY keyboard layout.

If you are not familiar with this type of keyboard it is very different to the usual QWERTY one and can be quite aukward to use if you're not use to it.

After rebuilding the computer and adding all kind of options, such as extra RAM, hard disk, extra floppy etc., I then had to solve the keyboard problem.

Loading a U.S. Version of LS-DOS solved part of the problem. The keys were at the right location now, but the key caps were showing the wrong characters.

I was looking for a used keyboard for weeks without any luck so I decided to use special keyboard stickers to do the job.
The result was better than I expected, see picture AZERTY1, but some problems remained.
The punctuation characters were moved and the numeric keys on the right part of the keyboard did not work.

At last Ian (Mav) came to the rescue with a nice QWERTY keyboard, which he sold to me and after a few weeks waiting for the post, I could, at last, get to work and start to exchange it.

After opening the computer and disconnecting the cable, there was a problem! The old one had 20 wires and the new one only sixteen.

I first a mailed Ian, but he had never heard of this problem. So I decided to open both keyboards and try to locate the wires and find out what their function were. (see AZERTY3)

After following pin 1 up to 8 on both keyboards, both keyboards ended up on the same rows of keys.

Now the wires 9 up to 16. On the QWERY keyboard they went to the key columns, forming a matrix to detect each different key press.

On the AZERTY keyboard they went to the output of a set of two hex buffer chips.

The input of each buffer went to the same column as the wires on the the AZERTY keyboard.
On the keyboard, there are two more chip performing timing and debounce logic. These chips are powered by the thick cable on wire 17 and 19. The Wires 18 and 20 are not connected.

By comparing the signals of both keyboards, I could see that the signals and levels on both keyboards looked the same, so I then decided to swap them.

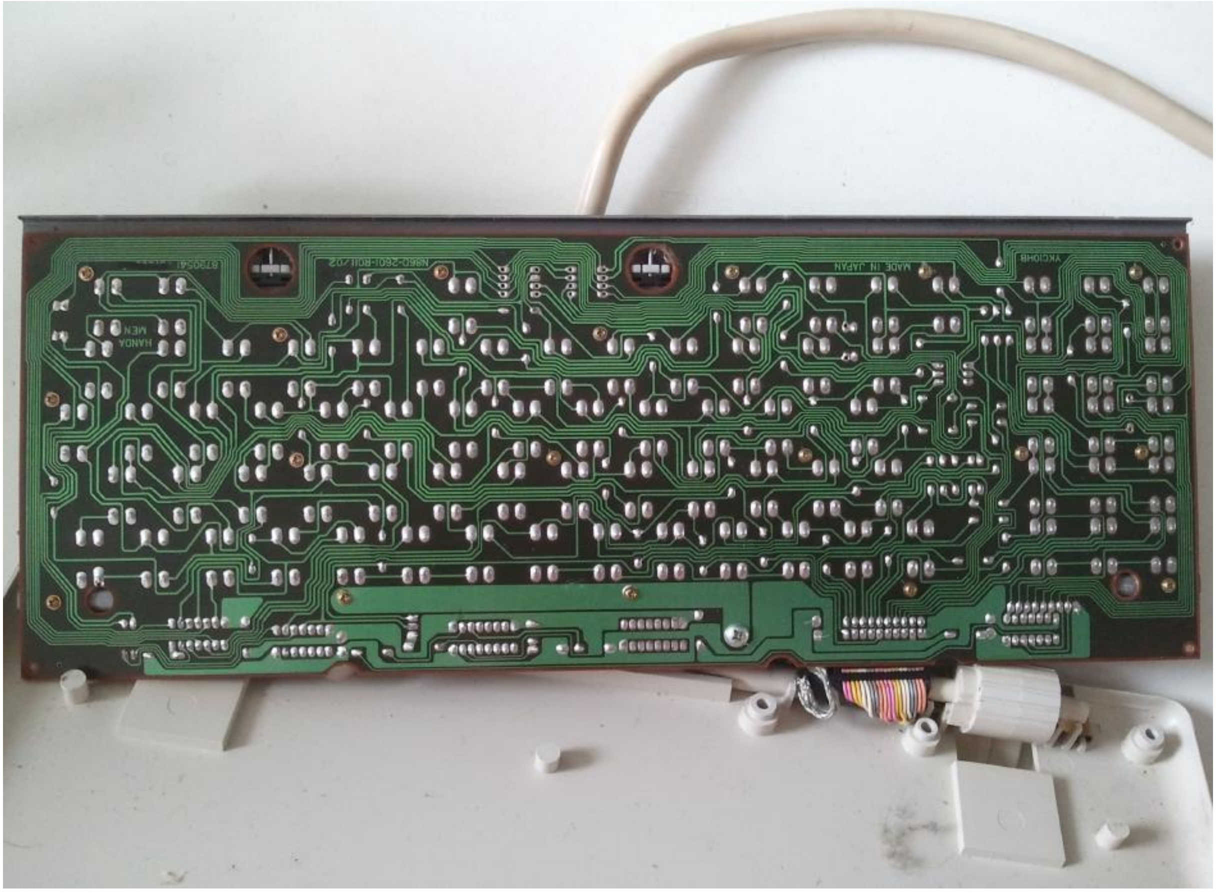
It was not possible to use the 16 wire cable on the motherboard without damaging the pins of the connector, therefore I used the 20 wire cable on the keyboard. I fitted the cable on the connector by using the first 16 holes.

Restarting the computer and testing all the keys showed me that the swap was very successful.

If you met this problem and want some help, please mail me.

Hans.

AZERTY 1



AZERTY 2

AZERTY 3

QWERTY 1

CUTTING DOWN ON GARBAGE.

HENRY SHEPHERD

THERE WAS A PROGRAM WAS PUBLISHED IN BYTE A WHILE AGO TO PRINT A LETTER 'C' ON THE SCREEN DURING GARBAGE COLLECTION SO THAT YOU KNEW WHY THE COMPUTER HAD STOPPED. I TRIED THIS OUT AND ALTHOUGH IT WORKED, IT DOUBLED THE TIME THE GARBAGE COLLECTION TOOK, AS WELL AS SLOWING DOWN ALL STRING OPERATIONS; ON BALANCE I PREFERRED IT WITHOUT.

TWO WAYS EXIST OF PREVENTING GARBAGE ARISING IN THE FIRST PLACE, AND IF IT DOESN'T ARISE IT DOESN'T NEED COLLECTING. BOTH ARE ONLY AVAILABLE IN DISK BASIC, UNFORTUNATELY. THEY RELY ON NEVER USING AN EXPRESSION OF THE SORT A$= ...... AFTER THE FIRST TIME IN ANY PROGRAM.

FIRST RULE - WHEN YOU FIRST ALLOCATE A STRING VARIABLE MAKE IT AS LONG AS IT WILL EVER NEED TO BE.

SECOND RULE - IF IT IS WRITTEN IN THE PROGRAM TEXT WRITE IT IN TWO BITS WITH A + SIGN IN BETWEEN THUS:

```
10 A$=''THIS PROGRAM "+''IS MINE!"
```

ALTERNATIVELY, DEFINE IT AS A BLANK STRING THUS:

```
10 A$=STRING$('" ",63)
```

BY DOING THIS YOU ENSURE THAT IT WILL BE STORED IN THE RESERVED STRING AREA, RATHER THAN IN THE PROGRAM TEXT.

THIRD RULE - WHENEVER YOU NEED TO CHANGE THE CONTENT OF A$, DO IT EITHER USING MID$(A$,X,Y)= (NEW STRING), OR USING LSET A$= (NEW STRING).

THE SECOND METHOD IS MUCH THE EASIER AS IT DOESN'T REQUIRE YOU TO COUNT LETTERS TO FIND THE VALUES OF X & Y. WHICHEVER METHOD YOU USE, THE RESULT IS THE SAME, YOUR ORIGINAL STRING IS AMENDED WHERE IT EXISTS IN MEMORY, AND NO GARBAGE IS CREATED.

THE MAGAZINE DOESN'T MAKE IT CLEAR YOU CAN USE LSET FOR ORDINARY STRINGS, BUT YOU CAN.
LSET PADS STRINGS WITH BLANKS SO IF YOU ARE PRINTING THEM ON THE SCREEN OR A PRINTER YOU MAY HAVE TO RESORT TO TACTICS LIKE INSERTING MARKER CHARACTERS TO SHOW HOW MUCH OF THE STRING YOU WANT TO PRINT.

FOR EXAMPLE, IF YOU HAVE SET UP A$ INITIALLY TO BE 63 BLANKS, AND YOU WANT TO PRINT A SHORTER STRING THAN THIS, THEN LSET A$=B$ + C$ + CHR$(1)
WILL GIVE YOU A 63-BYTE STRING CONSISTING OF A$ AND B$ JOINED TOGETHER, FOLLOWED BY A NON-PRINTING BYTE (01 HEX), WITH THE REMAINDER OF THE 63 BYTES ALL BLANKS (20 HEX).

TO PRINT ONLY THE BUSINESS PART OF THIS, USE:

```
PRINT LEFT$(A$,INSTR(A$,CHR$(L)));
```

BECAUSE CHR$(1) IS A NONPRINTING CHARACTER IT DOESN'T MATTER WHETHER YOU PRINT IT OR NOT - IT WON'T AFFECT THE DISPLAY.

DON'T USE CHR$(Ø) AS IT MAY UPSET BASIC.

NEITHER METHOD WILL WORK FOR STRINGS IN PROGRAM TEXT, HENCE RULE TWO.
IT'S NOT ALWAYS POSSIBLE TO DO THINGS THIS WAY, BUT EVEN IF YOU ONLY PREVENT SOME GARBAGE COLLECTING, EVERY LITTLE HELPS.

(THIS ARTICLE FIRST APPEARED IN NATGUG NEWS, JAN '83. ED)

BAS IS THE GUY WHO HAS MADE SUCH A BRILLIANT JOB OF REPAIRING MY MODEL 1 PRINTER INTERFACE. HE'S U.K. BASED AND A TANDY ENTHUSIAST.
NOT ONLY IS HE 'LOCAL' HE'S VERY KNOWLEDGEABLE TOO.

RE-INKING

DUSTY

WHO CAN RESIST A BARGAIN ON EBAY? AS I'M SURE YOU ARE ALL AWARE, NOT ME! SO WHEN, A COUPLE OF MONTHS AGO, A 'WET-RIBBON' RE-INKER CAME UP FOR SALE, I JUST HAD TO HAVE A BID.

IT SEEMED TO BE MOST FORTUITOUS, BECAUSE I WAS ATTEMPTING TO WRITE A ONE-LINER BASIC PROGRAM TO PRINT A LARGE SIGN FOR MY GRANDSON'S BIRTHDAY AND I HAD FOUND THAT THE INK RIBBON ON MY OKI MICROLINE-8Ø WAS SO FAINT, IT WAS ALMOST ILLEGIBLE!

AS IT HAPPENED, THERE WASN'T MUCH INTEREST SHOWN IN THE AUCTION, SO I 'WON' THIS QUITE UNIQUE BIT OF KIT FOR THE FIRST AND ONLY BID OF £4.99! DEE WAS IMPRESSED (NOT)! SHE DOESN'T SEEM TO SHARE MY EXCITEMENT FOR THESE WONDERFUL EBAY TREASURERS!

I'D NEVER COME ACROSS THIS MACHINE BEFORE, IN FACT, I'D NOT EVEN HEARD OF IT, BUT WHEN IT DULY ARRIVED, I THOUGHT THE BEST THING TO DO WOULD BE TO CLEAN UP ANY RESIDUE OF OLD INK THAT REMAINED, BY SOAKING IN WARM SOAPY WATER.
I HAD 'FOUND' SOME SUITABLE INK WHICH I HOPED TO USE, EVEN THOUGHT IT WAS OLD (3Ø YEARS PLUS) AND PERHAPS PAST ITS BEST, BUT I HAS HAPPY TO GIVE IT A GO.

THE MANUAL THAT ACCOMPANIED THE MACHINE STATED THAT IT WILL RE-INK QUICKLY, CLEANLY AND EFFICIENTLY, MOST DOT MATRIX PRINTER (CONTINUOUS LOOP) CARTRIDGES.

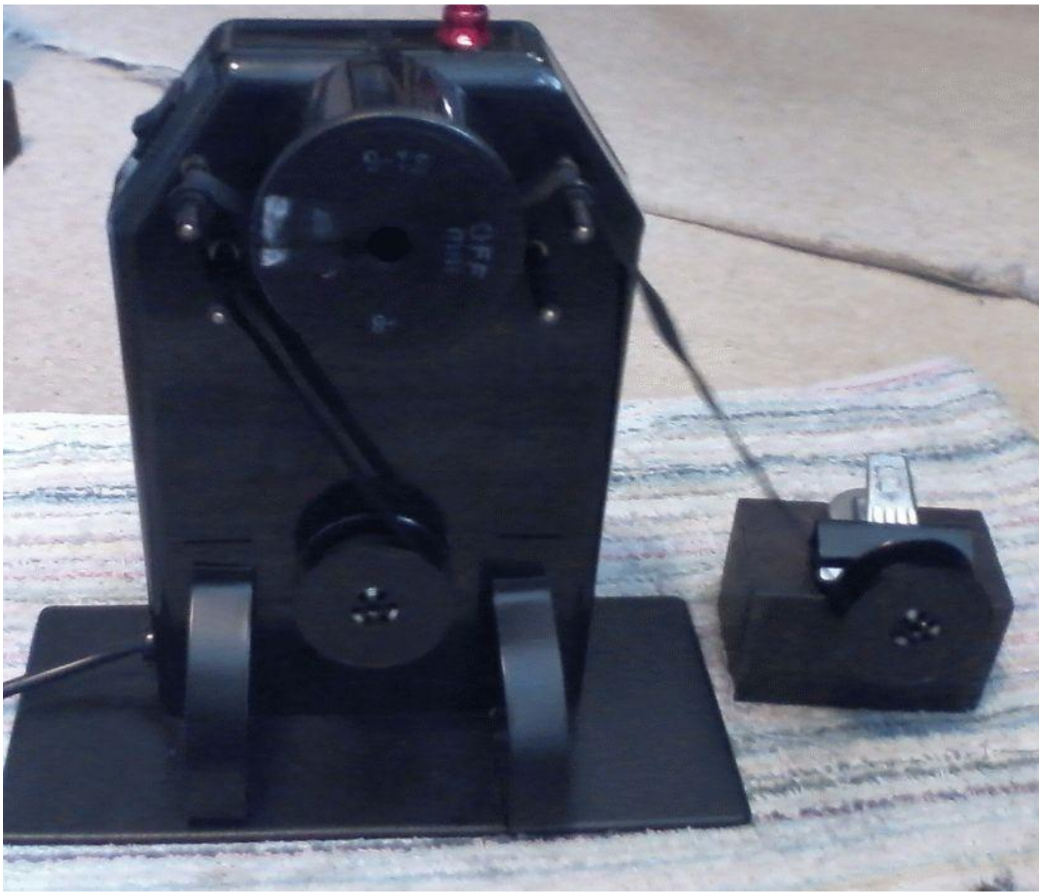
THERE ARE 3 SETTING TO ACCOMMODATE THE VARIOUS WIDTHS OF THE RIBBONS WHICH NEED RE-INKING, WHICH RANGE FROM 9-30 MM AND THEY ARE CHANGED BY TURNING THE WHEEL ON THE FRONT OF THE MACHINE. THIS ALSO HOUSES THE INK 'WELL'.
THE RIBBON IS PASSED ACROSS THE OUTER SURFACE OF THE INKWELL AT ITS LOWEST POINT. THE INK IS ABLE TO PASS TO THE RIBBON THROUGH NOZZLE HOLES, WHERE IT IS ABSORBED INTO THE RIBBON FABRIC.
AN ELECTRIC MOTOR, WITHIN THE CASE, DRIVES, BACKWARDS OR FORWARDS, A METAL OUTPUT DRIVE-SHAFT, ROTATING AT 25 RPM WHICH TURNS THE RIBBON WITHIN THE CARTRIDGE. THERE WERE VARIOUS ADAPTERS WHICH CAME WITH THE ORIGINAL PURCHASE.

WELL, AS YOU CAN MOST LIKELY GUESS, THE FIRST PROBLEM I ENCOUNTERED WAS THAT AFTER NOT BEING USED FOR 30ISH YEARS, THE INK LEFT IN THE INKWELL HAD SET LIKE CONCRETE CONVENIENTLY BLOCKING UP ALL THE NOZZLE HOLES TOO! ADDED TO THAT, THE FRONT OF THE 'WHEEL', DID NOT WANT TO FOLLOW INSTRUCTIONS AND TO QUOTE THE MANUAL "TO REMOVE THE WHOLE INKWELL FROM THE WET-RIBBON, FIRMLY HOLD THE CASE AND GRASPING THE FRONT OF THE WHEEL, PULL IT FIRMLY FORWARDS".
IN THE END, I HAD TO REMOVE THE BACK AND START PUSHING!

A COMBINATION OF DRIED INK AND RATHER STIFF PLASTIC SEEMED TO BE THE MAIN PROBLEM. HOWEVER, WITH A LITTLE, GENTLE PRESSURE FROM THE BACK, OUT POPPED THE WHEEL AND I SET ABOUT CLEANING IT.

ONCE IT WAS CLEANED UP, I REPLACED THE WHEEL, CLOSED UP THE MACHINE AND REFILLED THE INKWELL.

THE SECOND PROBLEM, WHICH YOU MIGHT HAVE ALREADY GUESSED WAS THAT THE OKI ML80 DIDN'T HAVE A CARTRIDGE RIBBON, BUT 2 SPOOLS JUST LIKE A NORMAL TYPEWRITER!



SO, WHAT TO DO!

I DECIDED THAT IF I COULD DEVISE A METHOD OF SPOOLING THE RIBBON ONE-WAY, BY PLACING THE EMPTY SPOOL ON THE DRIVE SHAFT, I COULD JUST RUN THE PROCEDURE, SWAP THE SPOOLS OVER WHEN COMPLETED AND KEEP RUNNING THE PROCESS AGAIN.
SO WITH THE HELP OF A HEAVY BLOCK OF METAL, A MAGNET ATTACHED TO A 'BULLDOG' CLIP AND A MODIFIED PAPER CLIP TO HOLD THE FULL SPOOL, I SET THE SYSTEM RUNNING.

TO BE FAIR, THE MACHINE PERFORMED FAULTLESSLY AND ALLOWING FOR THE AGE OF THE INK, I WAS VERY PLEASE WITH THE OUTCOME. I ONLY RAN THE INKING PROCESS 3 TIMES, AS IT MEANT WAITING FOR THE TAKE UP SPOOL TO BE FULL AND THEN SWAPPING THE SPOOLS AROUND.

WITH A CARTRIDGE, THE MANUAL RECOMMENDS A 45 MINUTES RUNNING TIME WHEN RE-INKING. AS A CARTRIDGE RIBBON IS A CONTINUOUS LOOP, IT WILL KEEP PASSING UNDER THE INKING NOZZLES, SO YOU CAN LEAVE IT TO GET ON WITH IT, BUT I RECOMMEND KEEPING AN EYE ON THE PROCESS. I WOULD ALSO RECOMMEND THAT YOU WEAR A PAIR OF PLASTIC GLOVES, AS THE INK HAS THIS KNACK OF GETTING EVERYWHERE, AND WHATEVER YOU DO, STAND THE WHOLE SETUP ON AN OLD CLOTH OR TOWEL.

HERE'S A FIRM I CAME ACROSS ON THE NET. THEY WERE VERY QUICK WITH MY ORDER AND HAVE A LARGE SELECTION OF CLASSIC PRINTER RIBBONS & CASSETTES, BUT NOT, I REGRET, FOR THE LP_VII THOUGH :(

Revival of aTRS-80 Model II

Hans Reitveld

Initial age related problem(s)

A month ago I purchased a Model II on an internet market site. It had not been used for a number of years and had been stored in a garden shed for some considerable time.
When I visited to the seller and took a look at the computer, it was covered in dust and, unfortunately, there was no software with it. The seller told me that it was working OK and to proved it to me, he turned it on for a short while. There was the message "Insert Disk" on the screen, so I decided to buy the computer and took it home.

The first thing I did was to open the computer and clean everything that I could reach. I put it on a table and turned it on. It ran fine for about ten minutes and then a flash and some flames! The power supply was burning. I quickly turned off the power and put out the flames. It seemed that one of the filter capacitors was blown. I could reach the capacitor by loosening the upper of three mounting nuts and bolts and placing a' spacer' between the PCB and the metal frame, which gave me just enough room to replace the blown and a second capacitor.

I turned the computer on again and went searching on the internet for an operating system. After a short time I noticed there was a burning smell coming from the computer so I decided to start looking for the source! It turned out to be the keyboard connecter, which was burning hot. I turned the computer off and opened the keyboard connector and found that three wires had melted together and were still smoking! I replaced the keyboard cable and connector, but, after that, the computer would not power up. There was nothing, only the horrible sound of just the fan running. No lights or any disk activity whatsoever. It seemed that the power supply was down. No 5 volts anymore.

I could not find an obvious reason why, so I decided to take out the power supply, a difficult and time consuming job as the video controller and the power supply are build together on one support and you need to take them out together. However, this allowed me to see the problem. The power supply is protected against overloading by dumping part of the current into a resistor, this in turn, reduces the load. If this situation takes too long, the fuse is blown. After replacing the fuse and refitting the power supply, I booted the computer again, but found that I now had no keyboard access.

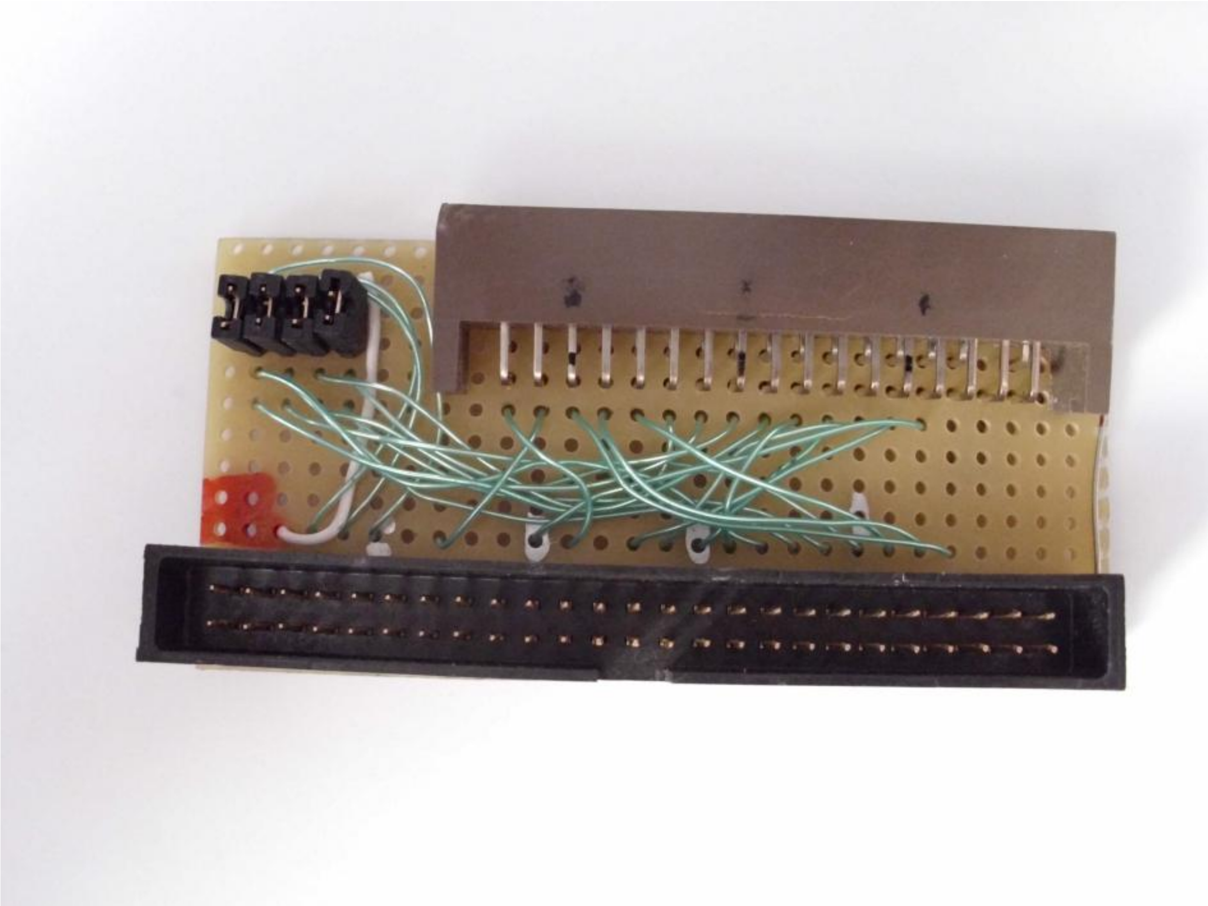
Opening the keyboard didn't show any obvious problems only that the keyboard plug was not fixed to the cover. This was quickly repaired with some hot glue. I checked the signals with the 'scope' and found that this showed all the lines were dead. I noticed that the 'busy' was low, (forcing this line high returns the keyboard activity). Tracing back the busy line took me to a 74LS74. I replaced this chip but this didn't help.

The problem(s) were turning out to be more serious than I had first hoped. I also found that the CTC controller on the CPU board was partly gone, however, I managed to source a new one in Germany and with this replaced, I was beginning to think that all the major problems had been solved.

The software on the internet was very sparse, but I managed to find an operating system, but only in an IMD format. I was lucky in that my HxC floppy emulator could read this format. Now I needed to build a converter to connect my emulator to the external floppy bus. This was not an easy job, but I succeeded by using a small PCB with two connectors and a lot of wire-wrap wire.
There was still a problem with the termination so I added the termination on the PCB with four jumpers to turn it on or off, depending on what was connected on the bus.

I'm very happy with my Model II, but all the problems I have encountered were much more than I initially had expected.

Below is a picture off my 'new' Model II and the interface PCB.
If someone is interested in further details of my solutions, please contact me.



Video display problem(s)

I noticed that every time I accessed the disk drive, the screen grew and shrunk. This only happened when there is a 'head-load'. When there is a load, the screen shrunk and when the head is released the screen grew.

When I first encountered this video display problem, I put in a request on the TRS-80 forum to see if anyone had come across this before.

I wondered if this is the normal state of affairs of the Model II or is it a problem that requires me to look for a solution ? The drive is a Shugart, full height, single sided 8" drive. The +12 Volt is very steady and is unaffected during disk access. The +24 Volt is normal and only drops about 0.9 volt during head-load. The high tension cables are free and clean. If I select an emulated disk on my HxC floppy emulator the display is steady. The video-board is a 8709048 rev. C

After receiving some replies, I made the following conclusions.

I was quite confident that it was not the power supply, or the magnetic field from the disk drive.
I monitored the output values of the power supply and I didn't see any

unexpected changes.
I also placed a metal sheet between the disk drive and the monitor and there was no change in the behaviour of the monitor.

I began to think that the solution was in the hardware of the video board. There is a lot of logic to disable video changes during some hardware actions and it could be that causing problems in those circuits. After checking this out, I came up with some interesting ideas.

I added some wires between the head-load solenoid and put in a switch, allowing me to turn off the head-load temporarily. Then, after booting, I did a disc access without the head-load activated and the screen shrunk just the same, so that confirmed is was not the solenoid.

If I attach my floppy emulator to the extension port, select it as drive 0 and disable the internal floppy drive there is no shrinking. Therefore I thought it was safe to assume that everything pointed to a software/ hardware problem. Making an educated guess, I assumed that during disk access, the 6845 chip becomes unstable which was causing the problem.

I then found a Tandy Technical Bulletin, Number 23, which related to a missing trace on the video PCB. After making this repair, although the shrinking doesn't disappear, some glitches on my screen are gone.

I have removed the extra card on my pc, but it didn't make any difference. The explanation about dropping the voltage on the power supply doesn't help to explain the source of the problem.
The monitor display is only connected to the device through the + 12 Volt which I have monitored during disc access, but this looked stable.

The only other likely possibility is the clock generator on the video card. This is 5 volt logic and could suffer from power drops during disc access. It would be difficult to measure this circuit and would need a bus extender making, in order to do that. If the voltage of the oscillator changes, the frequency will change and this would affect both the horizontal and vertical frequency which would change the screen format. I decided to keep on searching.

Then an even better idea came to me. Powering the monitor, only, from an external stabilised 12 volt power supply. There is a connector just above the power supply where the 12 volt is fed to the monitor part. It would be easy to connect an external power supply there. A second solution could be to drain the 12 volt from the 24 volt source by means of a 12 volt integrated regulator (7812).

I checked the specification of the monitor and the 24 volt supply and I monitored the 12 volt supply again and found that the lost during disk access is maximal of 0.4 volt.

It seemed that my second solution was not possible. According to the spec. the 24 volt can supply 1.7 amps and the 12 volt may not be left open. There must be a resistor connected to the 12 volt, of about 22 ohms, to keep some current flowing. The monitor part needs 1.2 amps so I'm afraid that there is not enough power left for the disk drive.

The first option with a small 12 volt stabilised power supply which fits in the computer is the better option. I quickly sorted out a 12v supply and as I suspected, the internal power supply was the source of the problems.

I easily modified the connector on the video board. This is the one that is built together with

5	Volts	@	8.6	Amps
12	Volts	@	4.5	Amps
24	Volts	@	1.7	Amps
-12	Volts	@	0.2	Amps

The power supply rectifies the AC line to DC, chops it at 20 kHz, then transforms the chopped DC to the required output voltages and finally rectifies the transformed output to low voltage isolated DC. Feedback loops are provided for voltage regulation and over current protection.

The power supply may be jumper selected for either 95 to 135 VAC or 190 to 270 VAC. It will operate at either 50Hz or 60Hz input frequency.

CAUTION

This power supply must have a load present, i.e., the computer and CRT, or damaging oscillations may result.

VIDEO MONITOR
(MOTOROLA)

A. FUNCTIONAL SPECIFICATIONS

General Information

All models are direct drive, requiring separate TTL vertical/horizontal drive and video inputs. All use 12" CRTs of the magnetic deflection type with integral implosion protection and require a power input of +12 volts@ 1.2 amps.

Input and output connections for the monitor are made through a 10-pin circuit card edge connector. The inputs are video, horizontal drive, vertical drive, system ground, and +12 volts.

A single circuit card with components mounted on one side is used. Schematic reference numbers are printed on both sides of the circuit card to aid in the location and identification of components for servicing.

Circuitry consists of one stage of video amplification, two stages of horizontal deflection processing, and three stages of vertical deflection processing.

SPECIFICATIONS

ITEM	SPECIFICATION
Cathode Ray Tube:	12" measured diagonally (305 mm); 74.86 square inches (483 sq. cm); 90° deflection angle; integral implosion protection, P4 phosphor standard.
Power Input:	+12 VDC at 1.2 amps typical, 1.5 amps maximum.

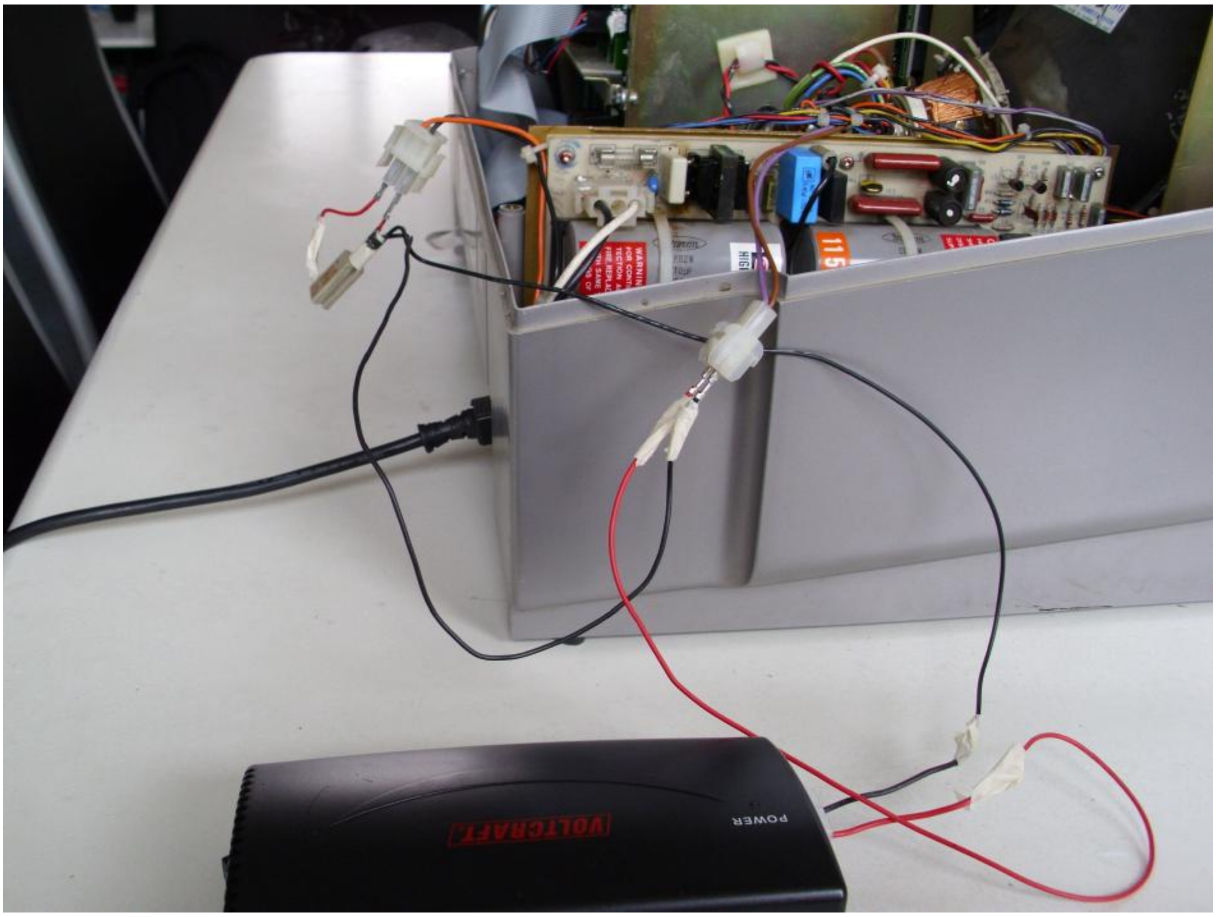
the power supply. After the modification the display is absolutely steady during disk access AND all other actions, it is now a pleasure to look at the screen.

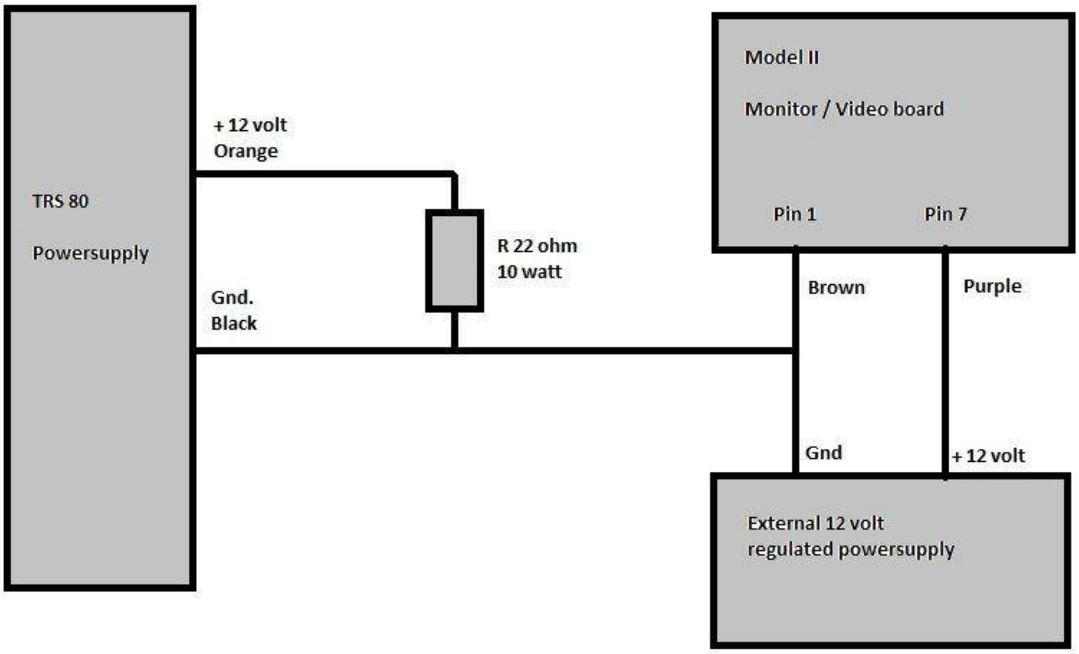
However, a word of caution ...
If you wish to make this modification please be very careful. Remove the mains supply cord and pay extra attention to very high voltage that remains on the picture tube. Also, the neck of the picture tube is very fragile, and is easily broken.

If you take a look at the picture, you see a connector is placed on the top of the power supply. On this connector you can make the modification. It has a black, orange, brown and purple wire.
You need an external regulated 12 volt power supply, which needs to be fitted anywhere in the computer case. I found one on Ebay, but I have not received it yet, so I don't know if it is usable.
But the specs are good and the size is perfect. Take a look at the picture.

On the schematic you can see how I connected the wires, but again, please pay special attention to the resistor of 22 ohm. This will get very hot so find a safe place to mount it.

I hope you have a better Model II after this modification.





Why an extra load on the unused 12 volt line?
The Model II supply is a switching power supply. To make that work satisfactorily, the load on all the outputs must be balanced. In the service manual there is shown a minimal load value for each voltage mentioned. If one of the loads is too low, other voltages could start drifting up and down.
As the video system is the only load on the 12 volt, I had to compensate this load if I disconnected the video system. The resistor gets very hot with a dissipation of about 8 watt, so it is advisable to mount it on a metal part of the frame.

The video board is a self supporting unit in the Model II and gets its only ground by this connector, which I had to open. So I need a new ground for the sync signals of the video board.

I forgot to mention the specs of the external supply I found on Ebay, here they are.

AC 110V/220V ± 20%
47/63Hz
12V
2A
(L*W*H)86*58*38mm
Metal case

Universal AC input / full range
It's good quality & high performance
100% full load burn-in test
Protections: overload/ over voltage/ short circuit
Cooling by free air convection

This leaves me with one last irritating problem. The awful sound of the fan (blower).
I have bought a new 220 volt fan but it seems that they all make this awful noise. After my power supply modification, there is a now a new option. As the 12 volt rail is now empty and loaded with only a dummy load, there must be 12 volt fans of this size and with ceramic or sleeve bearings.
The question was, that I was unsure if they supply enough air to cool the system. This would be a very neat solution.

A new load for the 12 volt and no more noise anymore!

This could be a useful replacement blower.
[http://azerty.nl/8-1088-97019/sharko...entilator.html]

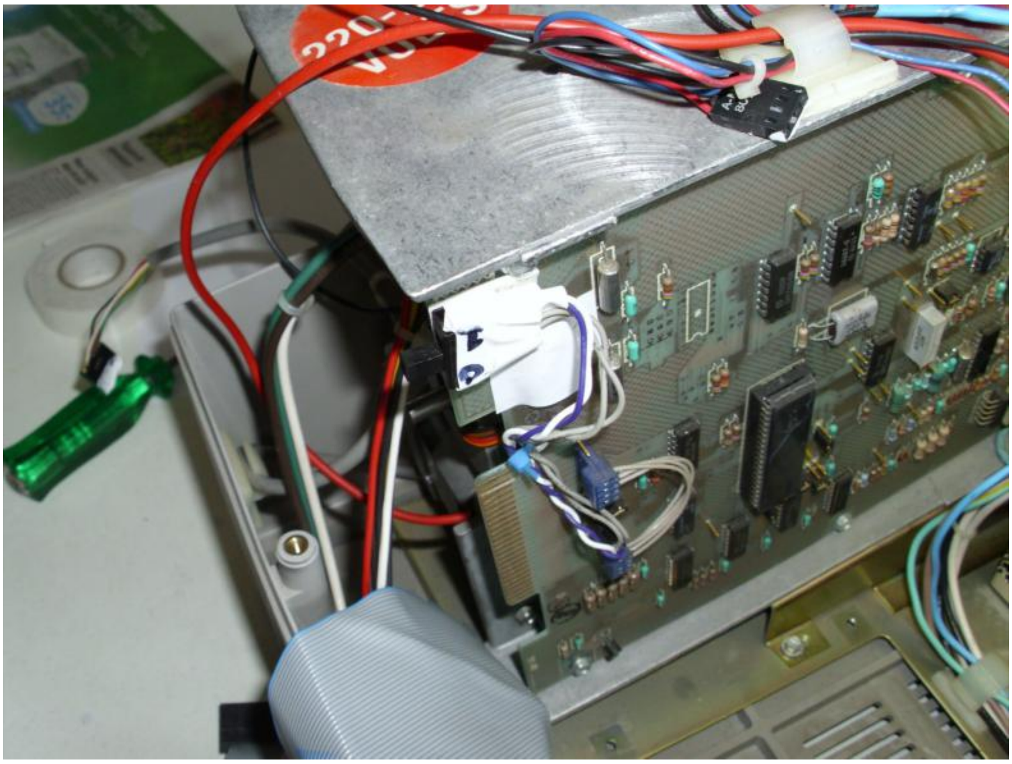
The finishing touch.

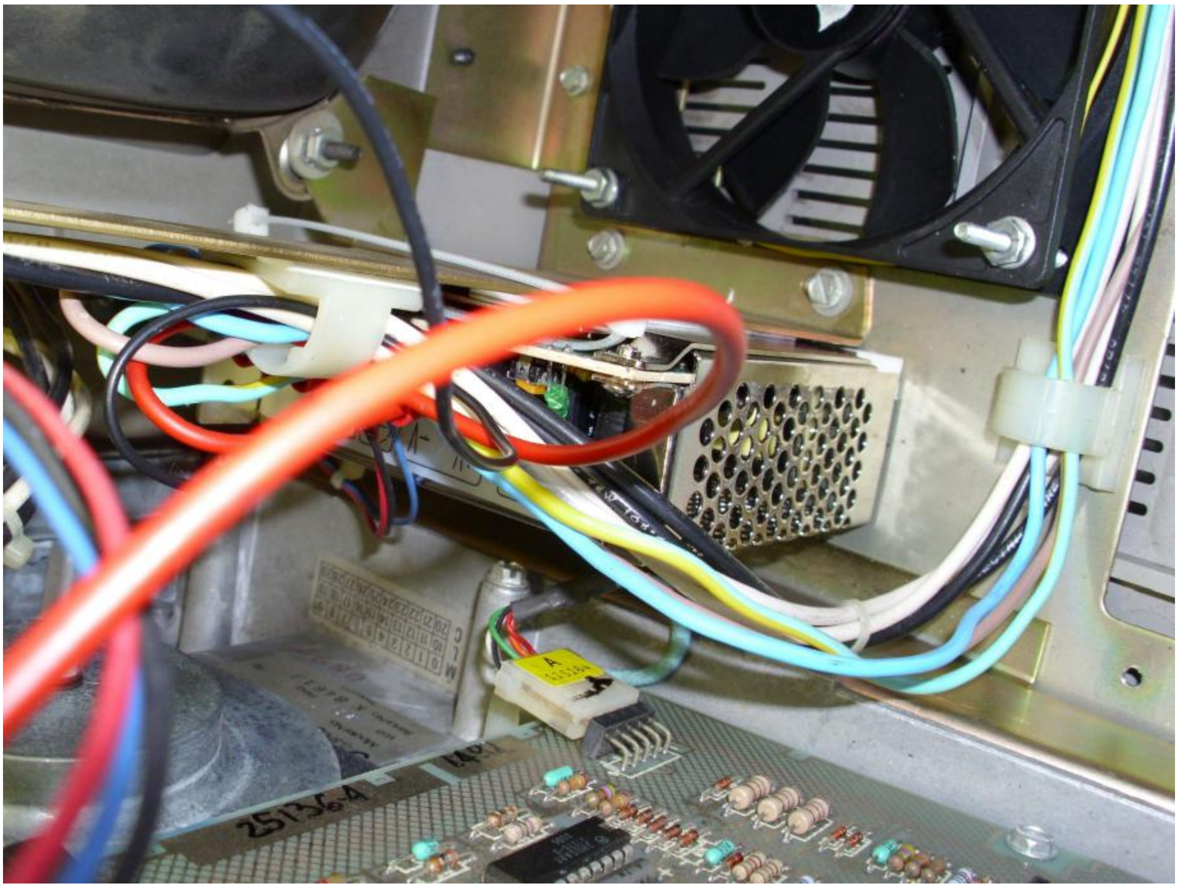
I found the perfect place for the power supply was just beside the floppy drive. I attached it using two sided tape and a tie wrap. The result was, as expected, no distortion of the video anymore.

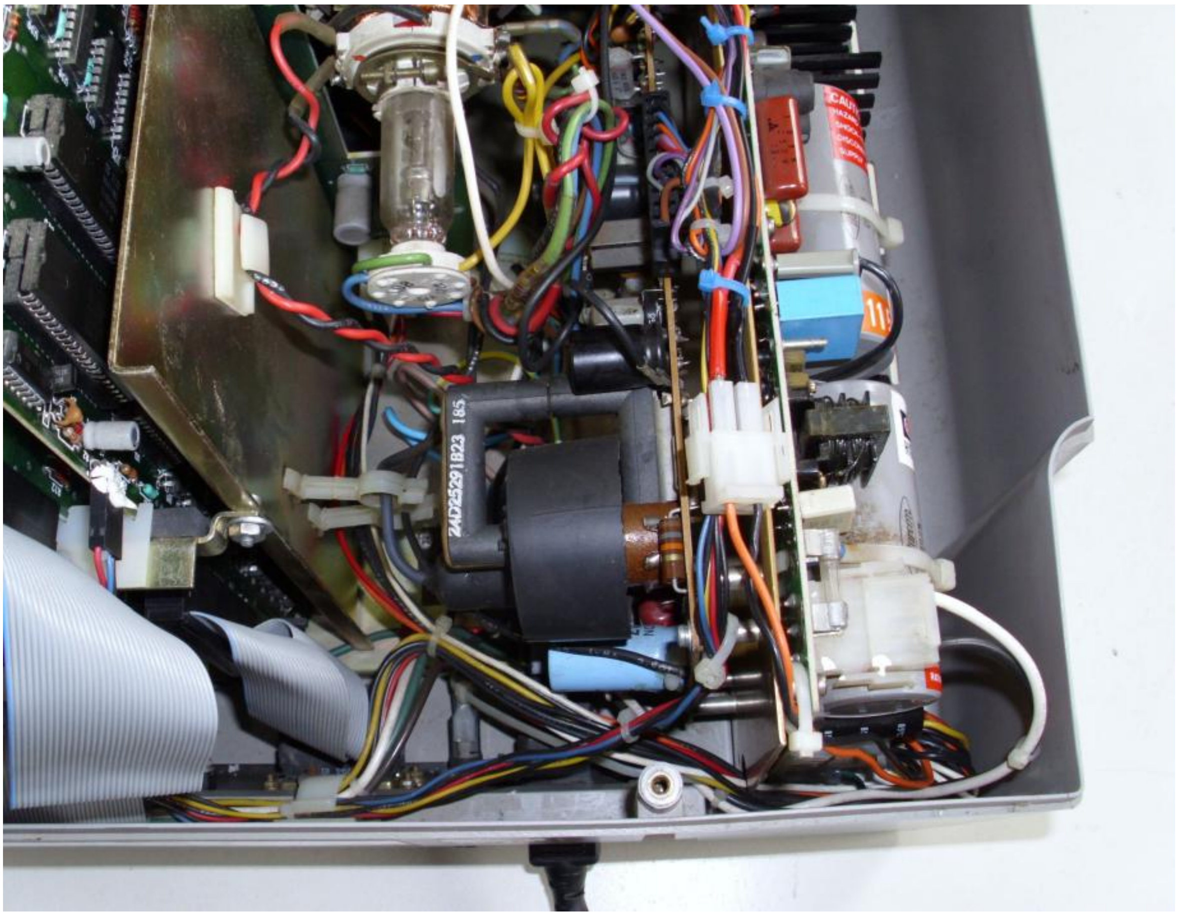
It was easy to modify the drive select 0 and 1, by adding a switch on the select jumpers. It is easier now to change the drive when using my HxC emulator.

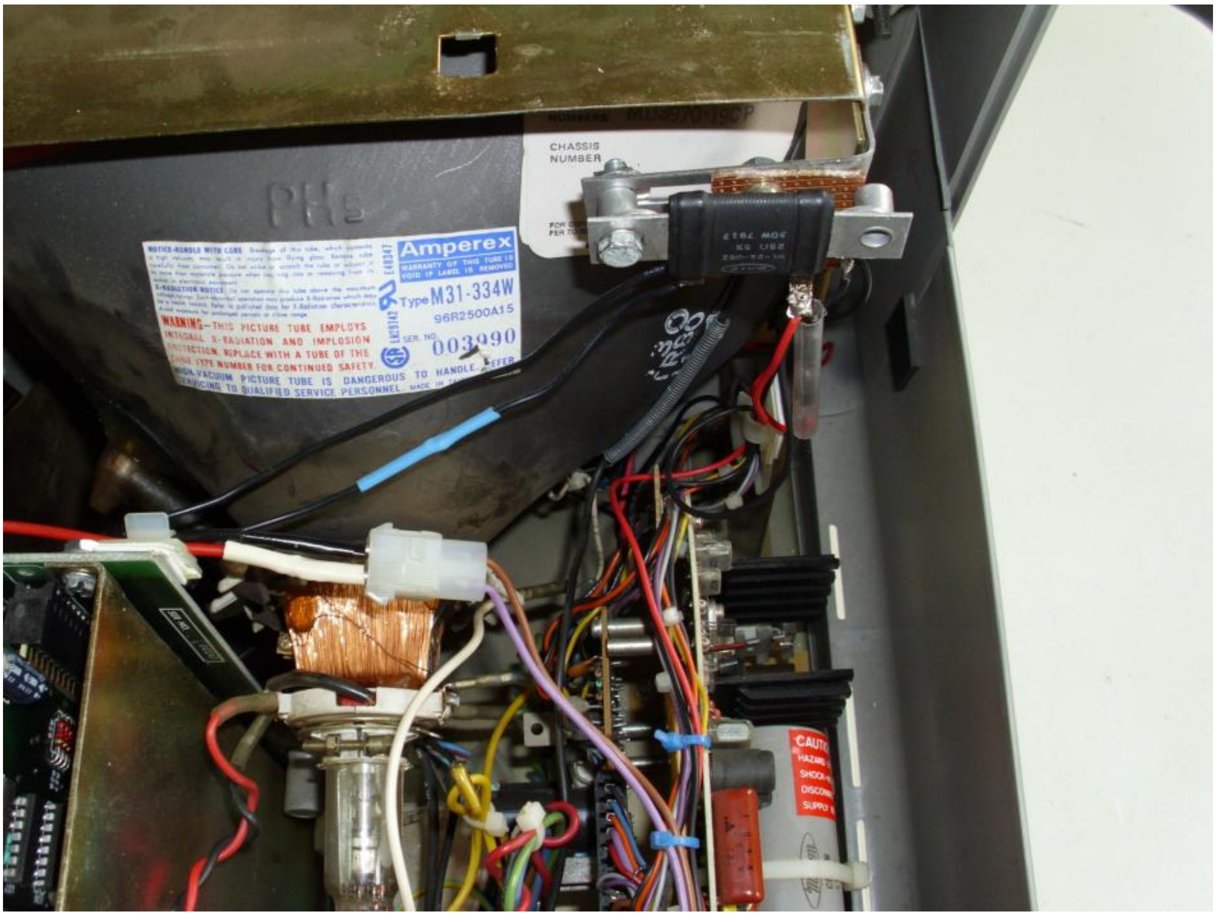
I could change the blower with the 12 volt version, without removing the outer case.

Just by fixing the some spare screws to the bottom of the inside of the computer case. The sound of the fan is now reduced by half. Some of the sound is produced by the motor of the disk drive and another part is produced by the airflow and the vibrations of the case. Lowering the speed of the fan by adding 3 diodes in the supply line makes the sound tolerable.
Just take a look at the pictures for the details.









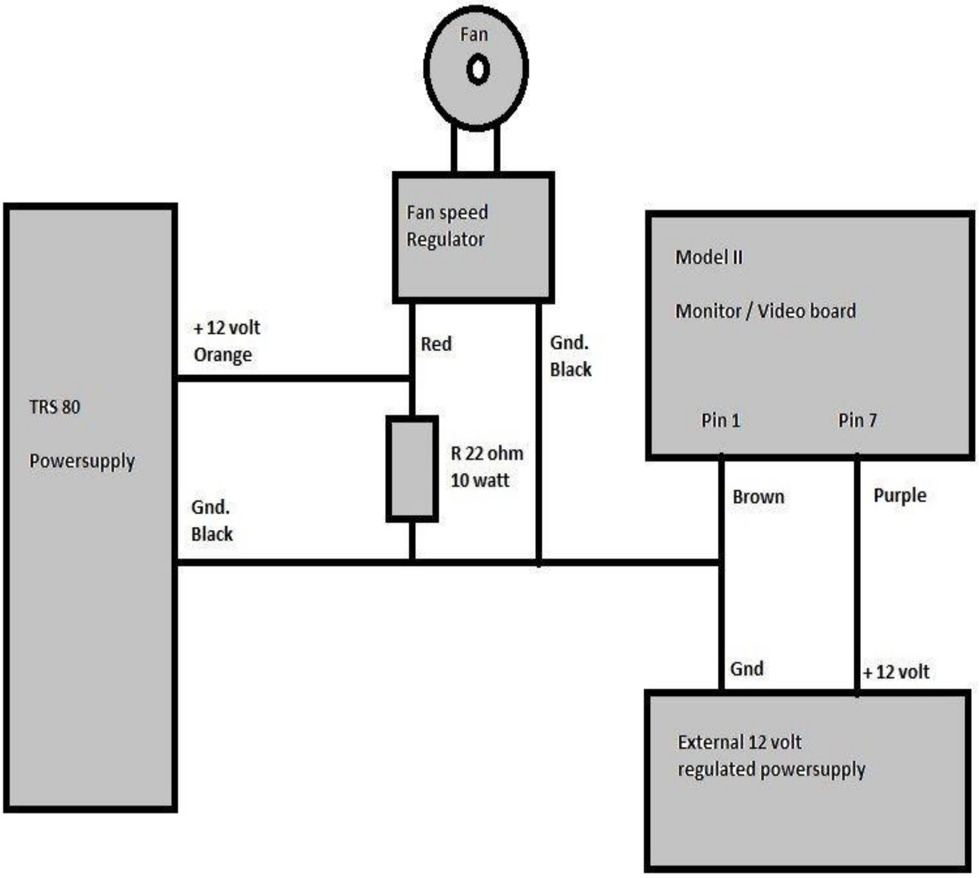
Some remarks about the fan.

To reduce the vibrations you can, according to the service manual do the following.

Tighten the four nuts from the fan down and wait a while, then turn them one by one, about 4 turns back, then turn them on with your fingers. Optionally, you can mount small rubber o-rings on the screws. One between the fan and the bottom and a second one on the fan, beneath the washer and the nut.
In order to reduce the speed, you can put some diodes in series with the plus wire. A better idea is to place a speed regulator between the fan and the power supply. They are easy to find on Ebay, eg. Zallman. It is possible to place the speed adjust control between the video controls on the front.

For those who want to do some modification on the Model II without building hardware, here are some old links for handy adapters.

For mixing 8", 5.25" and 3.5" drives in one system.
[http://www.dbit.com/fdadap.html]



To generate the needed voltages for an 8" drive in a PC computer, by using the default power supply.
[http://www.dbit.com/fddc.html]

These have not seen or tested, this is just for information only.

Further reducing the fan noise and help with cooling.

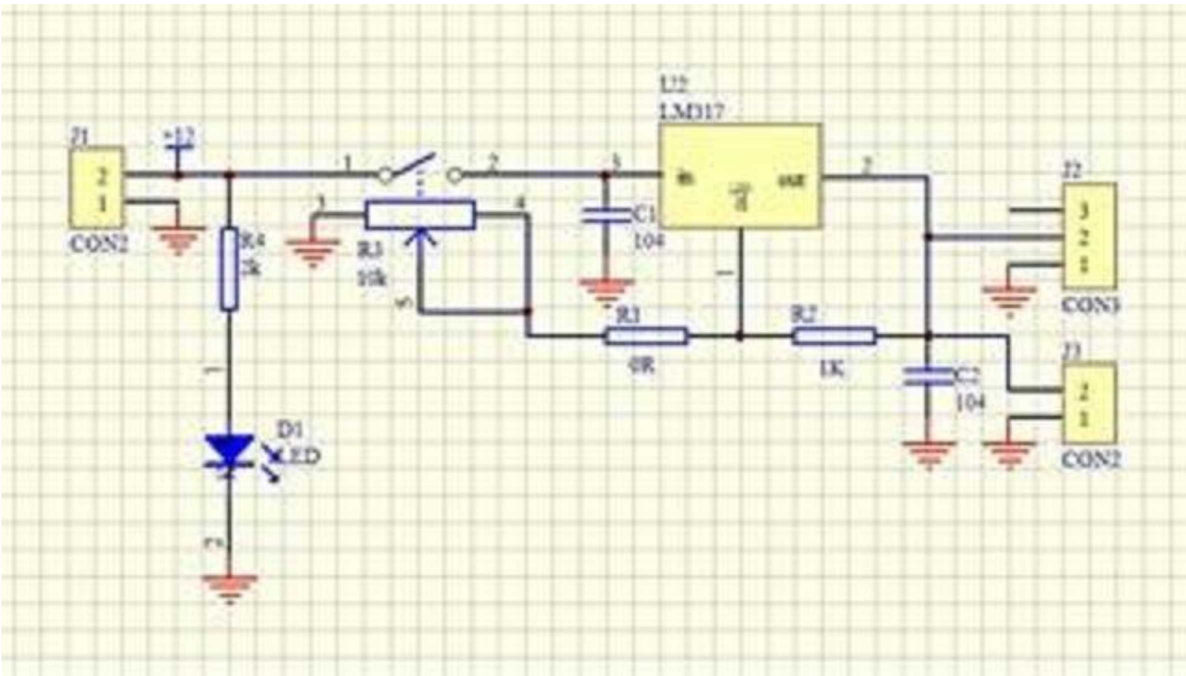
Here, I've added a list of additional modifications I have carried out to further reduce the noise of the fan.

I have added 2 rubber o-rings above and below the fan, two o-rings on each bolt.
I have replaced the rubber feet at the corners of the case with thicker and softer pads
The head spindle of the disk drive needed a few drops of oil.

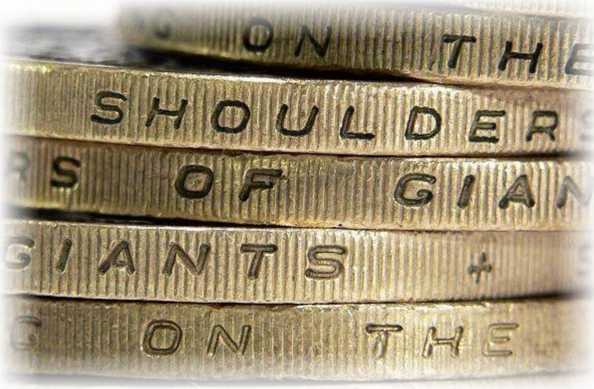
I also managed to find a perfect fan speed regulator at
http://stores.ebay.com/chanouny01?_t...p2047675.12563

It is very easy to move the pot-meter to the front of the case

De-solder the potmeter from the PCB at points 1,2,3,4,5. If you don't need a switch to turn the fan off and on, just bridge the points 1 & 2 with a jumper. Connect the middle and one of the outer points of the potmeter together and add a wire. Connect this wire to point 4. Connect the other point on the potmeter to point 3 (gnd.).



HAPPINESS IS A NOT
TOO-WARM M1 TANDY!



USING NEWTON'S QUOTE "STANDING ON THE SHOULDERS OF GIANTS" AS INSPIRATION, I OFFER ANOTHER IN THE SERIES OF SOME OF THE KNOWN, AND LESSER-KNOWN NAMES WHO INSPIRED MUCH OF WHAT WE, TODAY, TAKE FOR GRANTED.

NORBERT WIENER

Pioneers In Computing

Norbert Wiener

The child prodigy whose study of mathematics resulted in the birth of the science of cybernetics

Norbert Wiener was born in 1894 in Missouri, USA. After taking a degree in mathematics at the age of 14 and receiving a doctorate in logic at 18, he went to study with David Hilbert at Göttingen, Germany.

Wiener's contribution to computer science came late in his life. For many years he worked at the Massachusetts Institute of Technology, studying the new probabilistic physics, and concentrating on the statistical study of the motion of particles in a liquid (a phenomenon known as Brownian movement). The particle movements were so unpredictable that it was impossible to describe them using the traditional physics of deterministic forces. So a 'probabilistic' method, by which only the *probable* location of a particular particle at a given time could be predicted, was the best that could be applied.

When the Second World War broke out he offered his services to the US government and began work on the mathematical problems implicit in aiming a gun at a moving target. The development of automatic gunsight guidance systems, his studies in probabilistic physics and his broader interest in subjects ranging from philosophy to neurology all came together in 1948 when he published a book entitled *Cybernetics*.

Cybernetics is the study of the self-governing controls that are found in stable systems, be they mechanical, electrical or biological. It was Wiener who saw that information as a quantity was as important as energy or matter: copper wire, for example, can be studied for the energy it can transmit or the information it can communicate. The revolution that the computer promises is based in part on this idea: a shift in the source of power from the ownership of land, industry or business to the control of information. His contribution to computer science was not a piece of hardware but the creation of an intellectual environment in which computers and automata could be developed.

The word 'cybernetics' is derived from a Latin word meaning 'governor'. Wiener had studied the 'governor' of James Watt's steam engine, which automatically regulated the machine's speed, and he realised that for computers to develop they must be made to imitate the ability of human beings to regulate their own activities.

The thermostat in a house is an example of a control system. It regulates the heating according to fluctuations in temperature above or below an optimum level. A human is needed only to set this level. Wiener called this faculty for self-regulation and control 'negative feedback' — 'feedback' because the output of the system (the heat) affects the future behaviour of the system and 'negative' because the changes the thermostat brings about are made to restore the temperature to the one set.

A system that can do this and also choose its own temperature (and other goals) is called a 'positive feedback' system. When an automaton can do all this and reproduce itself as well, then it approaches the human condition.

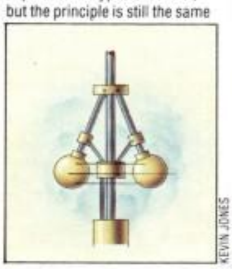
Wiener's theory of cybernetics could be regarded as a super science — a science of sciences — and it has encouraged research into many areas of control systems and systems that deal with information. Everything is information. What we know about the changes in the world comes to us through our eyes and ears and other sensory receivers, which are devices for selecting only certain data from a totality that would otherwise engulf us.

Information can also be studied in a statistical way, independent of any meaning it may have. For example, by observing the frequency with which certain symbols occur it is possible to break many types of codes. In the English language the letter 'e' occurs most often, and the letter 't' is the second most frequently used. By analysing large samples of a code and comparing the results with typical samples of English, it is possible to identify key letters and thus begin deciphering the code.

Wiener died in 1964, before the microcomputer revolution began, yet he foresaw and wrote about many of the problems that would arise in this new technology.

Speed Restriction

Wiener was fascinated by the idea of the steam governor — one of the best and simplest examples of negative feedback. Two weights are connected by pivoting arms to a spinning shaft, which is in turn connected to the flywheel of the steam engine. As the speed of the engine increases, the weights will fly outward. This movement, by means of a suitable linkage, shuts off the throttle of the engine slightly. This has the effect of stabilising the speed of the engine at any level set by the operator. Modern computers can implement far more sophisticated types of control, but the principle is still the same



KEVIN JONES

TRS-80 Emulators . com

TRS32: A Model I/III/4/4P Emulator For Windows

written by Matthew Reed

Unregistered Shareware Version:

- Works under all current versions of Windows
- Full Windows application — no low-level hardware conflicts!
- Model I, Model III, Model 4, and Model 4P emulation
- Four floppy disk drives (with optional realistic disk drive sound)
- Cassette tape drive with graphical on-screen controls
- Exatron Stringy Floppy emulation
- Printer support
- Serial port for RS-232 communications
- Joystick support (using a Windows joystick — TRISSTICK and Alpha Products joysticks are emulated)

Registered Version:

- All features included in the shareware version
- Built-in emulation of an Epson FX-80 dot matrix printer (including graphics and control codes)
- High resolution graphics (Radio Shack and Micro-Labs)
- Up to 1 megabyte of additional memory in Model 4 and 4P modes
- Hard disk support
- Orchestra 85/90 music generation

Actual Dotwriter output — emulated FX-80
click for larger sample

Interested?

- Read the TRS32 emulator documentation
- Download the shareware version
- Register online

IN MAV'S FreHD WORKSHOP
Ian Mavric

HOW TO CONVERT TRS-80 MODEL 1 TO FreHD AUTO BOOT

Over the next few editions of TRS8Bit I'm going to explain, step by step, how to upgrade the TRS-80s internal ROM to the auto-boot EPROM. The EPROM can be produced by downloading the BIN file from the downloads page from my web site, or if you don't have your own EPROM programmer, I can burn and send you the EPROM for a small cost.

As a starting point I chose the Model I, and future instalments will follow the TRS-80 line chronologically, so next time will be the Model III, then the Model 4NGA in the issue after that, then the 4P and finally the Model 4GA/4D. For those people who don't want to wait for future editions, just email me and I'll explain how to upgrade your Model III/4/4P/4D.

Without further ado, the Model I:

System requirements: 16K Level II Model I, with lowercase and attached to a 32K Expansion Interface. You also need a FreHD.

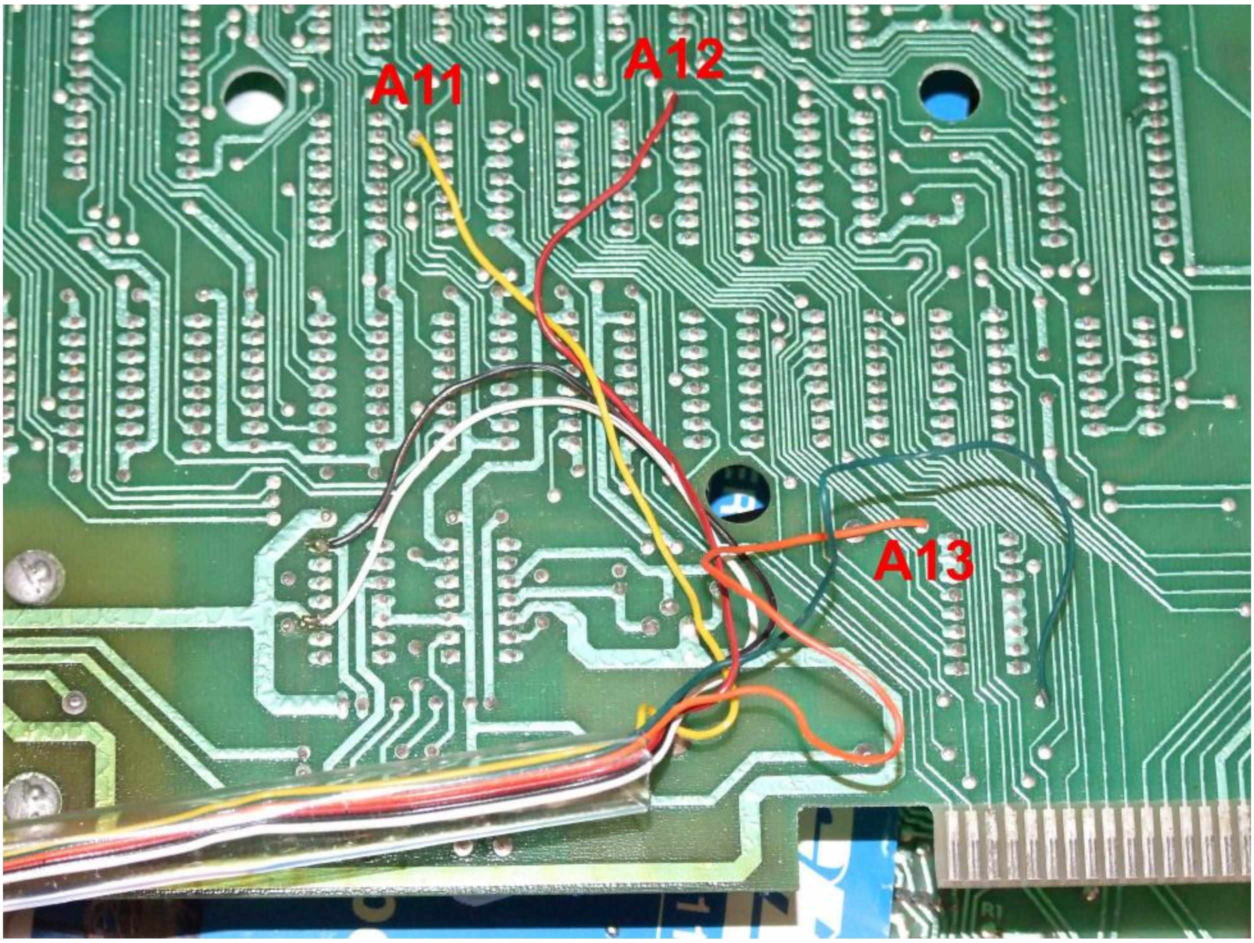
Modification summary: The existing Level II ROMs and/or daughter-board will be removed from the system, a programmed EPROM installed in Z33, 5 wires soldered to various points on the motherboard, and a toggle switch installed on the case to change between standard Level II BASIC and auto-boot FreHD. The modification is reversible because no trace cuts are made. As usual with modifications, make sure you understand each step because I can't be responsible for changed made to your computer if things go wrong.

The following seven steps will upgrade your computer:

1. Check Z3 and make sure all 8 shunts are in place. If they aren't, install a switch block so all 8 shunts connect. (Typically Level II systems with the daughterboard will have all shunts in place, and Level II systems without a daughterboard will have 7 of the 8 shunts in place.)

2. On systems with the daughterboard, you need to locate 3 long wires (out of 4 or 6) and <u>note where they go</u>. Call the orange wire A13, the red wire A12 and the yellow wire A11. We need to refer back to these points later on so mark them and then unsolder the 4 or 6 wires running to the daughter board and remove the whole Level II Rom assembly.
Z33 and Z34 should now be empty.

3. On systems without the daughter board, just remove the two chips in Z33 and 34. Referring to the picture below, make a note on your PCB where the orange, red and yellow wires solder to, and mark them as A13, A12 and A11 respectively.



4. Program your 27C256 and prepare it as follows. Bend out pins 20, 23, 26. Clip the pin portions from pins 1,2,27,28, leaving just the tabs left to solder wires to.
Solder a purple wire joining pins 1 and 28, and leave a short 5cm length of purple wire.
Solder a 5cm black wire to pin 20.
Solder a 20cm orange wire to Pin 26.
Solder a 10cm yellow wire to Pin 23.
Solder a 15cm red wire to Pin 2.
Solder a 30cm grey wire to Pin 27.

5. Insert the prepared 27C256 into Z33 with pins 1,2,27,28 overhanging at the front of the IC. Make sure none of these pins contact anything, or each other. Also confirm that pins 20, 23 and 26 of the Eprom don't go into the holes in Z33's socket.

6. Solder wires to motherboard:
Black wire to a source of GND (thick rail directly below Z33/34)
Purple wire to a source of +5V (solder point directly above Z34 pin 24)
Referring to the notes you took earlier, solder the yellow A11, red A12, and orange A13 wires to the points noted in step 2 or 3. Solder them on the component side of the PCB, confirming you have the correct place with a multimeter.

7. The one wire (grey) you have left over is the A14 line from the Eprom, and when high selects FreHD auto boot, and low selects regular Level II Rom. Mount a toggle switch on your Model I case somewhere, and run the grey wire to the centre pole. Wire the other two poles of the toggle switch to the GND line and the +5V line and the modification is complete.

Power up and test the machine. You may need to turn down the voltage at R5 and R10 (refer to TRS-80 Tech Ref manual 26-2103 page 60) because the single Eprom replaces high-power drawing devices (Z33 and 34 or the daughterboard assembly).

If problems crop up they will when you first power the machine up. First power up the computer without the Expansion Interface connected and run test routines to make sure the computer works properly. If the system fails to start check all the wires, that there are no solder-bridges shorting something out. It's possible if the Eprom had a bad burn then they system will not run, so unsolder all wires and go back to the regular Level II Rom set. Confirm the computer still runs and then go over the instructions and burn a new Eprom if needed, and try again. I've upgraded several machines this way and at least three other people have successfully made the modification as well.

I'd like to thank Dean Bear for all his hard work in coming up with this solution for Model I owners.

Next time: Upgrading the Model III with FreHD auto-boot.

Ian Mavric
ianm@trs-80.com

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.

Cartridge Drive or Stringy-Floppy

CW Hobbs

This article first appeared in INMC NEWS80 Edition No 3, FEB-APR81.
I was sent it because of my Aculab enthusiasm and although it was originally for NASCOM systems I hope you find it of interest. The 25-300 was news to me!

This article describes the fun I had interfacing the 'Electronic R/W System Model 25-300' to my Nascom 1. The peripheral itself, whilst certainly electronic and a read/write system, is probably better described as a cartridge/wafer drive, although I have seen it described in some magazines as a 'Stringy-Floppy'. I feel that this is a misleading and unnecessary name.

I order the unit by telephone from Breda, Holland on Wednesday morning after having all my technical queries answered immediately in excellent English and the unit arrived the next day by post. (I do live in Holland though).

I received a deck and control board, and for further payment, a box of wafers, which were of assorted lengths, from 5 to 50 feet, giving 20 to 200 second of recording time. These cartridges consist of an endless loop of magnetic tape with an End Of Tape (EOT), which is also the Beginning Of Tape (BOT) marked by a piece of reflective tape.

Unlike many pieces of kit, this drive came with very adequate documentation but with no case or mounting brackets. Any capable metalworker with nothing more than a small lathe, milling machine and a borer could knock uo a suitable mounting in a few weeks! Anyway, I hacked and puffed and blew and finally mounted it in such way which I don't think it will expose me to public ridicule. I feel more confident on the grounds of electronics and program design so I'll hastily move on!

The drive demands very modest +5V and +12V supplies and in return offers the following interfaces on a small edge connector :-

Input	Output
Start Motor	EOT detected
Fast Speed on Motor	Wafer is Write Protected
Enable Writing	Serial Data from Wafer
Select Deck	
Serial Data for Recording	

The beauty of the 'Select Deck' input is that when it is held false, everything in sight goes into high impedance state and consequently, more than one deck may be connected together.

The hardware interface wins no prizes for complexity but is very adequate. (A schematic from the original article is show at the end of this article ED). There are LED's, which are necessary because once the wafer is inserted into the drive, it is impossible to see whether there is a Write Protect tab attached.

The 100K resistor pulling up to the UART serves not only to make the pulses nicer but it is essential because when the deck is not selected the Data Out line goes high impedance and picks up all sorts of rubbish. And what does the Nascom do when it is idle? It picks up rubbish and displays it at great speed on the screen making it impossible to type a command in to select the deck again and stop the monitor picking up the rubbish!! I must be honest and admit that the resistor was not part of the original design.

After building a simple interface and a few test programs I felt ready to fill my second 2708 socket with the 'Hobbs Un-copyrighted Cartridge Operating System'. Surely a most impressive title. Here I tempered good design techniques (I design computer systems all day as well) with the knowledge that it all has to fit into 1K bytes plus whatever I could save by taking 'Dump' and 'Load' out of Nasbug. The original 4 layer design has reduced to 3 layers :-

A) Drive level interface with routines for
- rewinding the wafer
- erasing a wafer
- pre-marking (sectoring ? ED) a wafer
- finding a sector
- reading a sector
- writing a sector

B) File level interface which sits on top of the drive level interface and offers routines for
- create a file
- erase a file
- open a file
- writing a sector to a file
- reading a sector to a file
- returning the status of the file

C) Operator level interface which sits on top of both the other levels and which offers the following commands, augmenting those of Nasbug
- i) Drive level commands
  - W3 rewind the wafer in drive 3 and initialise the PIO
  - V5 display the catalogue of file on the wafer in drive 5
  - P6 1234 initialise a new wafer in drive 6 as number 1234

ii) File level commands

R5:PRINTFILE remove the file named 'PRINTFILE' from the wafer in drive 5

A3:PRINTFILE 12 add a file named 'PRINTFILE' 12 sectors long to the wafer in drive 3

L3:PROGFILE load the object code in file named 'PROGFILE' on the wafer in drive 3

D1:PROGFILE 400 800 dump the contents of memory between 400H and 800H to the file named 'PROGFILE' on the wafer in drive 1

So what does the tape actually look like when viewed under a magnetic microscope, how are the sectors formatted, what does the directory look like, how do I fit it into 1K bytes? Indeed, can I?

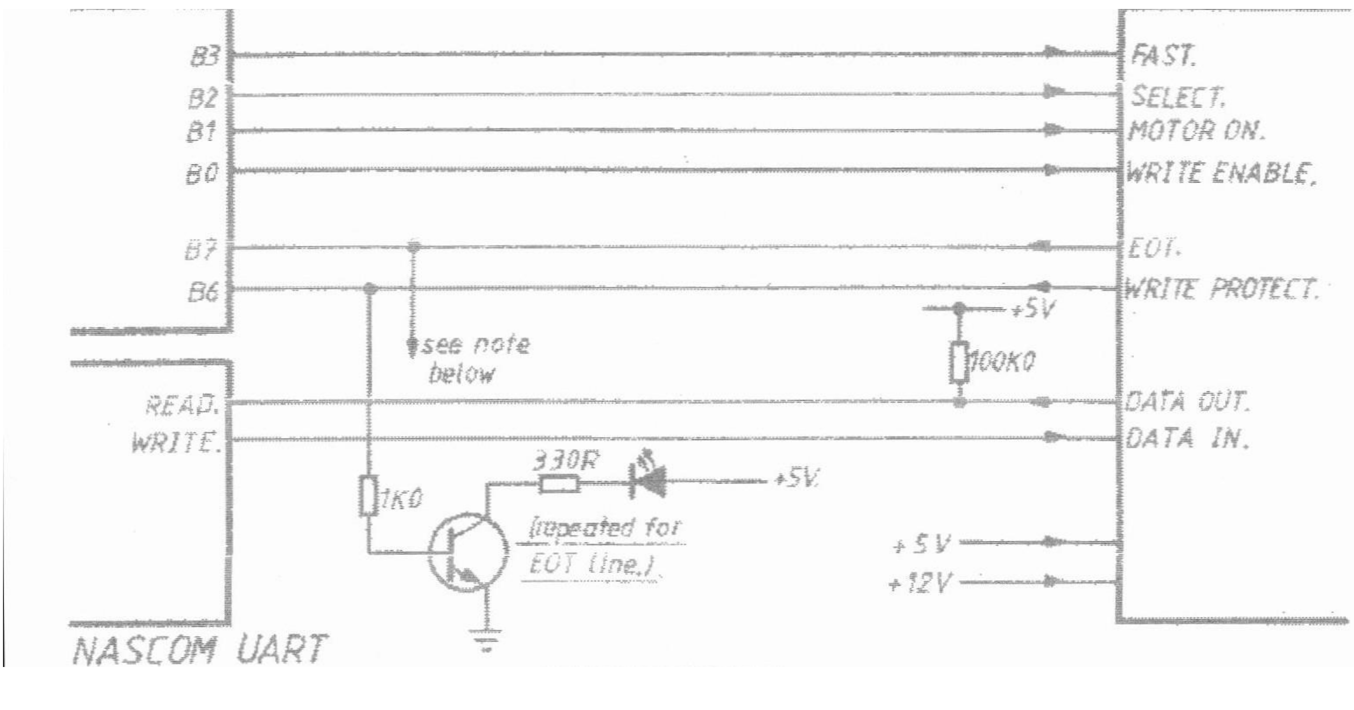
I refuse to answer here, but can offer some morals viz.

Make the hardware as simple as possible at first. Software is more flexible and once you know the interface really well, it is easy to increase the load on the hardware and decrease the hardware.

Separate the software functions into clearly defined layers, decide whether subroutines will save registers or not, and start the design at the top with the coding at the bottom.

When you have only 1K bytes of memory do not stick too rigidly to the strict layering of design.

(AT THIS POINT, THE INMC NEWS80 EDITOR ASKS FOR FURTHER DETAILS, PERHAPS IN ANOTHER ARTICLE, OR PROGRAM(S) FOR THE LIBRARY. I WOULD LOVE TO KNOW IF ANY FURTHER PROGRESS WAS MADE AND A RUNNING, WORKING SYSTEM WAS EVER PRODUCED? I DON'T SUPPOSE WE EVER WILL THOUGH, UNLESS YOU KNOW BETTER!)



IN MAV'S WORKSHOP

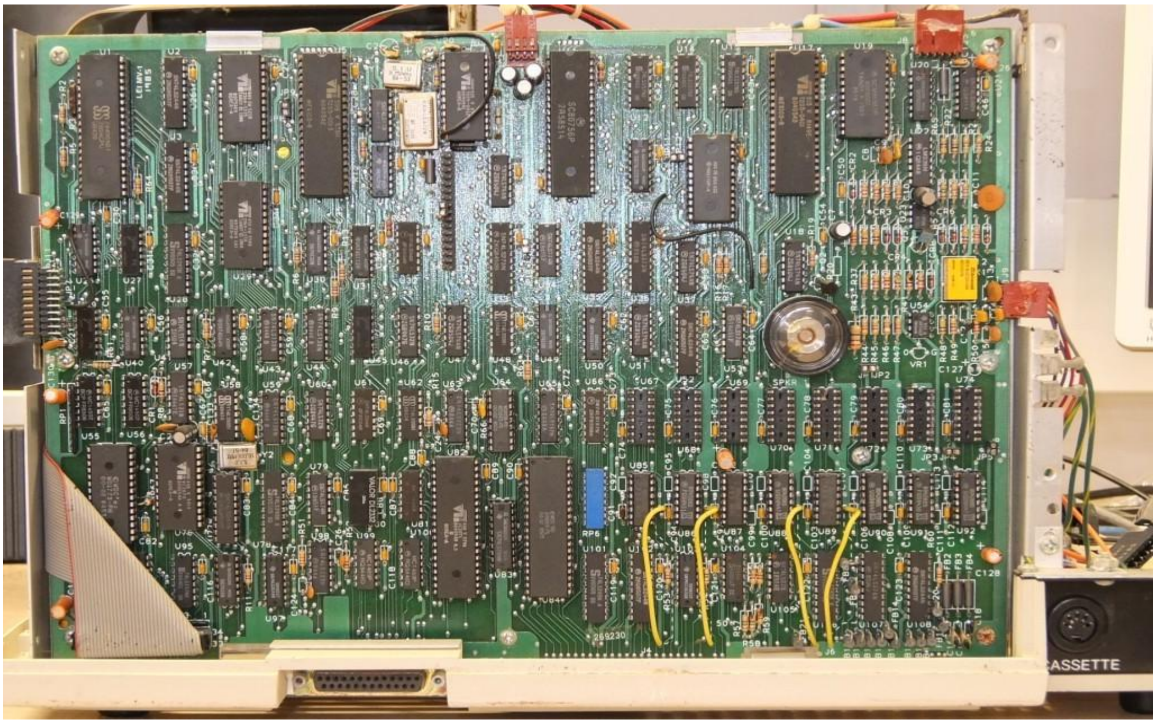
Ian Mavric

UNDERSTANDING THE MODEL 4 GATE-ARRAY MOTHERBOARD

In the middle of 1983 Tandy released the Model 4 Microcomputer, one of the finest 8-bit Z80-based computers of all time, and the rest was history. Tandy was onto a winner and for those who like to tinker with TRS-80s, the Model 4 is the natural choice, and it sold in the hundreds of thousands of units. Looking for ways to cut production costs and improve reliability (that was not to say the Model 4 was unreliable) advances in electronics in the early 1980s led to an entirely new motherboard for the Model 4 around mid-1984. Instead of a plain motherboard with separate FDC and serial board, everything was combined into a single motherboard, with a lower chip count due to combining arrays of TTL logic chips into single chips called gate-arrays. We (and Radio Shack, for that matter), call these Gate-Array motherboards, hereafter known as M4GA boards.

A LOOK AT THE M4GA BOARD

We see it has a number of large 24, 28, or 40 pin chips which are logically laid out and while the board may look imposing, especially when repairs are needed, if you know what each part does you can logically track down problems fairly quickly.



U1: Z80A CPU - Needs to be a Z80A, I have seen boards which have a plain Z80 which is not fast enough to run at 4MHz in Model 4 mode for any length of time. If your computer won't start check for CLK signal on Pin 6.

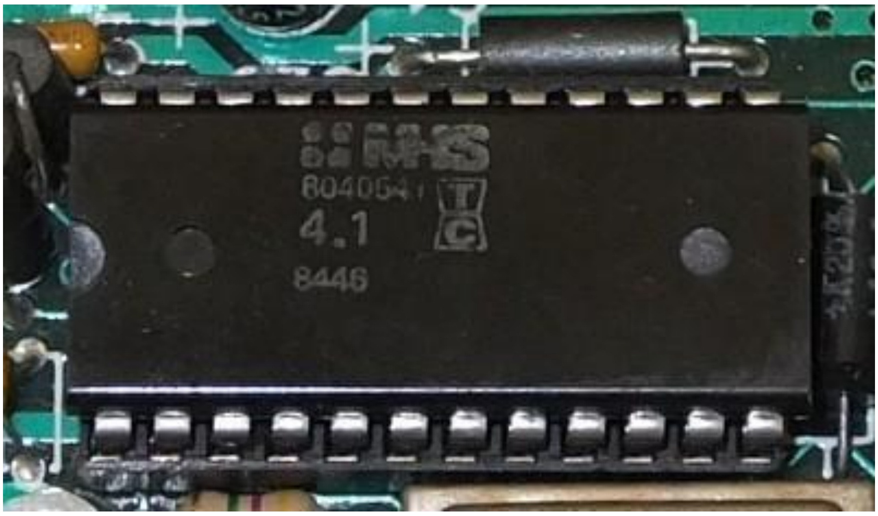
U4 (ROM B/C) and U29 (ROM A): Model III Basic and boot disk routine. When we upgrade a M4GA to FreHD auto-boot, U4 is replaced with a programmed EPROM.

U5 8040542 Gate Array "4.2":

Address Decoder - if the computer won't get past a junk screen, and you know the memory chips are all OK and that there is a CLK signal on Pin 6 of the Z80A, suspect this chip.

Y1 and U8 (Crystals): Y1 is for Model III mode (2.02MHz) and U9 is for Model 4 mode (4MHz). I have seen systems where Model III mode works but Model 4 mode doesn't - replace U8. If Y1 doesn't work then the computer will not start at all.

U9 8040541 Gate Array "4.1":



Clock switchover chip controls which frequency is used in which mode. If this chip doesn't work then the computer may run slow (2MHz) in Model 4 mode, or the system will just not start up.

U11 SC80756P: CRT Controller - controls the video sync on the screen - if you have lost horizontal or vertical sync that can't be adjusted by the controls on the Video PCB, then suspect this chip.

U16 HM6116P: Video RAM chip - if you see spelling errors on the screen or characters is missing but the software otherwise works properly, suspect this chip.

U17 8040543 Gate Array "4.3":

Video Gate Array - works in conjunction with U11, U16, and U19 to display information on the CRT. When this chip fails, nothing will be visible on the CRT, even though the computer still runs and responds to commands.

U19 8049007 Character Generator: forms all the letters, numbers, or symbols you see on the screen. If the characters are malformed or missing dots, suspect this chip.

U64 WD8116: Baud Rate generator - if the serial port doesn't work or if some of the baud rates work and others don't, suspect this chip. But check other RS232 problems below on chips U82, U83 and U84 (below).

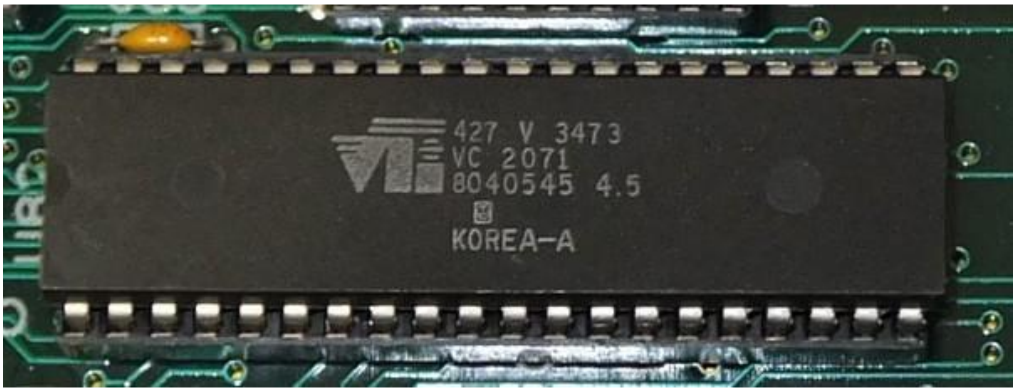
U75 WD1773 Floppy Disk Controller chip: industry standard double density floppy disk controller. If the computer works in cassette mode but won't boot known good disks from known good disk drives (as tested on another computer) then suspect this chip or U76.

U76 8040544 Gate Array "4.4":

Disk Drive Interface Oscillator - critical timing array to make U75 work. If U75 is known good (as tested on another computer) but the disk drives still don't work right, then suspect this chip.

Y2 (Crystal): needed for floppy disk system (U75, U76) to work. If it doesn't work then neither U75 nor U76 will do anything.

U82 8040545 Gate Array "4.5":



RS232 support gate array, controls the interface between the RS232 serial port (U64, U84, U81, U99, U100) and the processor. If diagnostic software reports no RS232 port installed, suspect this chip.

U84 TR1865PL: UART - Universal Asynchronous Receiver Transmitter which is the heart of the RS232 port. Easy to blow this chip by plugging in poorly designed RS232 devices or any device which sends uncontrolled voltages into the RS232 port.

THE LEGACY OF THE GATE-ARRAY MOTHERBOARD

The M4GA motherboard was such a good design that it allowed Radio Shack, two years later, to take a great computer and install new modern double-sided disk drives, add an extra key and a new nameplate, and bingo the "Tandy TRS-80 Model 4D Microcomputer System" was born. Yes, the 4D uses the M4GA motherboard. The 4D was on the market for another four years making the M4GA motherboard one of the longest production motherboards in any Tandy/TRS-80 series.

Next time: Understanding the Model 4P Gate-Array motherboard.

Ian Mavric

ianm@trs-80.com

TROUBLESHOOTING GUIDE FOR "NO BOOT" PROBLEMS -- TRS-80 MODEL 4

Larry Kraemer

1. The very first thing to VERIFY is the Proper Voltages of the Power Supply(s). The Radio Shack Model 4 shipped with one of two manufacturer's Power Supplies, Astec & Tandy.
Both have three Output Voltages:
+5.00 VDC -- Must be within 3%.
-- Power Supply regulates from +5 VDC and +12.00 VDC, so these must be correct.
+12.00 VDC -- 5%
-12.00 VDC -- +25%, -8.3%

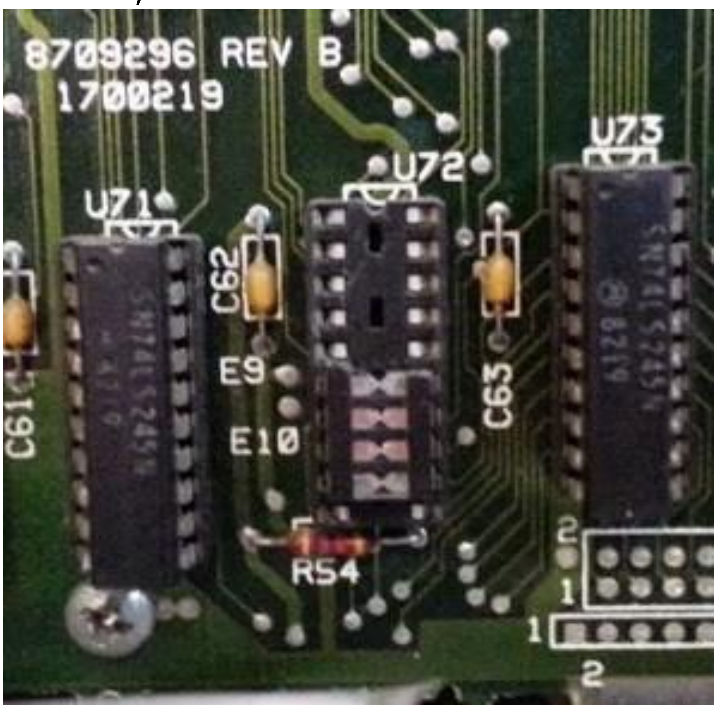
The Model 3 and Model 4 also have a -5.0 VDC Supply, either behind the Motherboard (Model 3) or located on the Motherboard (Model 4 - Q5). This is generated from the -12 VDC Supply, with a 7905 Voltage Regulator, and is ONLY used with 4116B RAM Chips (16 x 1).

If your Motherboard has MCM6665BP15 or Tandy #8040665 RAM Chips and the (Exx) Jumpers have been set correctly, the -5 VDC is not used.

2. Verify Jumpers for 16K or 64K RAM IC's -- These are located immediately above the RAM IC's.

16K (NO RAM IC's in Sockets U85-U92, 16 x 1 4116B RAM in Sockets U77-U84)
E5-E6 Jumpered
E1-E2 Jumpered
E12-E13 Jumpered

SHUNT at U72 has Pins 6-15, 7-14, 8-13 & 9-12 Jumpered in U72

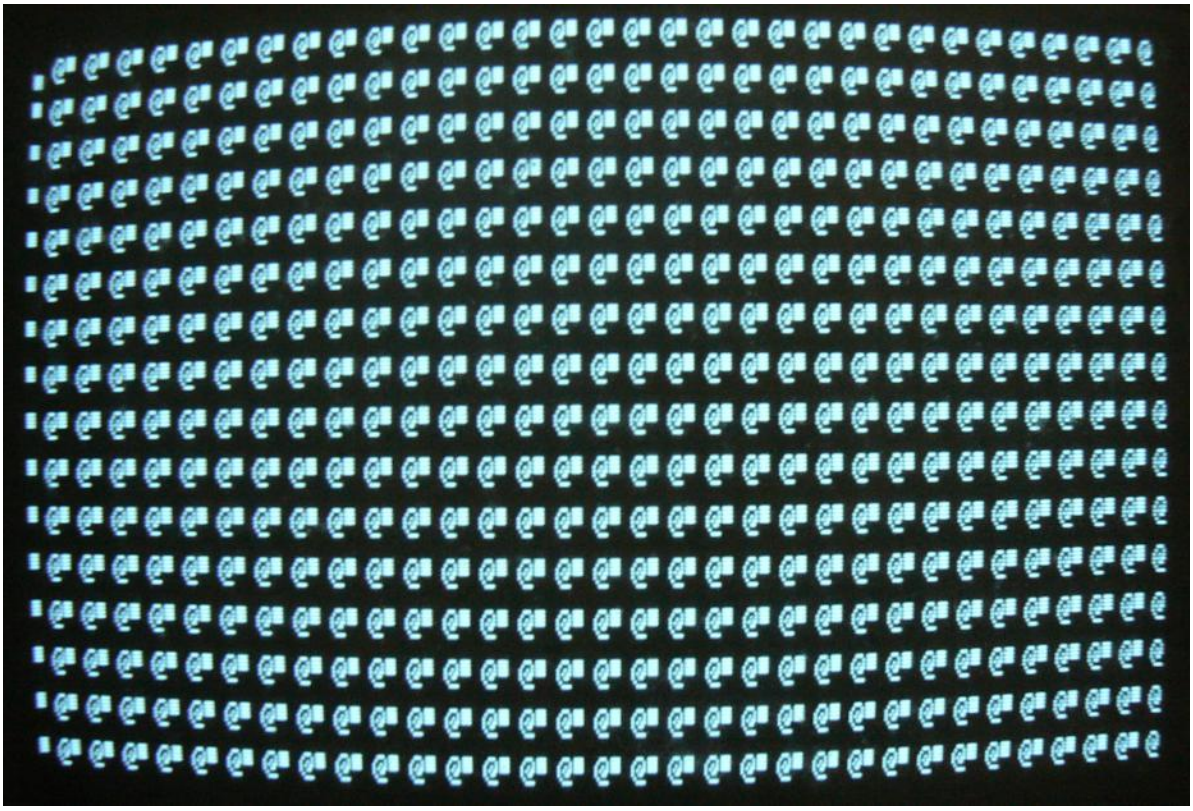


64K (64 x 1 MCM6665BP15 RAM IC's in Sockets U77-U84)
E5-E4 Jumpered
E3-E2 Jumpered
E12-E11 Jumpered
SHUNT at U72 has Pins 6-15, 7-14, 8-13 & 9-12 Jumpered in U72

128K (64 x 1 MCM6665BP15 RAM IC's in Sockets U77-U84 & UU86-U92)
E5-E4 Jumpered
E3-E2 Jumpered
E12-E11 Jumpered
REMOVE SHUNT at U72 and insert Tandy #8075468 PAL CHIP

3. Power up the Model 4, with the Break Key Held Depressed.
You should see "cass??" on the Display. If you see nothing but repeating characters, it suggests that the Z-80 is OK, and that video RAM (and character display, etc.) are fine. It just didn't boot properly.

While this doesn't rule out a defective: CPU, Buffer, Video Controller, RAM, Video RAM, or something fiddling with an Address or Data Line, it is a good sign to have Garbage Characters or repeating Characters on the Display. The Computer is at least trying to boot.



My Motherboard says 1982 and has the following numbers -- 8709296 Rev C 1700219
It is a Non-Gate Array Motherboard.

Check CPU (Power, GND, CLK, & Control Signals that are INPUTS to CPU: /WAIT, /INT, /NMI, /RESET)

These Signals are Active LOW, and must NOT BE STUCK LOW, for Z80 to boot properly.
RESET is the ONLY one you can easily manually activate, by the RESET Pushbutton.

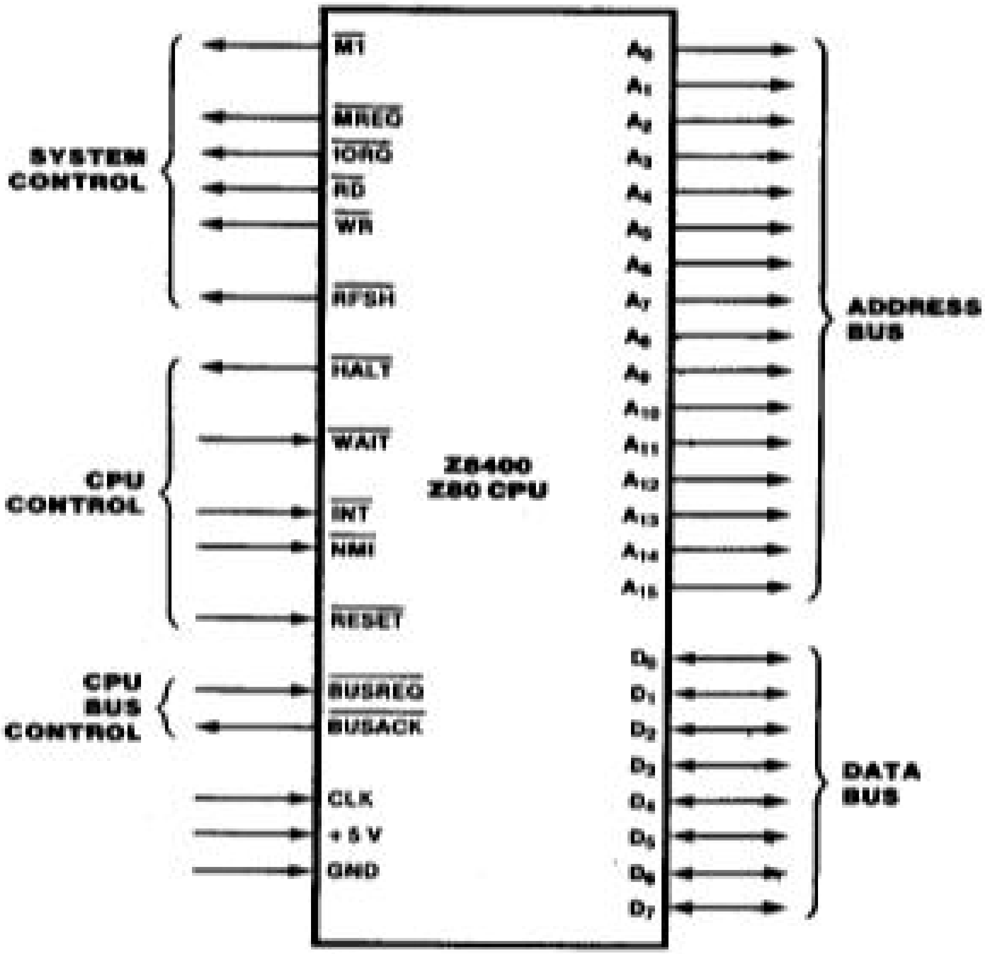


Figure 1. Pin Functions

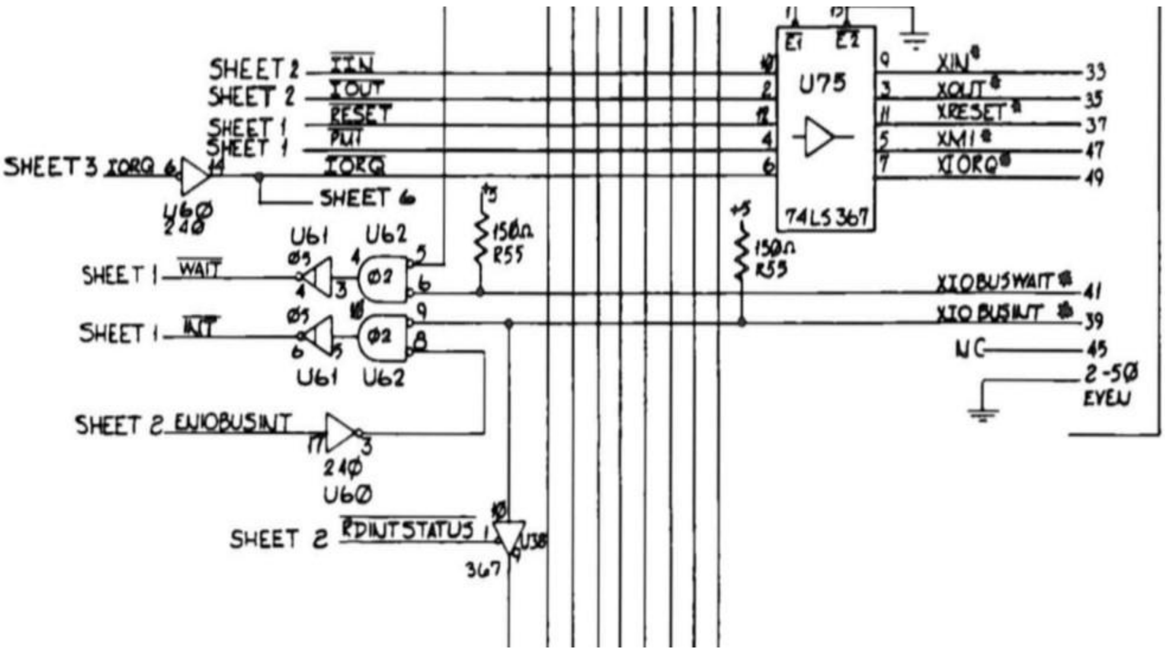
If you have any of the INPUT Control Signals (/WAIT, /INT, /NMI, / RESET) that are STUCK LOW all the time, that problem has to be tracked down, and repaired.
Don't forget that most of these Control Signals are created by using PULLUP Resistors that are bussed, in a SIP Package.

Pin 1 is COMMON to all other Pins and from the Common Pin to any other Pin is 4.7K OHMS
(RP2). +5.00 VDC is applied to Pin #1, and all other Pins of the SIP are PULLED UP to +5.00 VDC.
Logic Chips at different locations take a Pin on the SIP to GND to produce the Active LOW Signals.

A bad Solder Joint, a defective Resistor internal to the SIP, or some Logic IC can create a condition where the Control Signal is ALWAYS LOW (or not being PULLED UP to +5 VDC because of a bad connection.)

I happened to have /INT LOW all the time.

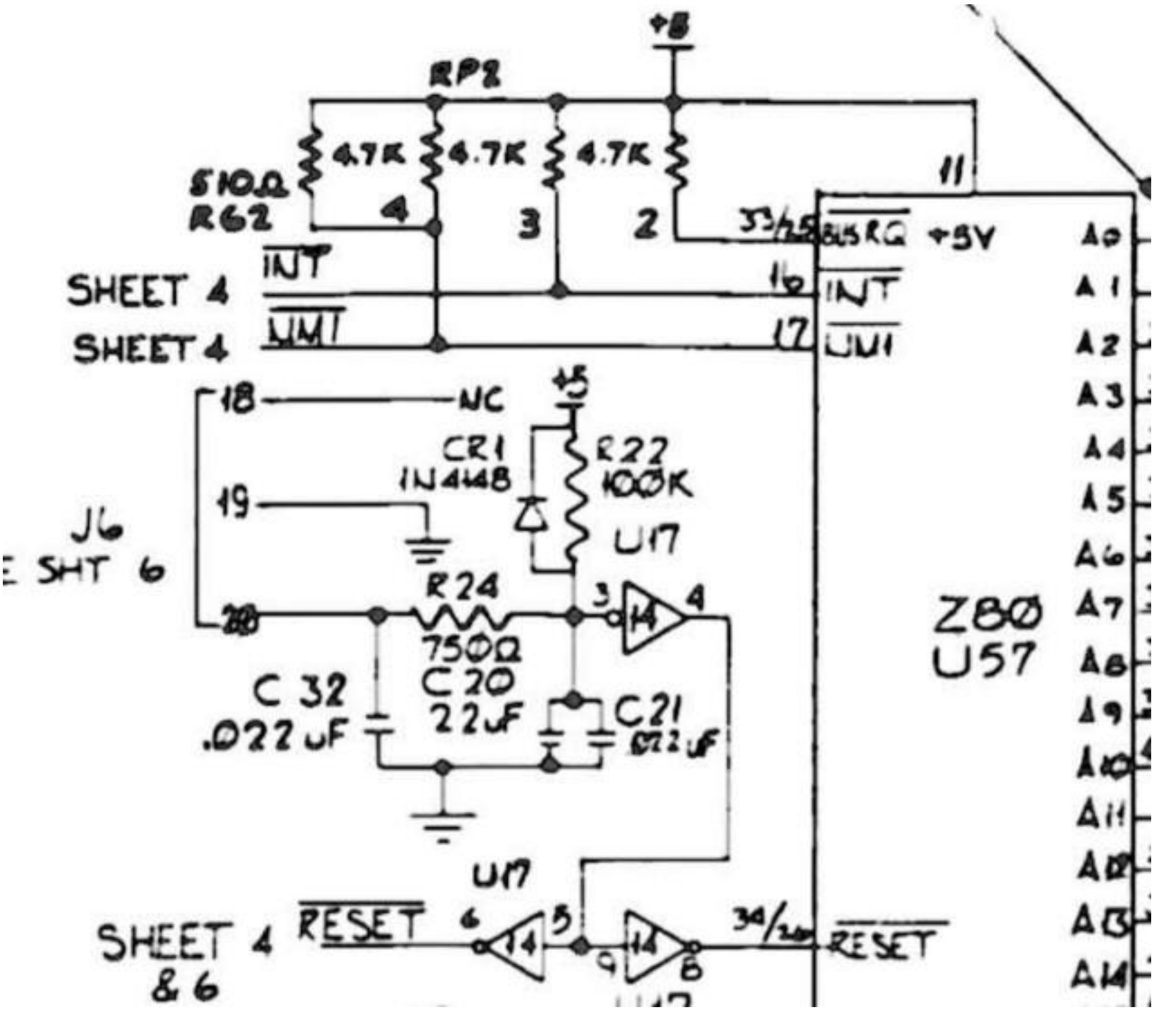
1. /WAIT Pin 24 Toggles
2. /INT Pin 16 LOW ALL THE TIME
3. /NMI Pin 17 Toggles
4. /RESET Pin 26 Toggles
5. /BUSREQ Pin 25 Appears to be High all the time
So, I started with the Circuit that generates /INT.



It appears that U61 (7405) has a LOW on Pin 5 and also a LOW on Pin 6. I've removed the CPU and have the same Signal levels. The IC U62 (74LS02) has a HIGH on Pins 8 & 9 which should produce a LOW on Pin 10.

Now, it's time to replace IC U61.

After removing U61 (SN7405), and locating a 14 Pin Socket, I soldered in the socket, and inserted a new SN7405. After replacing the Motherboard in the case, I applied Power. I still had a LOW on the /INT (Pin 16) of the CPU. I thought that's impossible. But, as I searched further in the Schematics I located a SIP of PULLUP Resistors. This package is labeled RP2, is immediately above the CPU Socket, and has +5VDC applied to Pin 1, with Pins 2 thru 8 being 4.7K PULLUP Resistors.



I've got the /INT Signal on the Z80 (Pin 16) working properly now. I had the problem narrowed down to the SIP Resistor RP2, but couldn't believe that it OHMED out at 4.7K, but didn't pull Pin 16 of the CPU "HIGH".

This morning I removed the Motherboard, and un-soldered only Pin 3 on the SIP RP2 and re-soldered it. After replacing the Motherboard I still didn't have a "HIGH" on /INT (Pin 16).
It was being held LOW by U61 (Pin 6). As it turned out, the New Old Stock SN7405
I had inserted was defective. Luckily, I had another SN7405, and I replaced the defective one. So, now /INT is "HIGH" for the Z80.

TIP:
When unplugging or plugging in the flimsy ribbon cables on top of the Motherboard, use an ordinary wooden lead pencil placed under the sharp bent. Then grip the flimsy cable with your fingers and easily remove or replace it. Be careful as these flimsy cables are fragile.

My Clock Signal now looks like a normal clock signal. All Address Bus signals look normal too. All CPU Control Signals are now in the correct state:

Pin 16 ___ /INT = HIGH
Pin 17 ___/NMI = HIGH
Pin 24 ___/WAIT = HIGH
Pin 25 ___/BUSREQ = HIGH
Pin 26 ___/RESET = TOGGLES "LOW" on RESET PUSHBUTTON

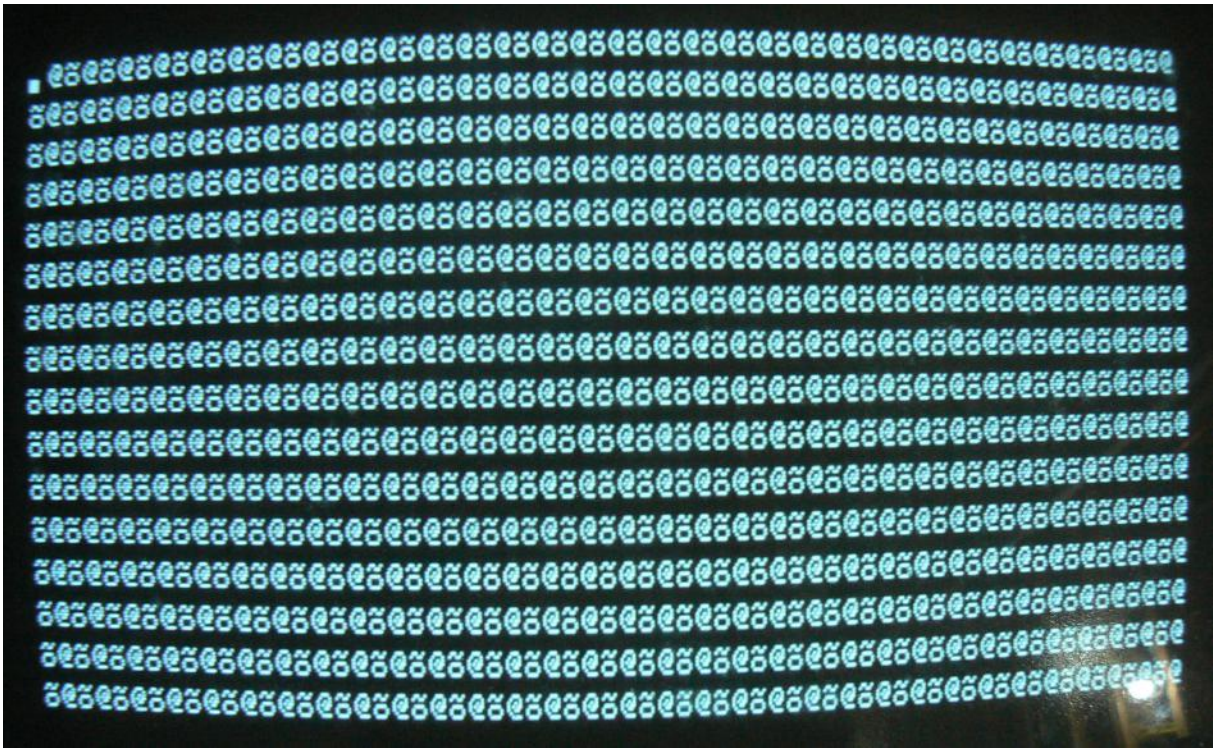
5. In addition to checking the CPU, it's a good idea to check the two SIP Resistors RP1
(820 OHM) & RP2 (4.7K) in two different ways. Check each one with an O'Scope while the
CPU is inserted in the CPU Socket, and with the CPU Removed from the socket. That way you can get a good check on the Signals that control the CPU. You may even want to remove an IC, then insert a IC Socket and insert a new Chip. That will allow you to Toggle the Control Signal that will end up being the INPUT Control Signal for the CPU. U61 or U62 would be a good example of an IC that may need to be inserted into a IC Socket so the /INT or /WAIT Signals
could be VERIFIED. In the table below you can write in the actual OHM value for each Pin and circle the pulse state as HIGH or LOW.

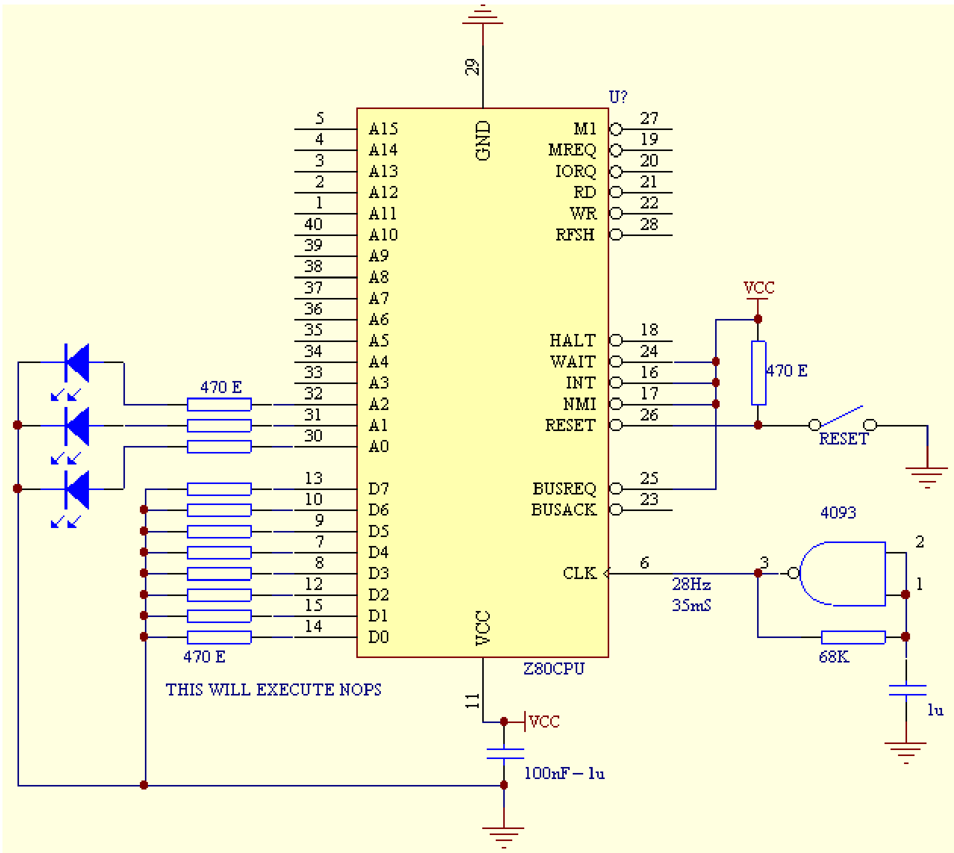
10 Pin SIP (820 OHM)___________8 Pin SIP (4.7K)

01 Common Pin tied to +5VDC__1 Common Pin tied to +5VDC
02 820 _______ HIGH/LOW_____2 4.7K ________ HIGH/LOW
03 820 _______ HIGH/LOW_____3 4.7K ________ HIGH/LOW
04 820 _______ HIGH/LOW_____4 4.7K ________ HIGH/LOW
05 820 _______ HIGH/LOW_____5 4.7K ________ HIGH/LOW
06 820 _______ HIGH/LOW_____6 4.7K ________ HIGH/LOW
07 820 _______ HIGH/LOW_____7 4.7K ________ HIGH/LOW
08 820 _______ HIGH/LOW_____8 4.7K ________ HIGH/LOW
09 820 _______ HIGH/LOW
10 820 _______ HIGH/LOW

6. At this point you need to verify each Address Line to make sure none (A0-A15) are STUCK LOW or HIGH. To do this I use an old Z80 CPU, and Bend the DATA Line Pins out so they won't go into the 40 Pin Socket. I solder a Jumper wire from D0 THRU D7 and solder it to GND. Now, the CPU is strapped to always get a 0x00 (NOP) on it's DATA Bus. The CPU will increment the Program Counter, and Increment the Address Lines to the next address. This way you can use your O'Scope to look at each Address Line, making sure they are all toggling, and aren't STUCK HIGH or LOW. You can even check the BUFFER IC's, Checking the INPUT side versus the OUTPUT Side of the Buffer. You can also remove IC like ROMS, Video Controller, Video RAM, and Buffers, to help Isolate the Address Line problems, assuming those IC's are in sockets.

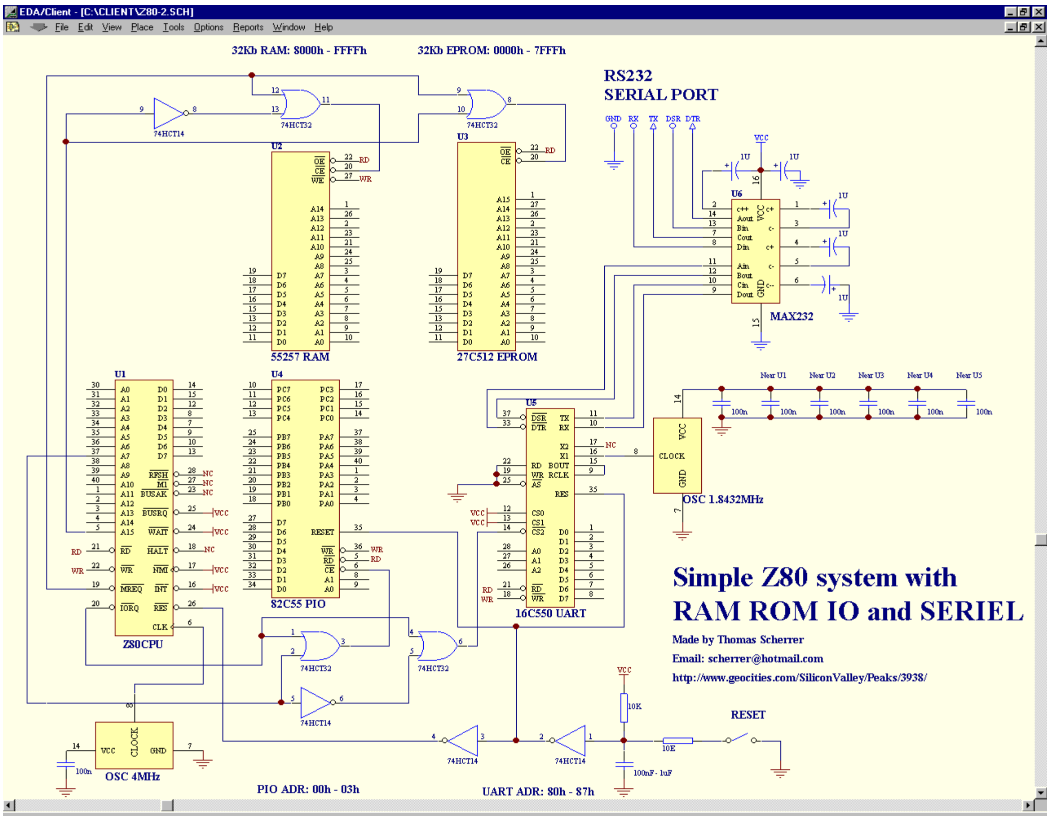
I removed all my 128K of RAM and now my Model 4 has this on the Video Display.
It appears to be 0x40 & 0x13 (19 Decimal is the same Graphic Character





in the Model 4 manual)

TEST Circuit that has DATA Strapped to 0x00 (NOP) to test just the Z80 CPU.

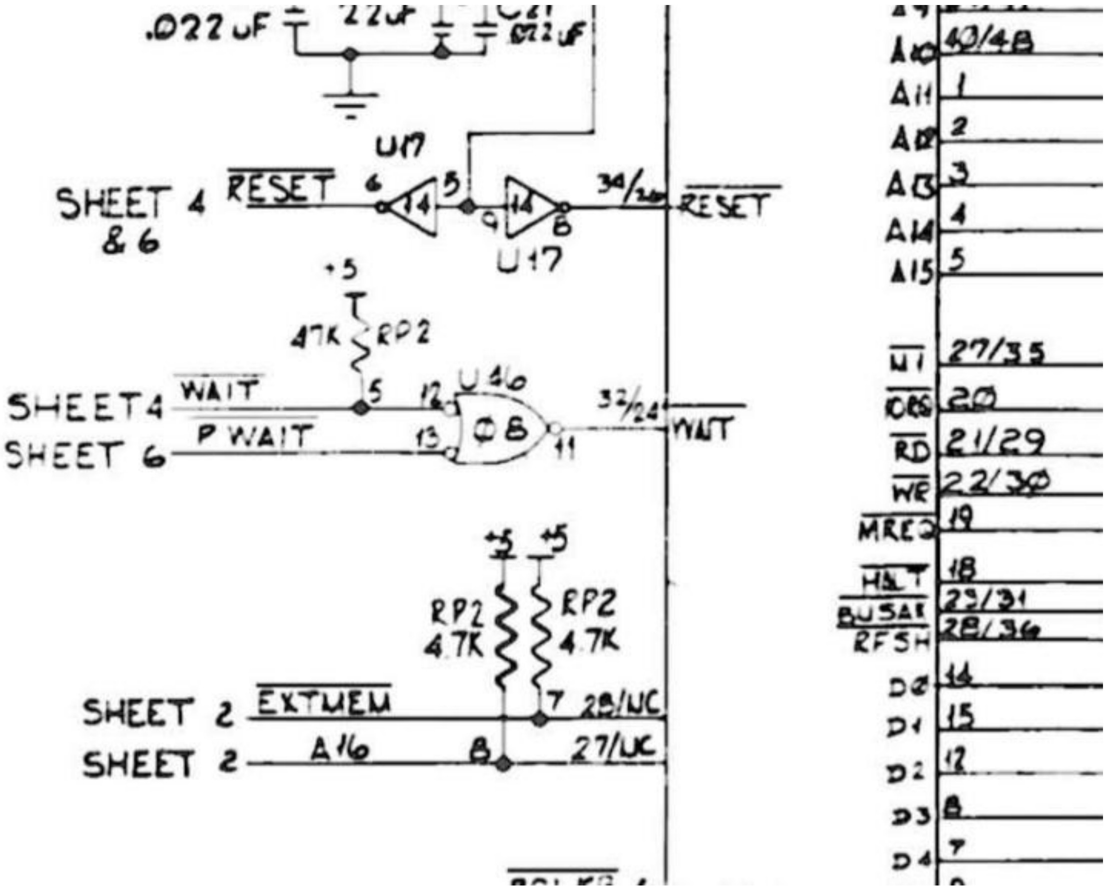


Another Z80 Circuit that shows Circuit Details about RESET Button.

It might also be wise to keep in mind that you can also break out the Clock Signal (CLK) and create a clock signal that you can manually toggle to single step through the Address Lines.
A Pushbutton, fed with +5 VDC can be connected to a 74HC14, and fed to the CLK Input of the CPU. If you do this, the RESET signal must be HELD LOW longer than about 6 Clock signals to reliably reset the CPU, or you will get odd results. So, just HOLD the RESET Pushbutton DEPRESSED (Line LOW) and give the CPU 6 or 8 CLK Signal Inputs from your CLK Pushbutton. The CPU should be correctly RESET. Now you are just 65536 CLK pulses from verifying all the Address Lines.

You can now single step through all Address Lines, and check or chase signals across the motherboard looking for problems.

Since I had trouble with the RP2 SIP 4.7K Resistor, I saw the RP1 SIP, slightly left and a bit lower than the Z80 CPU. I was wondering if any of those PULLUP resistors were not soldered good causing the same type of problem. So, I used my O'Scope to look at each pin. All the Signals were High. I just by chance stuck my O'Scope probe back on RP2, and looked at ALL the Pins. When I got to Pin 5, I had a CRAZY looking Signal for /WAIT. That feeds Pin 12 of a 74LS08 that goes from Pin 11 to the CPU /WAIT CPU (Pin 24) Input. At that point I powered down, and removed the CPU. Now, I had a CLEAN Signal on Wait.



I replaced the Z80 with a New Old Stock Z80B I had ordered years ago. It finally works again.

Shorted IC's or SHORTED conditions on the Motherboard can be located by building yourself a milliohm adapter for your Multimeter, or by purchasing the LEAKSEAKER 89 from:

http://eds-inc.com/product/leakseeker-89/
http://users.tpg.com.au/pschamb/lom.html

There is a good video on how to use the LeakSeeker 89 at:
https://www.youtube.com/watch?featur...&v=BUyEe8G50D4

If you build yourself the milliohm adapter, it will work the same way as the LeakSeeker. OHMING from a point on an Address Line to another pad will give you a changing (decreasing) milliohm value as you approach the SHORT.

There are several URL's of inexpensive SHORT BEEPERS, and SHORT DETECTORS to locate the SHORTED Device on a Circuit Trace.

http://www.electroniq.net/testers/el...t-diagram.html
http://edn.com/design/test-and-measu...acer-technique
http://electronics.stackexchange.com...ocation-on-pcb

7. Data Line problems will likely require an In Circuit Emulator (ICE) to be able to troubleshoot why the DATA is incorrect. It can also be used to READ and VERIFY the ROMS, as well as step through startup code. I've got one ordered, but it isn't here yet.

Hope this helps someone.

Thanks.

Larry

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.

http://www.electroniq.net/testers/electric-continuity-tester-circuit-diagram.html

A continuity tester for checking electrical connections, and cables can be designed using this circuit diagram.

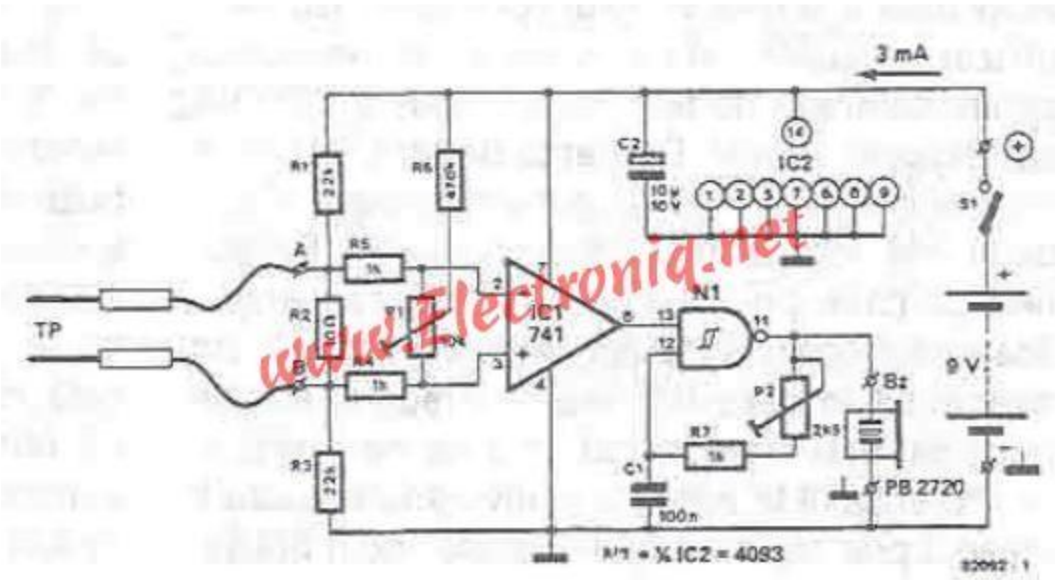
This continuity tester produces a beep when there is a short, and does not emit any sound when the circuit is interrupted, or when the resistance along the circuit exceeds 1ohm. To prevent failure of sensitive components, and for a long battery life, the tester injects a weak signal in the circuit being tested.

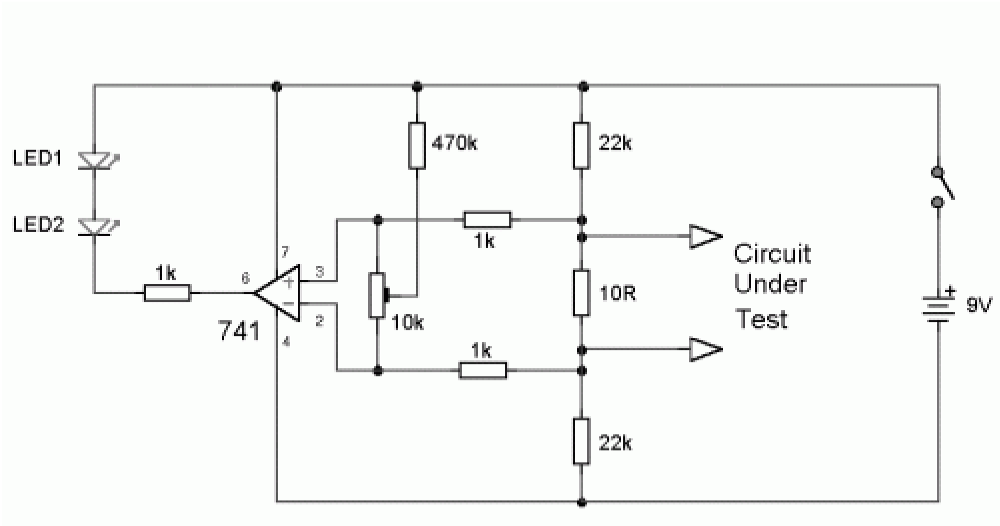
When testing circuits, there are few times when Semiconductors, Transistors, IC''s, will be tested as well as resistive devices. Furthermore, it is possible that some parts will not withstand the excessive current and / or voltage that testers inject into the circuit. For this reason a good tester will test the PN junctions with low impedance (diodes, transistors) and resistant. The tester must be sensitive enough to work with a weak signal.

Due to the large amplification of the operational amplifier 741, current and voltage for the test signal will be limited to 200 uA and 2 mV. The difference in voltage between the inverting input (pin 2) and non-inverting (pin 3) of the 741 is amplified considerably. Voltage falls out on R2 ensure that operational amplifier become negative once the inverting input has a greater potential than the non-inverting input. Non-inverting potential entry can be increased by adjusting P1, so that the non-inverting input is more positive than the inverting input. The Oscillator N1 will generate a buzzer sound. Voltage falls to R2 is the result of a good contact between terminals tester.

Frequency and volume can be adjusted with the potentiometer P2.
A value less than 1 ohm or a good contact point or a short circuit. To calibrate the continuity tester to follow the following procedure:
Put a resistance of 1 ohm (5 or 10%) between terminals, and adjust P1 so that the buzzer just about to start sounding. Remove 1 ohm resistance, and creates a short circuit between the terminals. Adjust potentiometer P2 for the proper volume. When the tester terminals are opened (or short circuit is removed), buzzer should stop sounding.

Circuit Diagram:


3 mA
IC2
IC1
741
N1
TP
A
B
9 V
PB 2720
100n
N1 = ¼ IC2 = 4093


LED1
LED2
1k
741
470k
10k
1k
1k
22k
10R
22k
Circuit
Under
Test
9V

http://edn.com/design/test-and-measurement/4328211/Quickly-find-pc-board-shorts-with-low-cost-tracer-technique

A predominant failure mechanism for production pc boards is shorted traces. Finding hidden shorts is often time-consuming and frustrating. Typical techniques of cutting traces, lifting pads, and "blowing" shorts are, at best, questionable because they may affect the reliability of the circuit, and the ever-decreasing geometries and lower voltage ICs make these practices tricky and risky. High-end, four-wire DMMs (digital multimeters) or ohmmeters, which can accurately measure the small resistance values, are expensive and sometimes not available on a designer's bench.

An inexpensive alternative approach for finding short circuits, using the concepts of four-wire DMMs and ohmmeters is simple and requires only the tools you already have on your bench and a basic understanding of Ohm's Law. This approach uses the principal that all conductors have resistance properties, and a distinct voltage drop exists between the various nodes in the shorted circuit. This approach systematically locates the nodes with lowest impendence between them and isolates the fault to two nodes.

Most digital buses have at least 1Ω over the length of the run, but a trace impedance of only 200 mΩ still has a 2-mV drop with 10-mA current applied. Most lab-grade handheld DMMs can easily resolve to 1 mV. Because you are looking for relative values, the absolute accuracy of the instrument isn't critical. However, the current must be constant to achieve repeatable results, and you must isolate its current source from the ground of the circuit under test.

A 1.5V battery in series with a 1.5-kΩ resistor is an adequate current source for this purpose. The battery provides the isolation and relatively constant voltage; select the resistor to source around 10 mA. (For lower impedance traces, such as power-supply lines, or in situations in which the DMM lacks millivolt resolution, use a higher current.) An optional clamping diode, with a cathode connected to the battery's negative terminal and an anode connected to the resistor's free end, provides protection for low-voltage logic circuits. If you use the diode, you may also need to add a power switch to keep the battery from depleting when the circuit is not in use.

A node can be any accessible part of the circuit path under test, such as a via, a pad, or a test point (Figure 1). Note the current path: When current is flowing between two nodes, a minute voltage drop occurs across the two nodes. When the current doesn't flow between two nodes, there is no voltage drop across those nodes.

To find the short in this example, put one DMM probe on any node on Trace A and the other on any node on Trace B, and note the voltage drop. In this example, if you had started with the positive probe on Node 1 and the negative probe on Node 5 and moved the negative probe to Node 6, you would note a slight voltage drop. Next, you move the probe to Node 7 and note that the voltage drop is equivalent to the voltage drop at Node 6. From this test, you can deduce that the short must exist between nodes 5 and 6 because no current flows from Node 6 to Node 7. Then, move the positive probe to Node 2 and note a small voltage drop. Continue down the line to Node 3 and note another small drop. Next, probe Node 4 and note there is no voltage drop. You can now deduce that the short must be between nodes 2 and 3 and nodes 5 and 6.

Redrawing Figure 1 with the equivalent circuit in Figure 2 makes clear how this technique works. You are now looking at a simple series network of resistors and looking for voltage drops across any resistor that has current flowing through it. When a node is outside the current path, no voltage drop occurs. By understanding the relationship of each of the vias and their position in the current path, you can systematically isolate the short by looking for lower voltage (current flowing) or higher voltage (current not flowing). When current is flowing, the short is farther from the current source. If no current is flowing, then the short is closer to the current source. This two-valued logic makes it simple to isolate the problem. The beauty of this technique is that it doesn't matter to which two nodes the current source is connected, as long as one side of the current source is connected to any node on Trace A and the other side of the current source is connected to any node on Trace B.

In this example, the short is between two node pairs, and you can isolate the short only to those pairs. A little knowledge of the board layout and common sense now come into play. You need to know only where the two traces are adjacent between nodes 5 and 6 and nodes 2 and 3, and you have found the most likely place for the short. If it is underneath a component, you have to remove the component; removing the component often removes the short. If the short is on an internal layer, you may have to do some selective cutting and jumping to isolate the short from the traces, but at least you minimize the number of cuts on the board.

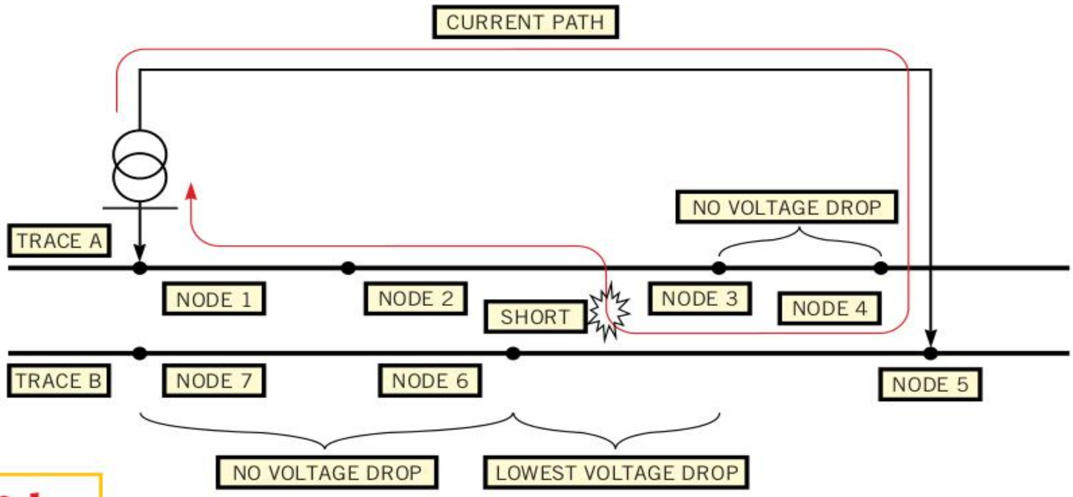


Figure 1

By applying a fixed current to various nodes and looking at the resultant voltage drops, you can home in on the likely location of a pc-board short circuit.

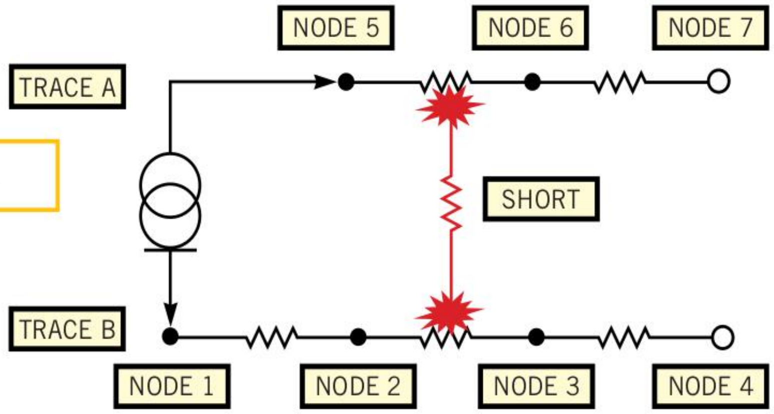


Figure 2

The equivalent circuit of the pc-board layout shows the principal of the source-and-probe technique.

http://www.eetimes.com/author.asp?section_id=30&doc_id=1284424

Determining the exact physical location of a short circuit in a printed circuit board can be fiendishly difficult. The millivolt measurement method described can help.

A few weeks ago I was asked to look at a telecom PCB fresh out of production with a small problem--the inventory EEPROM would not load the card revision information, and a new EEPROM did not solve the problem. I hunted up a card extender, blew off the dust, plugged it into one of our test racks, and set the system control card to repeatedly query the EPROM. I suspected the problem was one of the glue logic devices controlling the EEPROM, but rather than recommend a shotgun replacement of all the glue logic I decided to take a closer look first.

Good thing, shot-gunning would not have worked.

First clue – the scope showed the EEPROM serial data pin was stuck at a logic high (Vdd power supply) voltage in spite of the control and clock signals to the EPROM looking normal. Since the EPROM had already been replaced, was there a short from this pin to Vdd?

Yes, the EEPROM output did measure about 0.1 ohms to Vdd on a DVM. This 0.1 ohms is the normal reading of the probe wires, which sets the low-resistance measurement limit of a DVM, so obviously there was a 'dead short' (approaching 0 ohms) somewhere along the length of the trace that was connecting the output pin to a few other logic device inputs and tri-state outputs. Maybe one of them was shorted to Vdd inside a logic device?

Unlikely, since a defective device input or output when shorted to Vdd or ground usually has a least a few ohms of measured resistance. But just to make sure, I lifted all the IC pins that connected to the trace, including the new EPROM output. This is very easy to do with surface mount devices, wick the solder off and pry the pin off the pad slightly with sharp tweezers while hot under the soldering iron. Careful, don't overheat and lift the copper pad off the board!

With all the pins lifted off the trace I could still measure a dead short between the trace to Vdd. This exonerated all the attached devices, the short was hidden on the pcb somewhere along the branching run of that trace. Visual inspection under a microscope showed nothing obvious. Perhaps it was a solder bridge lurking where an external portion of the trace ran underneath an IC package.Or perhaps was invisible inside the pcb.

The short could be anywhere along the branches of the trace, and the limitations of the DVM in ohmmeter mode made it useless for resolving below 100 milliohms. Now it was time to get out the heavy artillery. Not a current probe, you cannot clamp a current probe around a pcb trace. But with the following procedure the physical location of a dead short is relatively easy to locate.

The trick is to apply a controlled current from a current-limited lab power supply through the short, in this case between the shorted trace and Vdd, and look where the current is going. The current flow will be directly through the short and the path leading to that short, but will not flow along any other path of the isolated trace (all logic device pins still lifted off their pads). Of course we cannot actually see where the current flows, but we can measure the result of the current flow to get that information.

The copper trace is not a perfect conductor, it has many milliohms of resistance depending on its thickness, width, and length. Forcing 1 amp through the short will produce 1 millivolt drop for every milliohm of trace resistance (Ohm's Law from Electronics 101). Even the cheapest modern DVM from the hardware store can resolve down to 0.1 millivolt, plenty good enough for this. The current level should be adjusted for as low as possible to still get meaningful voltage readings, too much current can burn open a very thin trace. On the other hand, if trying to locate a short that is between the power planes, 2 or 3 amps may be needed due to the much lower resistance of power planes.

Even when traces are buried on internal layers, measurement points are where the trace is brought to outer layers through vias. It helps if you know (from the pcb design artwork) or can deduce the internal paths of the trace layout.

Inject the current – find convenient places to solder wires to the shorted trace and Vdd. It doesn't really matter where the current starts, it will eventually end up at the short. A good place to solder a wire to Vdd is at one end of a decoupling capacitor. Solder a second Vdd wire to another point nearby. Another wire soldered to a via or component of the shorted trace, and you're in business.

Preset a lab power supply to about 0.4 volts open circuit, then short the power supply output and set the current limit to 200 mA. You can always increase the current later if needed. 0.4 volts is a safe starting point in the event that you overlooked a device and did not lift its pin from the trace; 0.4 volts will not bias input clamp diodes into conduction should the short suddenly disappear. Then connect the power supply to the trace and one of the Vdd test wires soldered onto the pcb.

Set the DVM to the millivolt range and connect one probe to the other Vdd wire, this avoids measuring voltage dropped along the Vdd wire from the supply. Touch the other DVM probe to the wire that is soldered to the trace. You will read several millivolts, this is the voltage dropped along the trace as the current flows to the short.

Now move the trace probe to the next access point (via or IC pad) and note the measured voltage drop. If the drop reduces, you have moved the probe closer to the short. If the drop does not change, there is no current flowing along the branch whose two points you have just measured.

Figure 1 below depicts a typical pcb trace routing, portions of the trace can be on inner or external layers, vias bring inner layer trace portions to the external layers for connection to surface mounted device pads.   For clarity most devices are not shown. Wires to the current-limited lab power supply and DVM can be soldered to either vias or components that are connected to the shorted trace and Vdd.

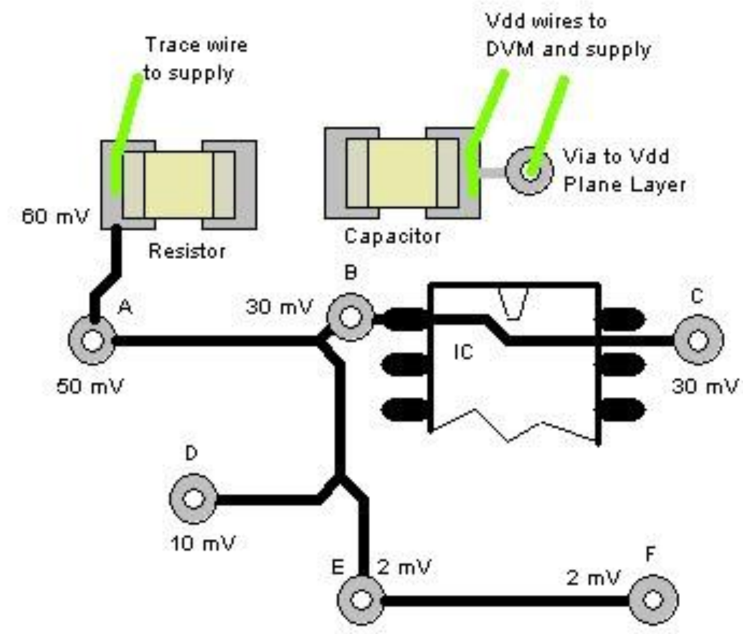


Figure 1: Typical trace route and via measurement points

With one DVM probe connected to Vdd through the connected wire, move the other probe around to the vias of the shorted trace. Note the voltage readings at each measurement point.

Starting at the resistor 60 mV is measured where the supply wire is attached. Via A measures 50 mV. Thus 10 mV is dropped along the trace between the resistor and via A, indicating that current is flowing through this portion of the trace on its way to the short.

Via B measures 30 mV. If the trace is already on a surface layer and via B is not part of the trace, pin 1 of the IC is a good point to measure instead. Hmm, the trace runs underneath the IC on it's way to via C. Could there be a solder bridge between the trace and where it runs between the IC pins? Via C measures the same 30 mV as via B or IC pin 1, obviously no voltage is dropped along this portion. Thus no current flows here, so no short under the IC.

Via D is 10 mV (getting closer!), but via E is even lower at 2 mV. The current must be flowing to via E. The 10 mV at via D happens because the trace forms a resistive voltage divider at the junction between the branch to via D and the section between via B and via E. The same effect causes vias B and C to measure 30 mV.

Via F is the same 2 mV as via E. No current is flowing along the trace between via E and via F, so the short is not at or beyond via F.

That leaves via E as the prime suspect. The current is draining into via E and no farther.

In my case, I had the luxury of an X-ray machine to see what might be lurking inside the pcb. Figure 2 below confirms the culprit--deep inside the pcb there is a small conductive particle touching the copper barrel of the via and shorting it to the internal Vdd plane layer near the upper center of the image. The thick object running to the left is a wire that I soldered to the via to confirm it was the location of the short prior to making the X-ray image. The cure was to use a small drill bit in a pin vise to drill out the via barrel and jumper the trace connections around the via using AWG 30 Kynar insulated wire.

Even without an X-ray machine, knowing the physical location of the short and probable cause enables one can drill out vias or cut external layer traces to isolate the short.

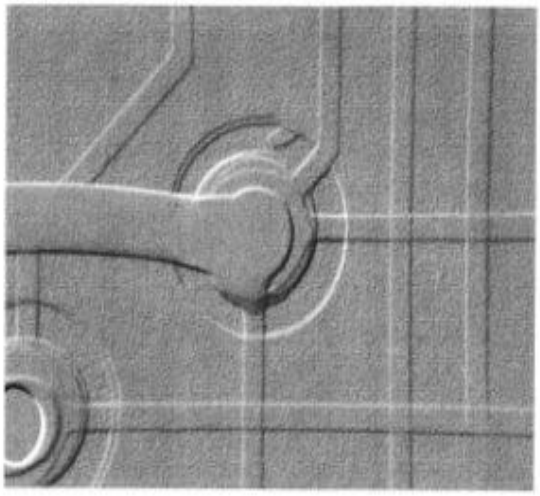
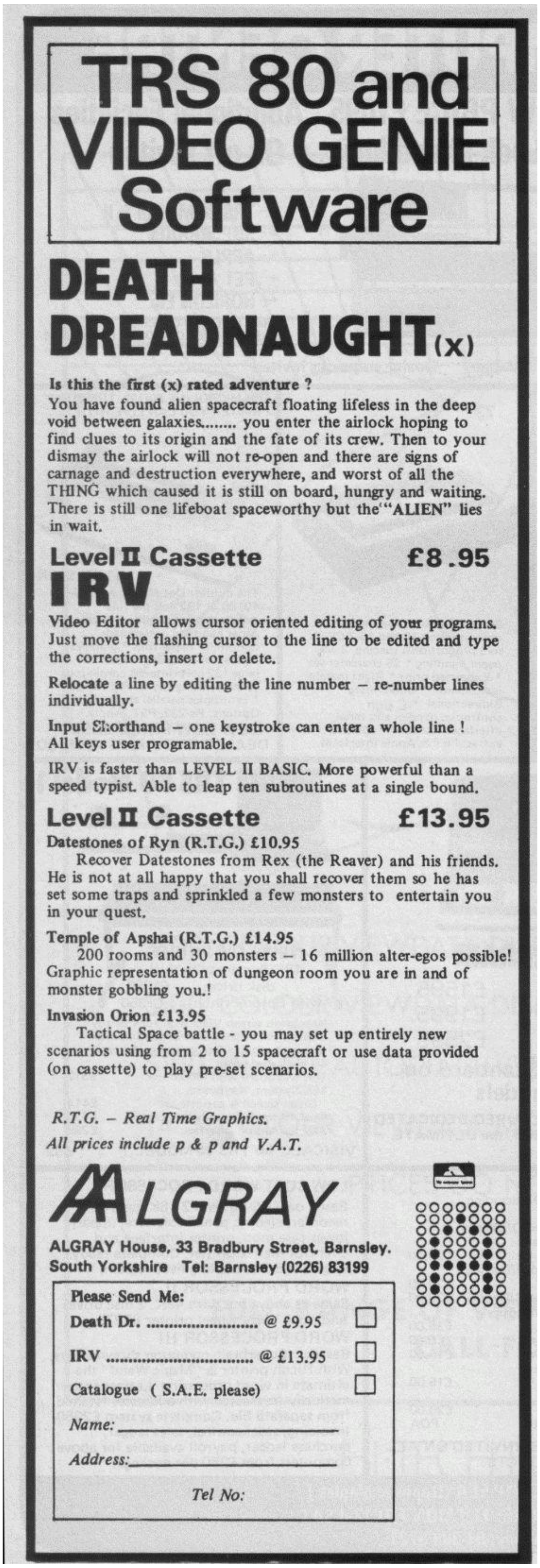


Figure 2: X-ray image of a pcb internal short. A small conductive particle upper center protrudes from an internal plane layer and contacts the barrel of the via at center. The wide object running to the left is a wire that was soldered to the via for final confirmation of the short location prior to making this X-ray image. (Courtesy of ICBS / EO Networks Division)

Glen Chenier is a design consultant based in Allen TX.

--== o0o ==--



Business Time with Kev

Well it's sometimes a sad fact that unpacking my rather voluminous computer collection has at times become a little overwhelming but the majestic heavyweights of the TRS-80 world, the Model 16's, have percolated to the surface and been reunited with their keyboards.

For those who viewed the photos in my last article showing my new facility for housing, running and enjoying my wonderful collection, imagine that now looking like the back room in a charity shop just after the collection truck has been unloaded (my apologies to any charity shop workers and operators who may be reading this - you all do a very wonderful job). I suppose metaphorically speaking a bomb has hit the place. But that's an essential part of getting it all unpacked, placed properly and being accessible for later pleasures. But this should be a lesson to all of us - when we moved from Adelaide (South Australia) to Melbourne (Victoria) we (the removalists) brought 84 cubic metres with us – my wife says 45 of that was my old computer stuff but I can't argue with her for two reasons – one is she's my wife and the other is she's right despite that.

So despite it all being packed away I continued to collect more stuff (my wife said I had enough but she can't be right about everything). My estimates of being operational by October 2014 are in a bucket again and I think quite a few more months are required. Somehow I think Moore's Law kind of applies in principle to classic computer collecting.

Now just to refresh memories I have two Model 16's (that's kind of correct but I'll expand more in my next article). I'm going to talk a little bit about the second one I acquired first – this won't be a long article because I'm still not in a position to fire this thing up. This one has not been fired up since I got it so there's no telling what shape it might be in internally. There's another very good reason to look inside first before the power goes on apart from any pre-emptive strikes on the power supply caps.
This particular machine came from a bakery in Maitland on the Yorke Peninsula of South Australia (see the map for our non-Australian readers

and I seem to recall it was Fenwick's Bakery [but don't quote me on that] for any South Australian readers) so I'm thinking lots of flour and various other types of powdery bakery dusty stuff that has possibly found its way into all sorts of nooks and crannies and is ready to overheat something or maybe even ignite!

Now for collectors it's nice to know a little bit of the history of any particular item so knowing that this machine came from a bakery in Maitland is kind of a nice story to know and tell people. But it's actually extremely significant but I won't expand on why until I introduce my other Model 16 in the next issue because this little bit of trivia then suddenly becomes hugely important, particularly given that the two acquisitions were entirely independent of each other by way of time and distance.

My second heavyweight acquisition was a Model 16b (Radio Shack Cat No: 26-6004). A search of IRA Goldklang's RS Cat Number Search shows that to be a single disk Model 16b. From the photo below you can see that this unit has an internal hard disk drive (referenced by the OS as Drive 4). The 8" floppy on the right (referenced by the OS as Drive 0) is of the Thinline type (think of it as like a half-height drive). The 8" machines shipped with three types of floppy drives: the Push-button, the Latch and the Thinline. They all served the same purpose of getting data on and off discs but had some different configuration requirements that were looked after in the operating system. The particular differences will probably make an article for a later date. You might also see from the photo below that the floppy drive has been stored with its storage and shipping card in place – that's how I got it and all bar a few minutes it's been like that the whole time so hopefully the floppy drive is still in good order. As with most early TRS-80 hard drives I fully expect the hard drive to be non-operational but sooner or later I'm going to have to find out for certain.

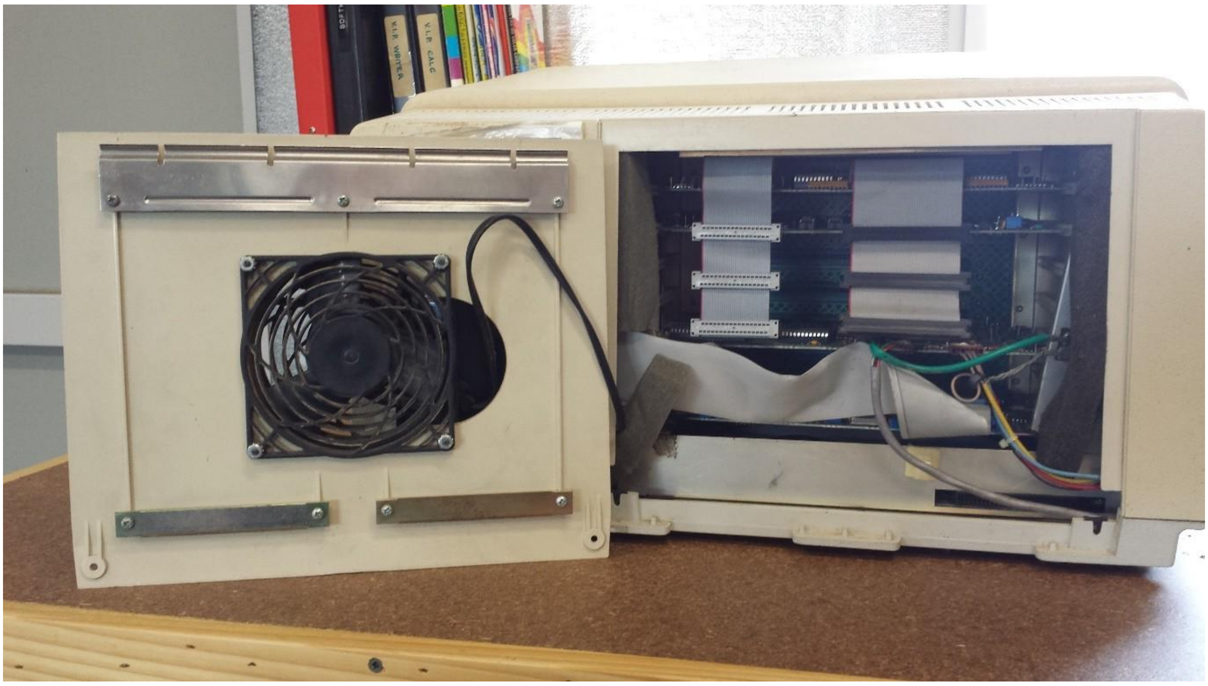
So it looks like when this was originally shipped it had a lonely single 8" floppy drive and at least 384KB of RAM being a Model 16b. The catalogue price at the time was $4,999 (that would be US dollars). Today that would be about $11,430 (a little tip for collectors – if you have the catalogue price of an item you can calculate its cost today [I used US Dollars as that was the catalogue price used for this] using Wolfram Alpha -http://www.wolframalpha.com/input/?i=US%244999+%281984+US+dollars%29 – or you can punch it straight into the search line in Wolfram Alpha like this US$4999 (1984 US dollars) - of course substitute your own dollars and year and I have not tested other currencies). For this article I have assumed a purchase year of 1984. On the back I found this little sticker.

Another search of Ira's cat numbers for 26-4154 showed that this is a "Model II/12/16/6000 - Accessory - 15 Meg Hard Disk Kit – Primary". The catalogue price at the time was $2,495 (today $5,703.51). These machines did ship already fitted with a (15MB) hard disk drive if so ordered as catalogue number 26-6006 at a price of $6,999 (today $16,000) – I don't know about you but in the early 1980's that was quite a frightening amount of money I suspect.

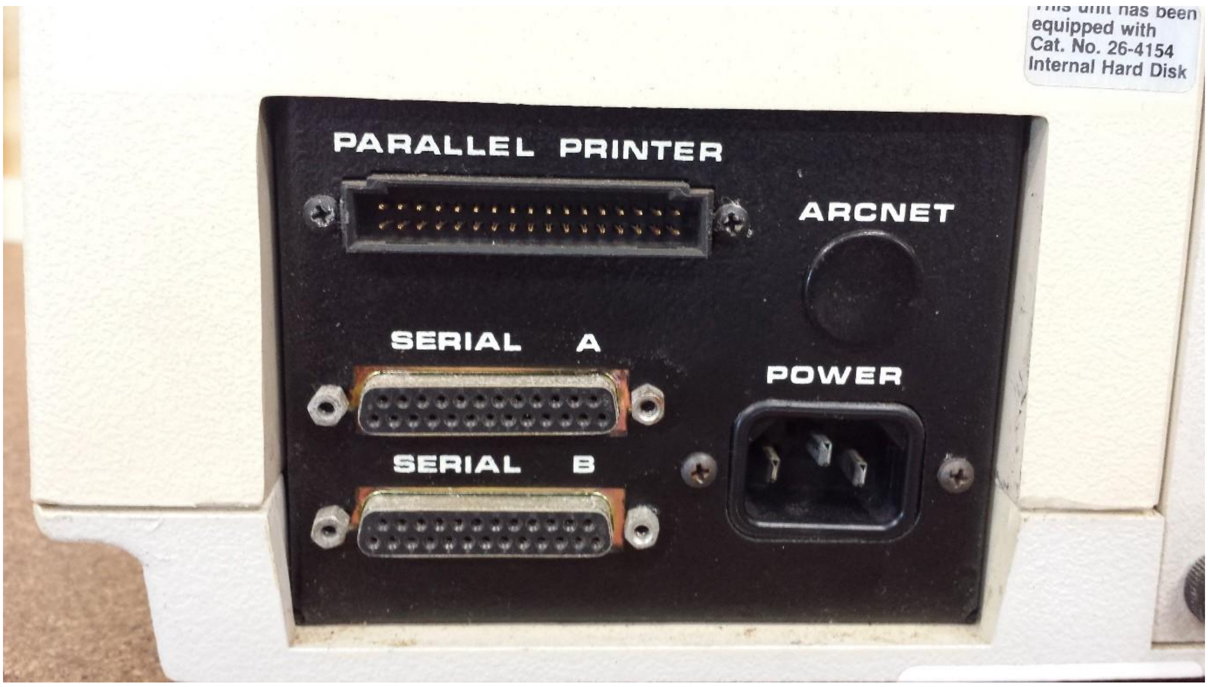
With a total investment in this machine at the time of $7,494 as opposed to buying it outright with a HDD installed at $6,999 you'd might assume that the fitting of a HDD for this machine was an afterthought. Of course this is guesswork – maybe at the time Radio Shack couldn't supply a 26-6006 so they fitted a 26-6004 with a HDD for them or maybe this was just the way they shipped a 26-6006 at the time in Australia - we most probably will never know.

Externally the unit is quite clean, particularly compared to my other

Model 16. I did open the back to look inside the cage (see the photo below) but ventured no further at this stage so there's been no particular investigation of the cards or other internals but it all looks relatively standard. Yes that's a big fan in the back and the cage is particularly insulated to ensure air passes through the cage and not around it. The two bars at the bottom of the fan door are cable strain stays. You can fit these machines with additional floppy and hard disk drives (the ports for doing that are accessed from inside the cage) and you fix the cables under these bars to take any strain on the cables.



The back also supports a parallel port and two serial connections. The serial connections, while being RS-232C ports, have some differences. Serial A allows asynchronous and synchronous transmission while Serial B only allows asynchronous transmission.



Three operating systems were on offer from Radio Shack at the time TRS-XENIX, TRSDOS-II and TRSDOS 2.0b. If the hard drive on this particular machine is still viable it will be interesting to see what is installed on it (if anything) and what other software may be there too. Hopefully it might tell me a bit more about the machine's origins and use.

Floppy discs could store about 600KB for a single sided and 1.2MB for a double sided disc. I do have several hundred 8" floppy discs and some of them are tending to go the way of the early hard drives of course with the surface looking like a cracked and parched desert – it's a common problem. Fortunately lots of stuff I have has been relatively well stored so it may still be OK but that's part of the fun in getting this back up and running. Anything that is still in its shrink wrap is gold of course.
The monochrome screen displays 80 x 40 characters – green on black or black on green (reversed). The machine can display the "full ASCII set" of characters and 32 graphic characters.

For input/output processing the machine uses the ever faithful Z80A and for computational processing uses the 68000.
Hard drive specs are as follows:

Cylinders per track	306
Tracks per unit	1,836
Tracks per platter	612
Sectors per track	34
Bytes per sector	256
Average latency	8.3msec
Rotational speed	3,600rpm
Recording density	9,625bpi
Storage unformatted	19MB
Track density	345tpi
Storage formatted (Primary drive)	15.5MB
Storage formatted (Secondary drive)	15.9MB*

* I suppose 400KB back in those days was a huge amount of space and worthy of getting a mention in the specs even though it was less that the space on a single sided floppy. You could fit a remarkable amount of stuff in 400KB back in 1984.
Well I hope that was of interest to TRS8BIT readers. Next article I hope to give the other Model 16 similar treatment and join some dots on the history of these two gentle giants.

KEVIN PARKER

ASK MAV
ABOUT THE TRS-80 MODEL 4P PORTABLE COMPUTER
by Ian Mavric

1.
Q. Compared to a present-day laptop, the 4P looks weird, was that as portable as they could make a computer in 1983?
A. For a short period between 1981 and 1984, computer companies produced what are termed "luggable" computers, and the 4P was one of these. Others similar in design include the Osborne 1 and Executive, Kaypro II-to-16 series, and the IBM Portable PC 5155. All feature small CRT monitors, dual disk drives, full keyboards, and are mains operated. They all approximate the size of a portable sewing machine, but to my mind weigh slightly more. The 4P was one of the later ones and is a few pounds lighter than Kaypros and the IBM. These 'portable' computers could be lugged from site to site, and used anywhere AC power was available. By 1985 LCD and battery technology had improved in performance and decreased in price so much so that these luggable computers were seen as obsolete and needlessly heavy. Once computers like the TRS-80 Model 100 showed the way forward with a small, lightweight LCD and battery-based notebook which could be used on one's lap, development of luggables not so much slowed down but halted completely. So did sales.

However I disagree about the perception that luggables were as weird as people say they were. The 4P and its cousins were fully capable systems running the same operating systems (TRSDOS and CP/M for the 4P, CP/M for the Osborne and Kaypro, and MS-DOS for the IBM 5155) and had the same performance and storage as a desktop computer. It would take another decade for laptop/notebook computers to catch up with desktops as far as performance goes.

Pictures: Kaypo (left) and Osbourne (right) are similar luggables to the 4P

The buying public made their preferences felt when Tandy released the 4P and the Model 100 at the same time. The 4P lasted only two years on the market whereas the Model 100 and its successors (the Tandy 102 and 200) continued to sell well for 7 or 8 years. This despite the fact that by the time you outfitted your Model 100 with a Disk-Video interface and a monitor it cost more than a 4P and still had less power.

2.
Q. My 4P has yellowed so much it looks like aged cheese. Can I do anything about it?
A. This happens with some white Tandy computers and not others. The Model 4 is made of the same high-impact ABS plastic but Tandy went one step further and painted the plastic case with textured white paint. This paint doesn't go yellow but its possible for the base can go yellow, as the base isn't painted, only the top.

Retr0brite (http://retr0bright.wikispaces.com/), which is a chemical process which restores computer plastics back to their original colour. It has been used successfully on other popular systems such as the C64 and Apple IIe. It has been used on the 4P with good success however the quantities of chemicals required can be costly as the 4P case is positively enormous compared to a Commodore 64.

People have also painted their 4Ps which has given them a pleasant appearance, just remember these computers were off-white when new, as opposed to "paper" white.

Picture: a yellowed Model 4P

3.
Q. What revisions of the 4P did they release?
A. The 4P came in two versions, the original and the gate array. The original has the catalogue number 26-1080 on it and has a non-gate array motherboard, traditional TRS-80 keyboard with up-down arrows on the to the left of the A and Q keys, and the left-right arrows next to the @ key. It also uses metal latches to hold the front cover on. On USDM machines they have a B&W CRT, though the rest of the world got a green CRT. The 2nd 4P has the catalogue number of 26-1080A and it has a re-designed gate-array motherboard, the keyboard has a cluster of arrows, and the front cover is held on with rubberised plastic latches. All of these 2nd generation 4Ps use green CRTs regardless of which country they were sold in.

Pictures: 4P keyboards - early (top) and late (bottom)

As far as performance goes they are equivalent. If you play a lot of old TRS-80 games then the arrow key arrangement on earlier systems makes it slightly more friendly to use. Other than that they are more or less the same computer to use.

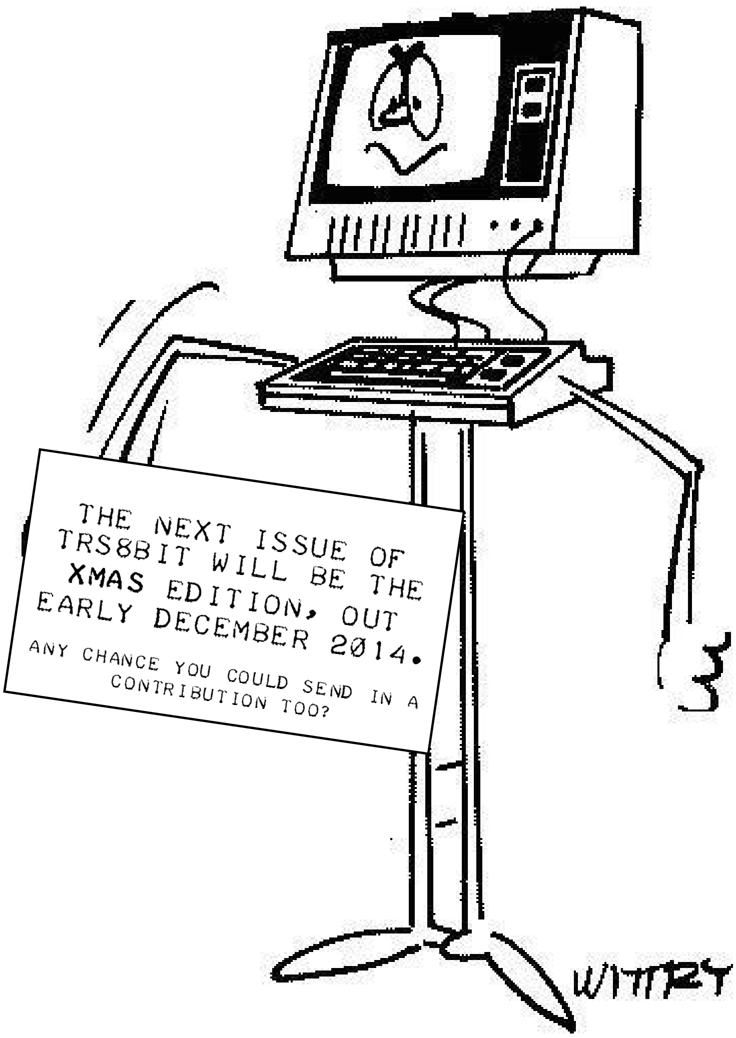
4.
Q. I heard Tandy included a hard disk auto-boot routine in the 4P BIOS - is this true?
A. When Tandy designed the 4P the team responsible for the BIOS thought it would be neat to have the 4P auto-boot from a TRS-80 5Mb hard disk drive without a hard disk start-up boot floppy. Previously all TRS-80s required a boot floppy and so losing the boot floppy was considered a good thing. However the method used required the hard drive to be set up with a very specific set of hard drive parameters in order for it to work, particularly in relation to the first partition of the hard drive. Also, necessary patches to the hard disk operating system were easy to flub, resulting in loss of information on drive. By the time the 4P was ready for release, Tandy has discontinued the 5Mb hard drive and was selling 10Mb, 12Mb and within a year, 15Mb hard drives, all of which were basically incompatible with the 4Ps auto-boot method. A decision at Tandy was to leave the routine undocumented.

Eventually news of the undocumented boot routine surfaced and some clever people worked out how to set up the system so that it does indeed boot from a hard drive. For those interested they should check TRS-Link issue #3, TMQ Spring 1990 Volume IV.iii page 19, and Bits&Bytes Issue 100, pages 7-9. Naturally since a FreHD emulates a TRS-80 hard drive, the patches used to turn a real TRS-80 hard drive into an auto-boot hard drive can also be used with a FreHD - no internal 4P EPROM modification needed.

5.
Q. I'd like to add double-sided disk drives to my 4P - is it hard?
A. It's fairly easy modification to make as the FDC on the 4P motherboard is already capable of addressing double sided disk drives. What you need is a pair of drives, such as Teac FD55-BR, which are 360K double-sided double-density black faced disk drives. (You find them in Tandy 1000s.) Also, need a cable which has all the teeth in place. Tandy uses a cable with teeth pulled for drive select, but using double sided drives need all teeth in place. Next you need to set the drive number but setting the drive-select jumper on each disk drive. DS0 will be the booting drive and DS1 the data drive. Typically one installs the DS0 drive closest to the CRT. Finally you need to drill new holes in the sheet metal assembly which holds the disk drives. The type of disk drive chosen by Tandy has the drive mounted about 0.5cm more forward than standard drives, so you need to drill new holes so the replacement drives fit flush with the front bezel of the 4P.

Once completed, verify the systems works properly still with normal single-sided operating systems, then read the manual of your favourite operating system about how to format double sided bootable disks, are you are set.

ianm@trs-80.com



TRS-80.ORG.UK

COMPETITION

Mav came up with the great idea for a competition to see who can design the best Dr Who graphical re-creation of its iconic title sequence on a TRS-80 Model I/III, II/4/12/16/6000, or Coco 1-2-3.

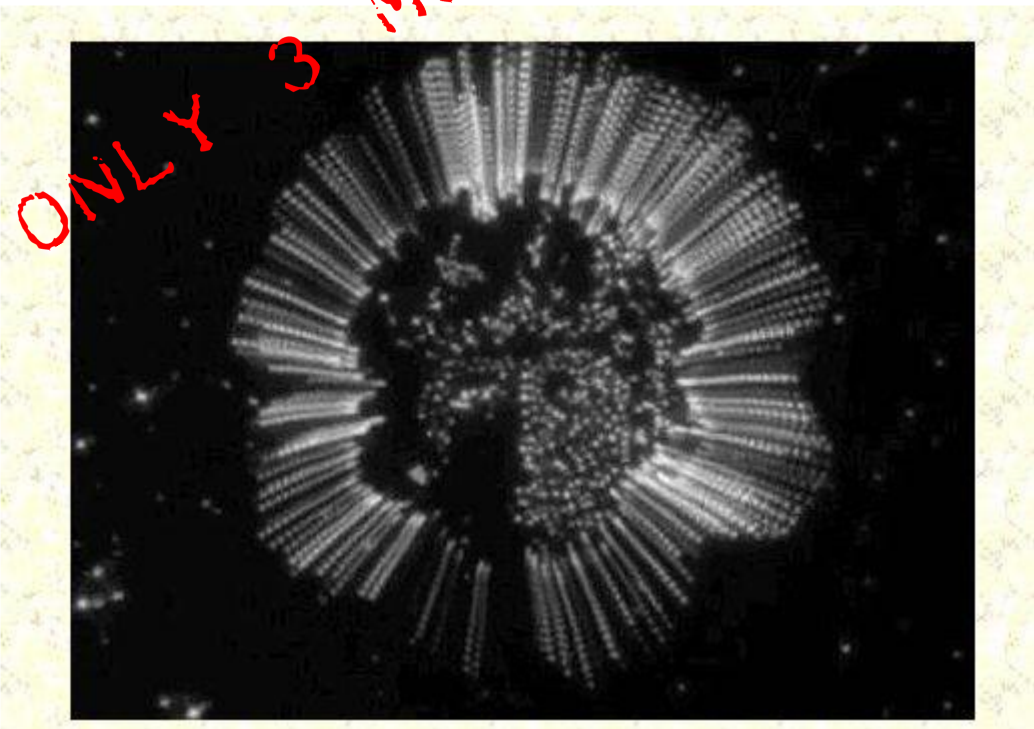
Chose any Dr Who title sequence you like, and upload a video to Youtube and email your program entry to me, (dustym @ fabsitesuk.com).

Dee will judge the entries, (her decision will be final and not necessarily made on programing merit), which must be received by the **1st December 2014, 21:00 hrs GMT**. The **prize**, being offered by Mav, **is a FreHD**, **kit B**, which will be delivered during January 2015.

So, you have 6 months to complete and submit your masterpiece. Language and memory requirements are open, as are hardware and memory requirements.

Do you have a Hi-Res Model 4 with 128K memory and can program in Z80 assembly language? Then a killer Tom Baker intro would look incredible.

ONLY 3 MONTHS LEFT



USEFUL PROGRAM FROM THE PAST... THAT IS STILL USEFUL TODAY

PART 1: JOHN SCHEER'S MODEL 4 CRT SCREEN SAVER

Presented by Ian Mavric

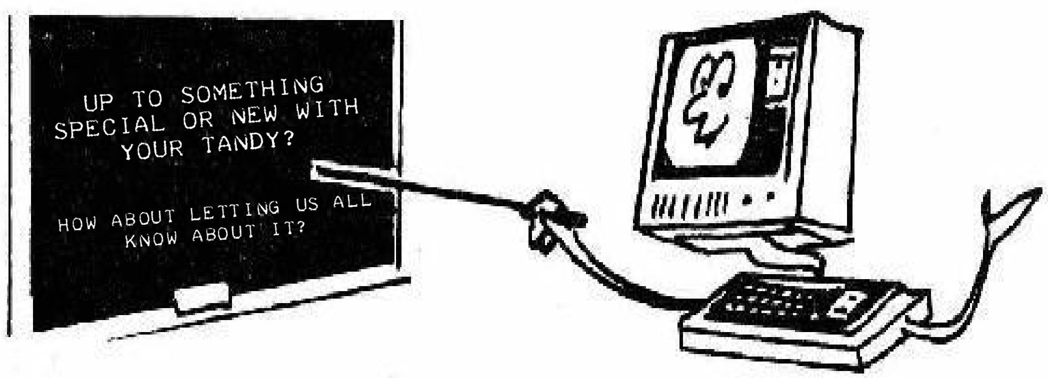
This new series brings back great old programs from defunct magazines from back in the day. Magazines like 80micro, Computer User, Computer News 80, Micro-80 and 80-US had program listings for useful utilities. Each issue I, or another regular TRS8Bit columnist, will present a program that we used back in the day, and continue to use today.

Today I present: CRT Screen Saver program by John Scheer from 80micro, October 1987, pages 75-79.

In order to keep these old machines running we need to preserve our CRTs and John Scheer provided a great screen saver program in October 1987 which did exactly that. While screen savers are commonplace these days and in fact were commonly installed on PCs in 1987, one for a TRS-80 was a novel idea. I've been using it for 30 years and have only found a couple of programs which don't work with it (LeScript and Super-Scripsit).

If your Model 4 or 4P is on for hours or days at a time like mine are, you should really benefit from using this program to make sure your CRT lasts as long as the rest of your TRS-80 does.

ianm@trs-80.com



by Jon Scheer

CRT Saver

Protect your video screen from burn-in.

If you leave a static image on your screen for a long period of time, it can burn into the phosphor and leave a permanent shadow. If you must abandon the computer for any length of time, it is desirable to turn the screen off.

My CRT Saver (see the Program Listing) automatically clears the Model 4's screen if you don't display a character for three minutes. The next character you send to the display restores the screen contents. The program has three parts: a low-priority background task, a video filter, and a routine to install the task and filter.

The task uses a counter (DELAY) to clear the screen. Every time the task is called, the counter is decremented. Once the counter reaches zero, the contents of the screen are copied to a buffer in memory. The filter restores the contents of the screen and is attached to the output device *DO. The filter is called whenever you send a character to the screen. Each time the filter is called, it restores the screen, if necessary, and it resets the tasks counter.

The initialization routine installs the task and filter after copying them into high memory. (This routine is based on several Hardin Brothers' filter-installation routines.) This routine also sets up a short routine to display a message when you reboot the system. This second routine is used only if you sysgened your boot disk after installing the task.

The Task

To determine the value the counter starts with, calculate the number of times it will be called in three minutes. The task is installed in slot 2, which is called every 266.67 milliseconds. Therefore, the routine is called 255 times a minute, or 765 times every three minutes.

When the filter is called, it resets the tasks counter to 765. The filter should execute quickly, and it is faster for the filter to set a flag (OUTFLG) that tells the task to reset the counter. Now when the task is called, it first checks if the counter should be reset. After the counter is decremented, if it is zero, the task copies the screen contents to a buffer (SCRBUF), clears the screen, and turns off the cursor. A problem occurs when the task tries to accomplish these last two things. Both of these operations require special characters to be sent to the screen, thus calling the filter. To overcome this problem, a flag, IGNOR, tells the filter to ignore these characters and not to reset OUTFLG. Once you save the screen, the task resets IGNOR. Another flag, SCROFF, tells the filter that the screen was saved. This way the filter knows if it should restore the screen before displaying the next character to the screen.

The Filter

When the filter is called, it checks IGNOR to see if it should ignore setting OUTFLG. If IGNOR is not set, the filter looks at SCROFF to determine if the screen was saved. If it was, it copies the buffer to the screen, turns on the cursor, and resets SCROFF. Next, the filter sets OUTFLG, and the current character appears on the screen.

For the task and the filter routines to synchronize with each other, they must share the following variables: SCRBUF, the buffer to store the screen contents in, and three flags; OUTFLG, the counter-re-

System Requirements

Model 4
64K RAM
Assembly language
Editor/assembler
Available on The Disk Series

Program Listing. A filter that clears CRT screen to avoid phosphor burns.

```
;
;          CRTSAV -- CRT Saver.  This filter will 'save'
;          the CRT screen by turning it off (copying it
;          to a buffer, erasing the screen, and turning
;          off the cursor) if a character is not received
;          after 3 minutes.  Once a character is sent, the
;          filter will turn back on the screen (copy from
;          the buffer, and turn on the cursor).
;
;          By Jon Scheer  (C) 1986.
;          Version 1.6
;
VERNUM    EQU      16H                 ; Version 1.6.
;
;          System calls:
;
ADTSK     EQU      1DH                 ; Add task.
CHNIO     EQU      14H                 ; Chain I/O routine.
CKTSK     EQU      1CH                 ; Check slot for task.
DSP       EQU      02H                 ; Display a character.
DSPLY     EQU      0AH                 ; Display a line.
EXIT      EQU      16H                 ; Exit program.
FLAGS     EQU      65H                 ; Get system flags.
GTMOD     EQU      53H                 ; Get module address.
HIGH      EQU      64H                 ; Get/Alter HIGH$.
VDCTL     EQU      0FH                 ; Video control.
;
;          ASCII equates:
;
LF        EQU      0AH                 ; Line feed.
CR        EQU      0DH                 ; Carriage return.
SO        EQU      0EH                 ; Turn on cursor.
SI        EQU      0FH                 ; Turn off cursor.
FS        EQU      1CH                 ; Home cursor.
US        EQU      1FH                 ; Clear to End-Of-Screen.
;
;          Misc equates:
;
DELVAL    EQU      02A3H               ; 3 minutes (3 * 225).
SLOT      EQU      2                   ; Task goes in slot # 2.
TRUE      EQU      01H
FALSE     EQU      00H
;
;          Start of program:
;
;          Filter header:
;
          ORG      3000H
FLTBEG:   JR       START               ; Branch to start of flt.
OLDHI:    DEFW     $-$                 ; Used for old HIGH$.
          DEFB     6                   ; Length of module name.
          DEFM     'CRTSAV'            ; Module name.
MODDCB:   DEFW     $-$                 ; Used for DCB address.
          DEFW     $-$                 ; Reserved by TRSDOS.
;
;          Start of filter code:
;
START:    JR       NZ,FLT01            ; Jump if not PUT.
          DI                           ; Disable interupts.
```

Listing continued

Listing continued

```
00690          PUSH   AF               ; Save A & flags.
00700          LD     A,(IGNOR)        ; Load ignor flag.
00710 REL00    EQU    $-2
00720          CP     TRUE             ; Flag set?
00730          JR     Z,FLT00          ; Jump if ignor true.
00740          LD     A,TRUE           ; Load value.
00750          LD     (OUTFLG),A       ; Set output flag.
00760 REL01    EQU    $-2
00770          LD     A,(SCROFF)       ; Get screen status.
00780 REL02    EQU    $-2
00790          CP     TRUE             ; Set flags.
00800          JR     NZ,FLT00         ; Jump if not off.
00810          CALL   RESSCR           ; Restore screen.
00820 REL03    EQU    $-2
00830          PUSH   IX               ; Save IX.
00840          LD     IX,(MODDCB)      ; Get DCB vector.
00850 REL04    EQU    $-2
00860          LD     A,CHNIO          ; Set up call.
00870          PUSH   BC               ; Save BC.
00880          LD     C,SO             ; Turn on cursor char.
00890          RST    28H              ; Output char.
00900          POP    BC               ; Restore BC.
00910          POP    IX               ; Restore IX.
00920          LD     A,FALSE          ; Load value.
00930          LD     (SCROFF),A       ; Save new status.
00940 REL05    EQU    $-2
00950 FLT00:   POP    AF               ; Restore A & flags.
00960          EI                      ; Enable interupts.
00970 FLT01:   PUSH   IX               ; Save IX.
00980          LD     IX,(MODDCB)      ; Get DCB vector.
00990 REL06    EQU    $-2
01000          LD     A,CHNIO          ; Set up call.
01010          RST    28H              ; Chain to DCB.
01020          POP    IX               ; Restore IX.
01030          RET                     ; Return.
01040 ;
01050 ;        Interupt routine (task):
01060 ;
01070 TCB:     DEFW   INTBEG           ; Point to start of task.
01080 RELTCB   EQU    $-2
01090 REL10    EQU    $-2
01100 INTBEG:  LD     A,(OUTFLG)       ; Get output flag.
01110 REL11    EQU    $-2
```

Listing continued

set flag, SCROFF, the CRT-saved flag, and IGNOR, the ignore-character flag.

Installation

The program installs easily. After compiling the program to CRTSAV/FLT, issue the following TRSDOS commands:

SET *CS CRTSAV/FLT
FILTER *DO *CS.

Once the filter is installed initially, if you issue the TRSDOS command Sysgen, it automatically installs when you boot your system. When the disk boots, you receive a short message informing you that the filter is installed.

I have been using the filter for several months and have only encountered one problem. When using Scripsit, the screen is saved once after three minutes, regardless of whether I type anything. (Apparently, Scripsit uses its own routines to access the screen memory.) Pressing any key restores the screen. I don't know if the same problem occurs with Superscripsit. Except for that one problem, the filter works well. I've left my machine on for weeks at a time.■

John Scheer is a computer programmer for Science Applications International Corp. You can write him at 822 Lincoln Rd., Apt. 201, Bellevue, NE 68005.

Listing continued

```
        CP      TRUE            ; Was char displayed?
        JR      NZ,INT00        ; Jump if not.
        LD      HL,DELVAL       ; Load delay value.
        LD      (DELAY),HL      ; Save new delay.
REL12   EQU     $-2
        LD      A,FALSE         ; Load value.
        LD      (OUTFLG),A      ; Reset output flag.
REL13   EQU     $-2
INT00:  LD      HL,(DELAY)      ; Get delay.
REL14   EQU     $-2
        LD      A,H             ; Get high byte.
        OR      A               ; Set flags.
        JR      NZ,INT01        ; Jump if not 0.
        LD      A,L             ; Get low byte.
        OR      A               ; Set flags.
        JR      Z,INTEXI        ; Jump if 0.
INT01:  DEC     HL              ; HL = HL - 1.
        LD      (DELAY),HL      ; Save new value.
REL15   EQU     $-2
        LD      A,H             ; Get high byte.
        OR      A               ; Set flags.
        JR      NZ,INTEXI       ; Jump if not 0.
        LD      A,L             ; Get low byte.
        OR      A               ; Set flags.
        JR      NZ,INTEXI       ; Jump if not 0.
        CALL    SAVSCR          ; Save screen.
REL16   EQU     $-2
        LD      A,TRUE          ; Load value.
        LD      (SCROFF),A      ; Set screen-off flag.
REL17   EQU     $-2
INTEXI: RET                     ; Return.
;
;       SAVSCR -- Save screen.  This routine saves the
;       current screen into buffer SCRBUF, clears the
;       screen, and turns off the cursor.
;
SAVSCR: PUSH    AF              ; Save A & flags.
        PUSH    BC              ; Save BC.
        PUSH    DE              ; Save DE.
        PUSH    HL              ; Save HL.
        LD      A,TRUE          ; Load value.
        LD      (IGNOR),A       ; Set flag.
REL20   EQU     $-2
        LD      A,DSP           ; Set up call.
        LD      C,SI            ; Turn off cursor char
        RST     28H             ; Output char.
        LD      A,VDCTL         ; Set up call.
        LD      B,06            ; Copy to buffer.
        LD      HL,SCRBUF       ; Point to buffer.
REL21   EQU     $-2
        RST     28H             ; Copy screen > buffer
        LD      A,VDCTL         ; Set up call.
        LD      B,04            ; Want cursor position
        RST     28H             ; Get position.
        LD      A,DSP           ; Set up call.
        LD      C,FS            ; Home cursor char.
        RST     28H             ; Output char.
        LD      A,DSP           ; Set up call.
        LD      C,US            ; Clear to EOS char.
        RST     28H             ; Output char.
        LD      A,VDCTL         ; Set up call.
        LD      B,03            ; Want to move cursor.
        RST     28H             ; Move cursor.
        LD      A,FALSE         ; Load value.
        LD      (IGNOR),A       ; Reset flag.
REL22   EQU     $-2
        POP     HL              ; Restore HL.
        POP     DE              ; Restore DE.
        POP     BC              ; Restore BC.
        POP     AF              ; Restore A & flags.
        RET                     ; Return.
;
;       RESSCR -- Restore screen. This routine restores
;       the screen by copying the buffer (SCRBUF) back
;       to the screen.  NOTE: The calling routine must
;       turn the cursor on (it's easier that way :-).
;
RESSCR: PUSH    AF              ; Save A & flags.
        PUSH    BC              ; Save BC.
        PUSH    DE              ; Save DE.
        PUSH    HL              ; Save HL.
        LD      A,VDCTL         ; Set up call.
        LD      B,05            ; Want  copy to screen
        LD      HL,SCRBUF       ; Point to buffer.
REL30   EQU     $-2
        RST     28H             ; Copy buffer > screen
;       LD      A,DSP           ; Set up call.
;       LD      C,SO            ; Turn on cursor char.
;       RST     28H             ; Output char.
        POP     HL              ; Restore HL.
        POP     DE              ; Restore DE.
        POP     BC              ; Restore BC.
        POP     AF              ; Restore A & flags.
        RET                     ; Return.
;
; This routine will be activated upon booting the disk
; if the filter has been installed and the SYSGEN command
; has been issued.
;
BOOT:   PUSH    AF              ; Save A & flags.
        PUSH    BC              ; Save BC.
        PUSH    DE              ; Save DE.
        PUSH    HL              ; Save HL.
        LD      A,DSPLY         ; Set up call.
        LD      HL,BOTMSG       ; Point to text.
REL40   EQU     $-2
        RST     28H             ; Print text.
        POP     HL              ; Restore HL.
        POP     DE              ; Restore DE.
        POP     BC              ; Restore BC.
        POP     AF              ; Restore A & flags.
LINK:   DEFS    3               ; 3 bytes(Opcode+Addr)
;
;       Data storage area:
;
DELAY   DEFW    DELVAL          ; Delay left.
IGNOR   DEFB    FALSE           ; Ignor char flag.
OUTFLG  DEFB    TRUE            ; Output-a-char flag.
SCROFF  DEFB    FALSE           ; Scrn-turned-off flag
SCRBUF  DEFS    2048            ; Screen buffer.
;
;       Messages:
;
BOTMSG  DEFB    LF
        DEFB    LF
        DEFM    'Model IV CRT Saver.  Version '
BOTLOC  DEFM    'x.x  By Jon Scheer (C) 1986'
        DEFB    LF
        DEFM    '   Filter installed and active.'
        DEFM    '  Timeout occurs after 3 minutes.'
        DEFB    LF
        DEFB    LF
        DEFB    CR
;
FLTEND  EQU     $-1             ; End of filter.
FLTLEN  EQU     $-FLTBEG        ; Length of filter.
;
;       Filter Installation Routine.
;
INIT    EQU     $               ; Start  init routine
        PUSH    DE              ; Save DCP pointer.
        LD      (MODDCB),DE     ; Put into header.
        LD      A,VERNUM        ; Get version number.
        RLCA                    ; Swap the high bits
        RLCA                    ;   with the low bits.
        RLCA                    ;   (Rotate left.)
        RLCA                    ;
        AND     0FH             ; Clear high bits.
        ADD     A,'0'           ; Convert to ASCII.
        LD      (VERLOC),A      ; Put into string.
        LD      (BOTLOC),A      ; Put into boot string
        LD      A,VERNUM        ; Get version number.
        AND     0FH             ; Clear high bits.
        ADD     A,'0'           ; Convert to ASCII.
        LD      (VERLOC+2),A    ; Put into string.
        LD      (BOTLOC+2),A    ; Put into boot string
        LD      A,DSPLY         ; Set up call.
        LD      HL,SIGNON       ; Point to text.
        RST     28H             ; Print text.
;
;       Check to see if slot is available.
;
        LD      A,CKTSK         ; Set up call.
        LD      C,SLOT          ; Use slot # 2.
        RST     28H             ; Is slot in use?
        JR      Z,CANUSE        ; Jump if not in use.
        LD      HL,SLTBSY       ; Point to text.
        JP      ERROUT          ; Print error & quit.
;
 CANUSE: EQU    $
;
;       Activate task AFTER filter has been installed.
;       So install filter.
;
        LD      A,GTMOD         ; Set up call.
        LD      DE,MODNAM       ; Get module name.
        RST     28H             ; Already installed?
        JR      NZ,VIASET       ; Jump if not.
        LD      HL,INSTLD       ; Point to text.
        JP      ERROUT          ; Print error & exit.
;
VIASET: LD      A,FLAGS         ; Set up call.
        RST     28H             ; Get system flags.
        BIT     3,(IY+2)        ; Used SET?
        JR      NZ,SETHI        ; Jump if yes.
        LD      HL,NOSET        ; Point to text.
        JP      ERROUT          ; Print error & exit.
;
SETHI:  LD      A,HIGH          ; Set up call.
        LD      B,0             ; Want to use HIGH$.
        LD      HL,0            ; Want to return value.
        RST     28H             ; Get HIGH$.
        JR      Z,CHKHIG        ; Jump if no error.
        LD      HL,MEMERR       ; Point to text.
        JP      ERROUT          ; Print error & exit.
;
;       Must not use memory above F400, so move down
;       memory if too high.
;
CHKHIG: PUSH    HL              ; Save HL (HIGH$).
        PUSH    HL              ; Copy HL into BC.
        POP     BC              ;   (BC = HL)
        LD      HL,0F3FFH       ; Load value.
        SBC     HL,BC           ; HL = HL - BC.
```

Listing continued

Listing continued

```
        POP     HL              ; Restore HL (HIGH$).
        JR      NC,HIGHOK       ; Jump if memory ok.
        LD      HL,0F3FFH       ; Load value.
HIGHOK: LD      (OLDHI),HL      ; Save top mem location
;
;       Parse the relocation table.
;
        LD      IY,RELTAB       ; Point to reloc table.
        LD      DE,FLTEND       ; Point to end of filter.
        XOR     A               ; Clear A.
        SBC     HL,DE           ; HL = HL - DE.
        PUSH    HL              ; Copy HL...
        POP     BC              ;   into BC.
;
RELOC1: LD      L,(IY+0)        ; Get LSB of tbl value.
        LD      H,(IY+1)        ; Get MSB of tbl value.
        LD      A,H             ; A = H.
        OR      A               ; Set flags.
        JR      Z,PWRUP         ; Jump if done (0000).
        LD      E,(HL)          ; Get LSB of address.
        INC     HL              ; HL = HL + 1.
        LD      D,(HL)          ; Get MSB of address.
        EX      DE,HL           ; Swap DE & HL.
        ADD     HL,BC           ; HL = HL + offset.
        EX      DE,HL           ; Swap DE & HL.
        LD      (HL),D          ; Save MSB of new addr.
        DEC     HL              ; HL = HL - 1.
        LD      (HL),E          ; Save LSB of new addr.
        INC     IY              ; IY = IY + 1.
        INC     IY              ; IY = IY + 1.
        JR      RELOC1          ; Loop back.
;
; Before moving code, set up power-up initialization
; routine:
;
PWRUP:  LD      A,FLAGS         ; Set up call.
        RST     28H             ; Get flags.
        LD      A,(IY+28)       ; Get opcode.
        LD      (LINK),A        ; Save into code.
        LD      L,(IY+29)       ; Low byte of address
        LD      H,(IY+30)       ; High byte of address
        LD      (LINK+1),HL     ; Save into code.
        LD      A,0C3H          ; Opcode (CALL).
        LD      (IY+28),A       ; Save to init routine
        LD      HL,BOOT         ; Get address.
REL50   EQU     $-2             ; Calculate new address.
        LD      (IY+29),L       ; Save to init routine
        LD      (IY+30),H       ; Save to init routine
;
;       Now, move code:
;
MOVE:   LD      DE,(OLDHI)      ; Point to destination.
        LD      HL,FLTEND       ; Point to filter end
        LD      BC,FLTLEN       ; Get length of filter.
        LDDR                    ; Move filter.
        LD      A,HIGH          ; Set up call.
        LD      B,0             ; Want to use HIGH$.
        EX      DE,HL           ; HL = New HIGH$.
        RST     28H             ; Set new HIGH$.
        INC     HL              ; Point to flt start
        POP     IX              ; Get DCB off stack.
        LD      (IX),47H        ; Allow GET, PUT, CTL.
        LD      (IX+1),L        ; Save LSB of address.
        LD      (IX+2),H        ; Save MSB of address.
        LD      A,DSPLY         ; Set up call.
;
;       Now activate task.
;
        LD      A,ADTSK         ; Set up call.
        LD      C,SLOT          ; Use slot # 2.
        LD      HL,(RELTCB)     ; Point to new INTBEG.
        DEC     HL              ; HL = HL - 1.
        DEC     HL              ; Now HL -> TCB.
        EX      DE,HL           ; Swap DE and HL.
        RST     28H             ; Add task.
;
        LD      A,DSPLY         ; Set up call.
        LD      HL,SUCCES       ; Point to text.
        RST     28H             ; Print text.
        LD      A,EXIT          ; Set up call.
        LD      HL,0            ; Clear HL.
        RST     28H             ; Exit program.
;
ERROUT: LD      A,DSPLY         ; Set up call.
        RST     28H             ; Print text.
        LD      A,EXIT          ; Set up call.
        LD      HL,-1           ; Want to error out.
        RST     28H             ; Exit program.
;
;       Initialization data area:
;
;       Relocation table:
;
RELTAB: DEFW    REL00
        DEFW    REL01
        DEFW    REL02
        DEFW    REL03
        DEFW    REL04
        DEFW    REL05
        DEFW    REL06
        DEFW    REL10
        DEFW    REL11
        DEFW    REL12
        DEFW    REL13
```

Listing continued

Listing continued

```
                DEFW    REL14
                DEFW    REL15
                DEFW    REL16
                DEFW    REL17
                DEFW    REL20
                DEFW    REL21
                DEFW    REL22
                DEFW    REL30
                DEFW    REL40
                DEFW    REL50
                DEFW    0
        ;
        ;       Messages:
        ;
        SIGNON  DEFB    LF
                DEFM    'Model IV CRT Saver.  Version '
        VERLOC  DEFM    'x.x  By Jon Scheer (C) 1986'
                DEFB    LF
                DEFB    CR
        MODNAM  DEFM    'CRTSAV'
                DEFB    0
        SLTBSY  DEFM    'Slot #2 already running a task --'
                DEFM    ' process aborted.'
                DEFB    CR
        INSTLD  DEFM    'Filter already installed --'
                DEFM    ' process aborted.'
                DEFB    CR
        NOSET   DEFM    'Filter must be installed via SET.'
                DEFB    CR
        MEMERR  DEFM    'High memory not available --'
                DEFM    ' process aborted.'
                DEFB    CR
        SUCCES  DEFM    '   CRT Saver now installed.  Screen will'
                DEFM    ' clear after 3 minutes of inactivity.'
                DEFB    LF
                DEFM    '   Displaying another character will restore'
                DEFM    ' the screen.'
                DEFB    LF
                DEFB    LF
                DEFM    '*** Remember to FILTER the driver to *DO ***'
                DEFB    LF
                DEFB    CR
        ;
                END     INIT
```

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/

TRS8BIT

WELCOME TO THE 2014 XMAS EDITION.

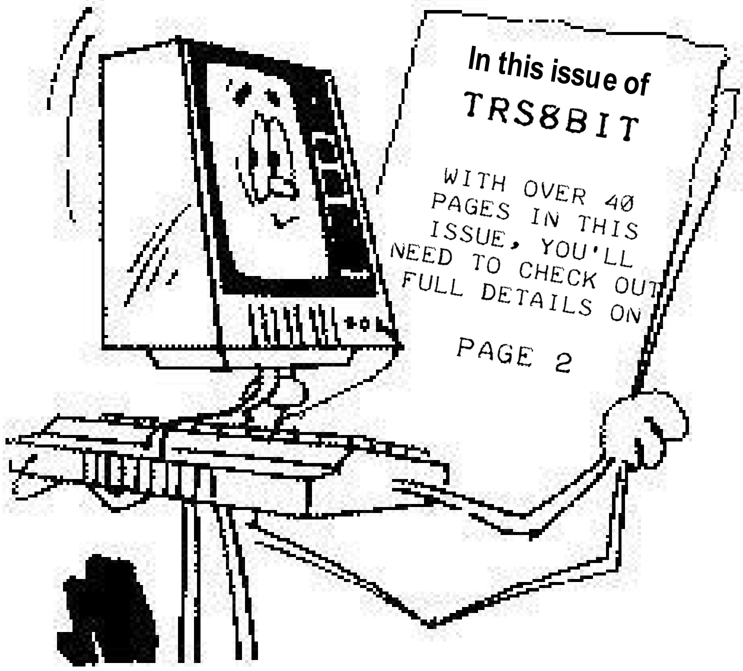
DEE AND I WISH YOU ALL A VERY MERRY CHRISTMAS AND A HAPPY NEW YEAR.

MANY THANKS TO EVERYONE WHO HAS CONTRIBUTED ARTICLES THROUGH OUT THIS YEAR. I KNOW THERE IS QUITE A LOT OF DEVELOPMENT IN PROCESS IN THE TANDY WORLD, SO I'M LOOKING FORWARD TO OUT 9TH YEAR WITH EAGER ANTICIPATION!

AND NOW FOR SOME EXCITING NEWS.

COMPETITION WINNERS.

DEE FELT THAT IT WAS NECESSARY TO SPLIT THE ENTRANTS INTO 2 SPECIFIC GROUPS. THOSE WHO HAD A HIRES SYSTEM AND THE OTHER LESSER MORTALS!



CONSEQUENTLY, I'M PLEASED TO REPORT THAT WE HAVE

2 WINNERS

THE FIRST, IN THE HIRES CLASS IS :-

GEORGE PHILLIPS

WITH A SPECTACULAR ENTRY USING HIS MODEL 4P

AND IN THE NON-HIRES CLASS THE WINNER IS :-

ROBERT SIEG

WITH A STUNNING ENTRY, USING HIS TANDY MC-10!

WELL DONE TO YOU BOTH. IF YOU WOULD LIKE TO CONTACT MAV, DIRECTLY, WITH YOUR POSTAL ADDRESSES, HE'LL ENSURE YOUR PRIZES ARE SENT OFF TO YOU BOTH.
I'VE PLACED A LINK ON THE COMPETITION PAGE ON THE WEBSITE IF YOU WOULD LIKE TO SEE BOTH ENTRIES UP AND RUNNING.

I'M HOPING WE'VE MANAGED TO PUT TOGETHER YET ANOTHER INTERESTING ISSUE FOR YOU ALL. WITH ARTICLES RANGING FROM M1'S TO M16'S THERE SHOULD BE SOMETHING OF INTEREST FOR MOST TANDY FANS!

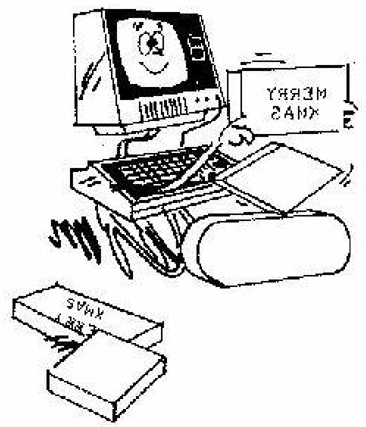
DON'T FORGET, IF YOU WOULD LIKE TO SEND IN A CONTRIBUTION, IT WOULD BE MOST WELCOME. ANYTHING TANDY RELATED WILL SUFFICE.

IN THE MEAN TIME:-

TAKE CARE EVERYONE

DUSTY

CONTENTS



PRINTER INTERFACE WOES

YOU MIGHT RECALL THAT IN THE SEPTEMBER ISSUE, I MENTIONED THAT I WAS HAVING A LOT OF FUN, MESSING ABOUT WITH A RE-INKER, TRY TO BREATH SOME LIFE INTO A 30-ODD YEAR OLD PRINTER RIBBON. BUT, AS USUAL, WITH MOST OF OUR LITTLE EXPERIMENTS, THEY SEEM TO LEAD TO ANOTHER PROBLEM AS SOON AS YOU'VE MANAGED TO SORT OUT THE INITIAL ONE! AND, OF COURSE, THIS TURNED OUT TO BE NO EXCEPTION.

ON THE OCCASIONS WHEN I USE MY 'REAL' MODEL 1, IT WILL INEVITABLY CALL FOR THE USE OF A PRINTER, IF ONLY TO LIST OUT A BASIC PROGRAM. TO SAVE USING THE EXPANSION INTERFACE JUST BECAUSE IT HAS A PARALLEL PORT, I WILL, ON MOST OCCASIONS, USE THE TANDY PRINTER INTERFACE, WHICH ATTACHES TO THE 40WAY CONNECTOR AT THE REAR OF THE M1. (TANDY CAT. NO. 26 1411).
IT'S SO EASY TO USE, YOU LITERALLY JUST PLUG ONE END INTO THE PORT AT THE BACK OF THE M1 AND THE OTHER END INTO THE PARALLEL PORT OF THE PRINTER AND AWAY YOU GO, NO MEMORY TO RESERVE FOR A PRINTER DRIVER AND ALL THE 'L' COMMANDS ARE AVAILABLE FROM EITHER BASIC OR IN IMMEDIATE MODE.

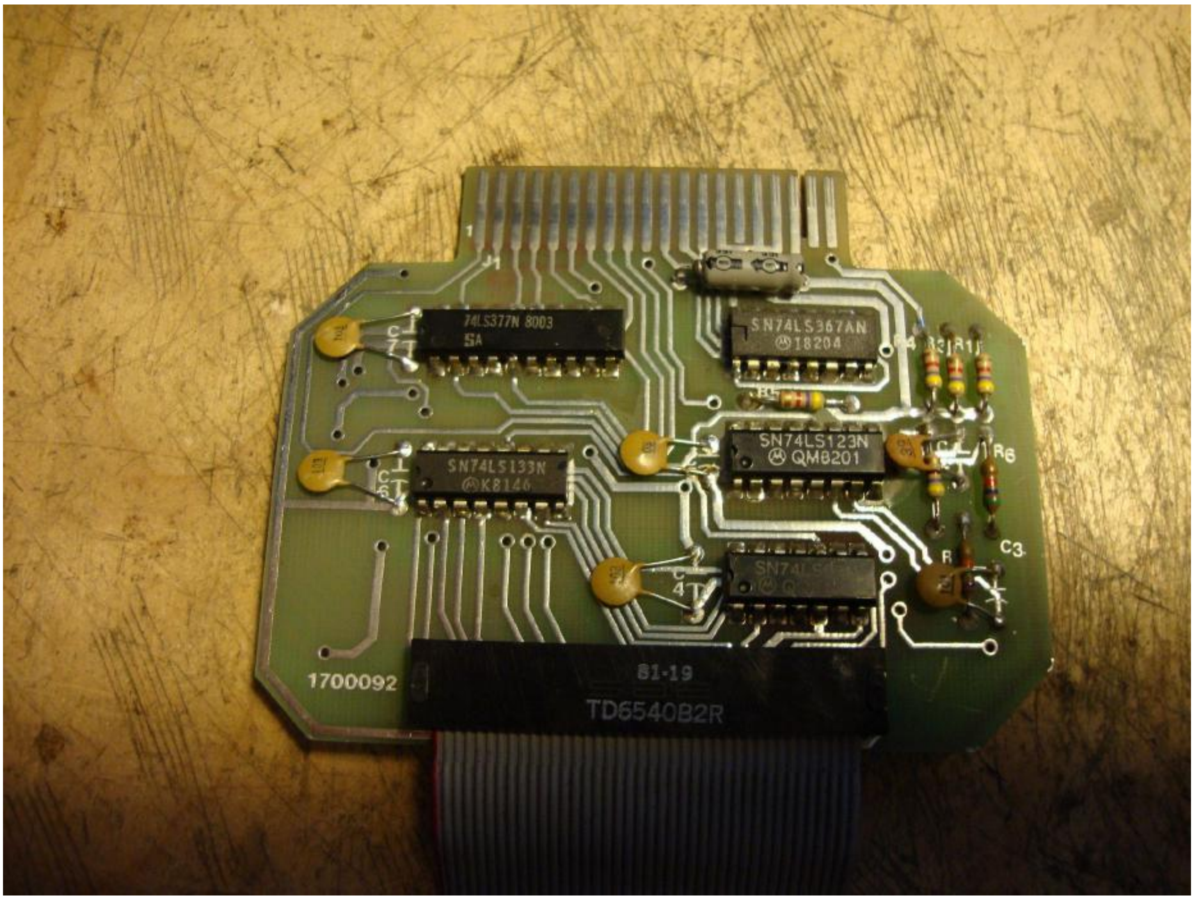
WELL, THERE I WAS HAPPILY ORGANISING THE PRODUCTION OF A 'BANNER' FOR MY GRAND-DAUGHTER'S BIRTHDAY. THE ONLY PROBLEM WAS, WHEN I ISSUED THE 'LPRINT' COMMAND, THE PRINTER JUST SAT THERE, NOT PRINTING BUT STILL SMILING AT ME AND THE MODEL 1 HAD RETURNED STRAIGHT BACK TO THE 'READY' PROMPT. JUST (AS I'M SURE YOU'VE GUESSED), WHEN I WAS IN A BIT OF A RUSH TOO!

AFTER JIGGLING THE CABLES ABOUT, TRYING TO TEST-PRINT THE MICROLINE-80 ETC. ETC. I DECIDED THAT I'D NO MORE TIME TO PLAY ABOUT. SO I HAD TO PLUG IN THE EXPANSION INTERFACE AND RE-LOAD THE PROGRAM (FROM CASSETTE!) AND THEN TRY PRINTING AGAIN, USING THE SAME CABLE, BUT THIS TIME USING THE PARALLEL PORT ON THE E.I.

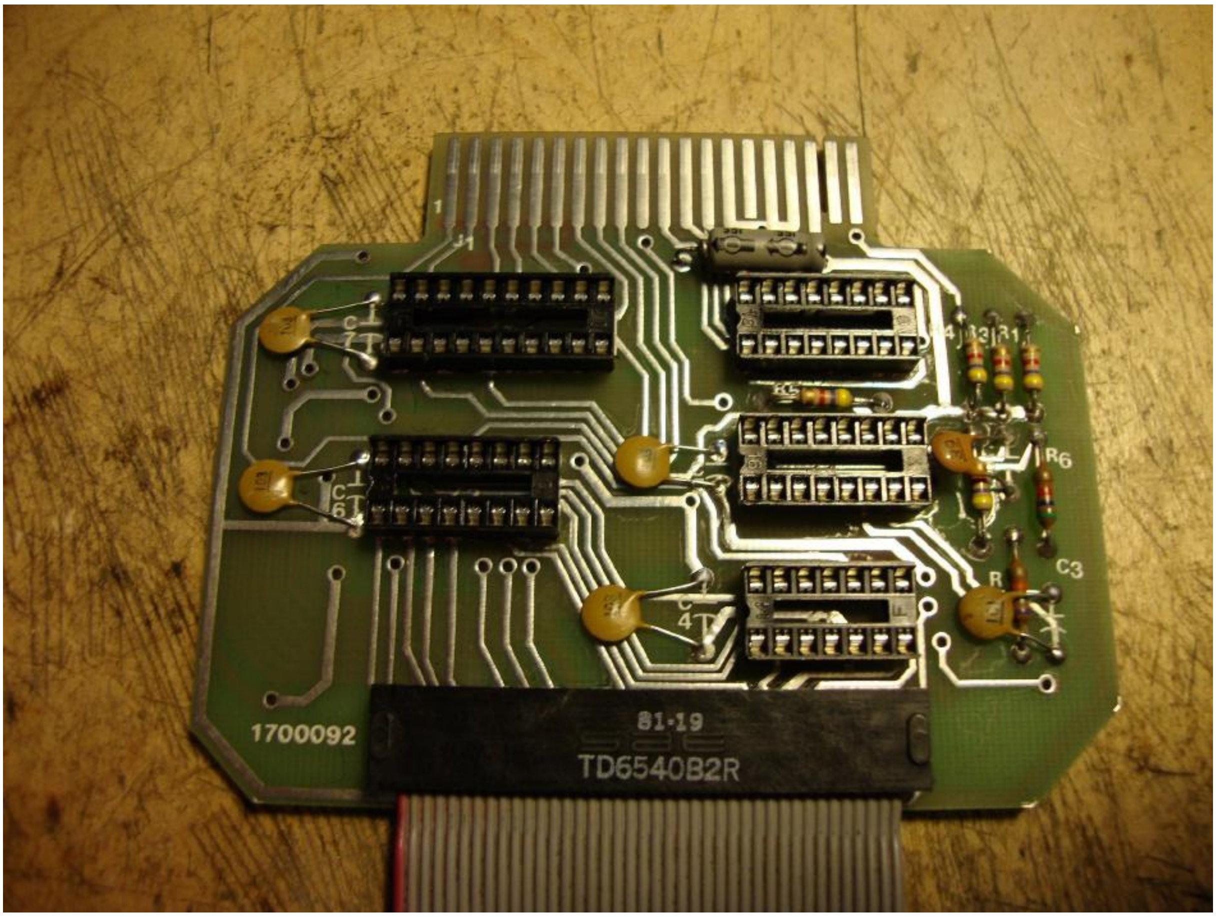
THIS WORKED FINE SO I ASSUMED THAT THIS PROVED THAT THE MODEL 1 AND THE PRINTER WERE WORKING OK, SO THE PROBLEM SHOULD BE WITH THE PRINTER INTERFACE. THIS WAS A FIRST, IN NEARLY 35 YEAR OF USING TANDYS, THAT I'VE EVER COME ACROSS THIS SPECIFIC PROBLEM BEFORE!

THE PRINTER INTERFACE IS ACTUALLY A SEALED UNIT AND AS MY ELECTRONIC EXPERIENCE IS VERY LIMITED, I THOUGHT I'D END UP DOING MORE DAMAGE BY OPENING THE BOX, SO I DECIDED, I THOUGHT SENSIBLY, TO LEAVE WELL ALONE, AND CONTACT BAS GIALOPSOS AND SEE IF HE COULD HELP.

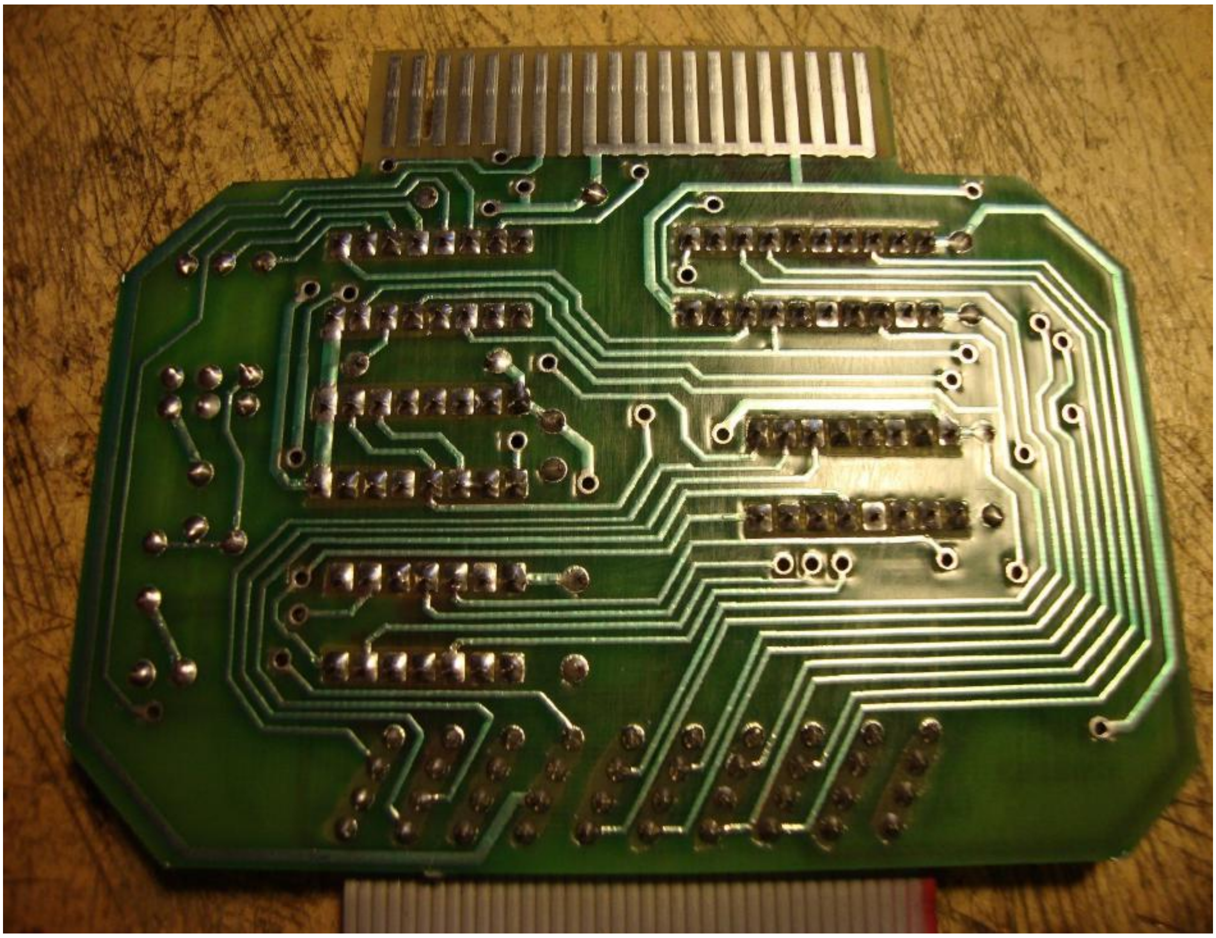
BAS WAS JUST BRILLIANT, HE CHEERED ME UP NO-END WITH THOSE MAGIC WORDS "NO PROBLEM", JUST SEND IT OFF TO ME AND I'LL SORT IT OUT FOR YOU. THIS HE DID, IN RECORD TIME, AND I WAS UP AND RUNNING AGAIN IN UNDER A WEEK. BAS INFORMED ME THAT AS THERE ARE ONLY 5 CHIPS ON THE INTERFACE, NONE OF WHICH WERE PARTICULARLY EXPENSIVE, HE'D REPLACE THEM ALL TO SPEED



THE ORIGINAL, TOP SIDE, PRINTED CIRCUIT BOARD



AS ENHANCED, WITH SOCKETS, AWAITING CHIPS



UNDERNEATH SIDE, AS ENHANCED, WITH SOCKETS. WHAT A NEAT JOB TOO! I WISH I COULD SOLDER LIKE THAT.

THE REPAIR UP. NOT ONLY DID HE REPLACE THE CHIPS, BUT HE PUT IN SOCKETS TO MAKE ANY FURTHER CHIP REPLACEMENT PROBLEMS, WHICH I MIGHT ENCOUNTER IN THE FUTURE, MUCH EASIER TO DEAL WITH.

MANY THANKS BAS, YOU DID A BRILLIANT JOB FOR ME, WHICH IS REALLY APPRECIATED. BAS'S MOTTO IS TO FIX, 'ANYTHING THAT COMPUTES' SO IF YOU NEED SOME ASSISTANCE, CHECK HIM OUT, HE COMES VERY HIGHLY RECOMMENDED.

IN MAV'S WORKSHOP

by Ian Mavric

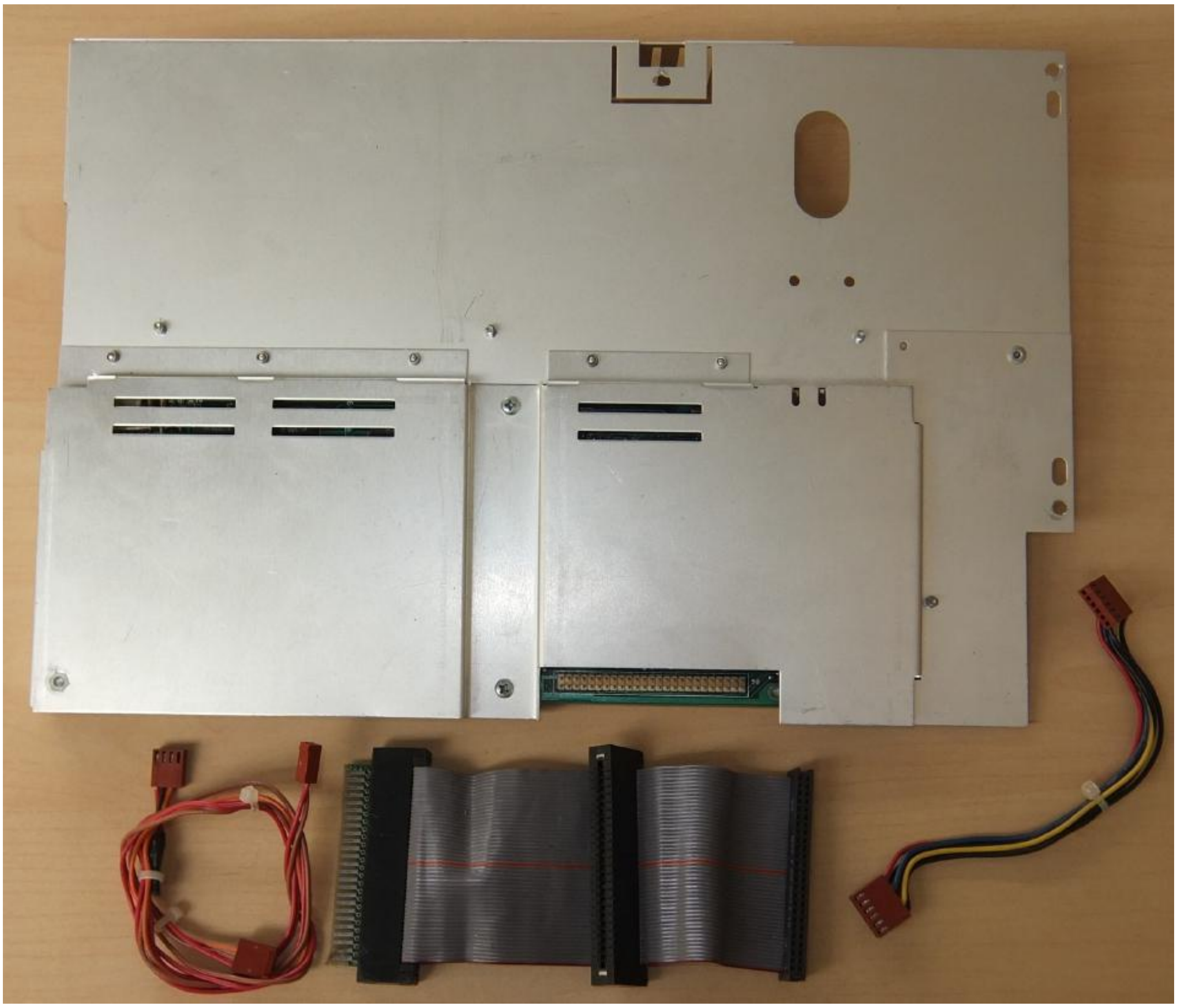
MODEL III GRAPHICS BOARD INTO A MODEL 4

Some time ago I purchased a Model III off eBay which had the rare hi-res graphics board installed. For those who are unaware, Tandy sold a hi-res graphics board (Cat. No. 26-1125) which added 640x240 (153,600 pixels) hi-resolution graphics to the Model III[1]. It was expensive and was not initially supported by a lot of software. As time went on and Tandy released the Model 4 and its version of the graphics board with the same capabilities, more software came along, and the hi-res board became an interesting and useful upgrade.

When my eBay Model III arrived it was smashed due to poor packing by the seller and/or poor shipping by the carrier. The CRT and cabinet was busted, but it turned out the rest of the machine was in pretty good condition and the main guts of the machine looked pretty undamaged and still worked. Pondering which way to move forward, I remembered that Ira's www.trs-80.com site had details about how to make a Model III graphics board work on the Model 4[2], and this seemed like the perfect opportunity to try out that project.

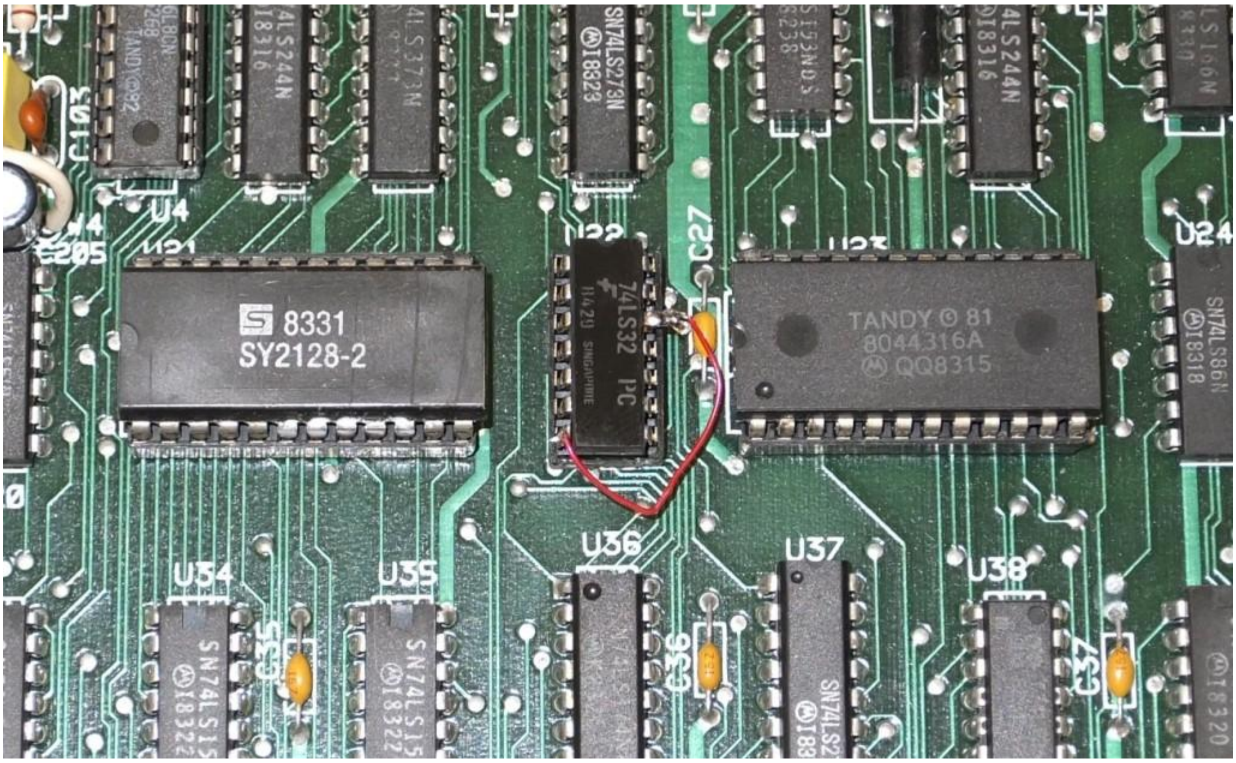
WHAT YOU NEED TO PERFORM THE UPGRADE

The hardware of the Model III hi-res board comprises of four parts: the hi-res board assembly, a bus interconnect cable assembly, a video pass-through cable, and a power cable (see picture). The upgrade only works on Model 4 non-Gate Array systems. It won't work on Gate Array systems due to differences in the video output section of the motherboard. Check your Model 4 to see if it has a 2-pin power connector hidden away near the power supply - most have them, it is used to power network cards on Model 4 student workstations - but in this case it will be used to power the hi-res board. If your Model 4 started out as a 16K diskless machine which has the two Model III-style 35-watt power supplies inside, then you will need to use the power cable supplied with the hi-res board - it replaces the existing FDC power cable.



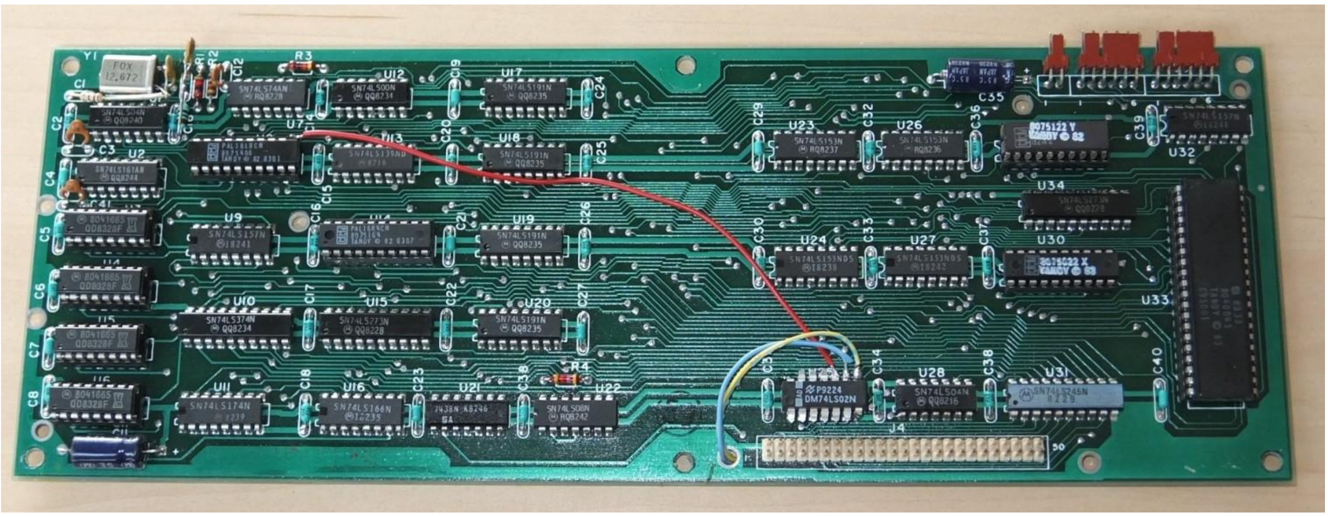
MODIFY THE MOTHERBOARD

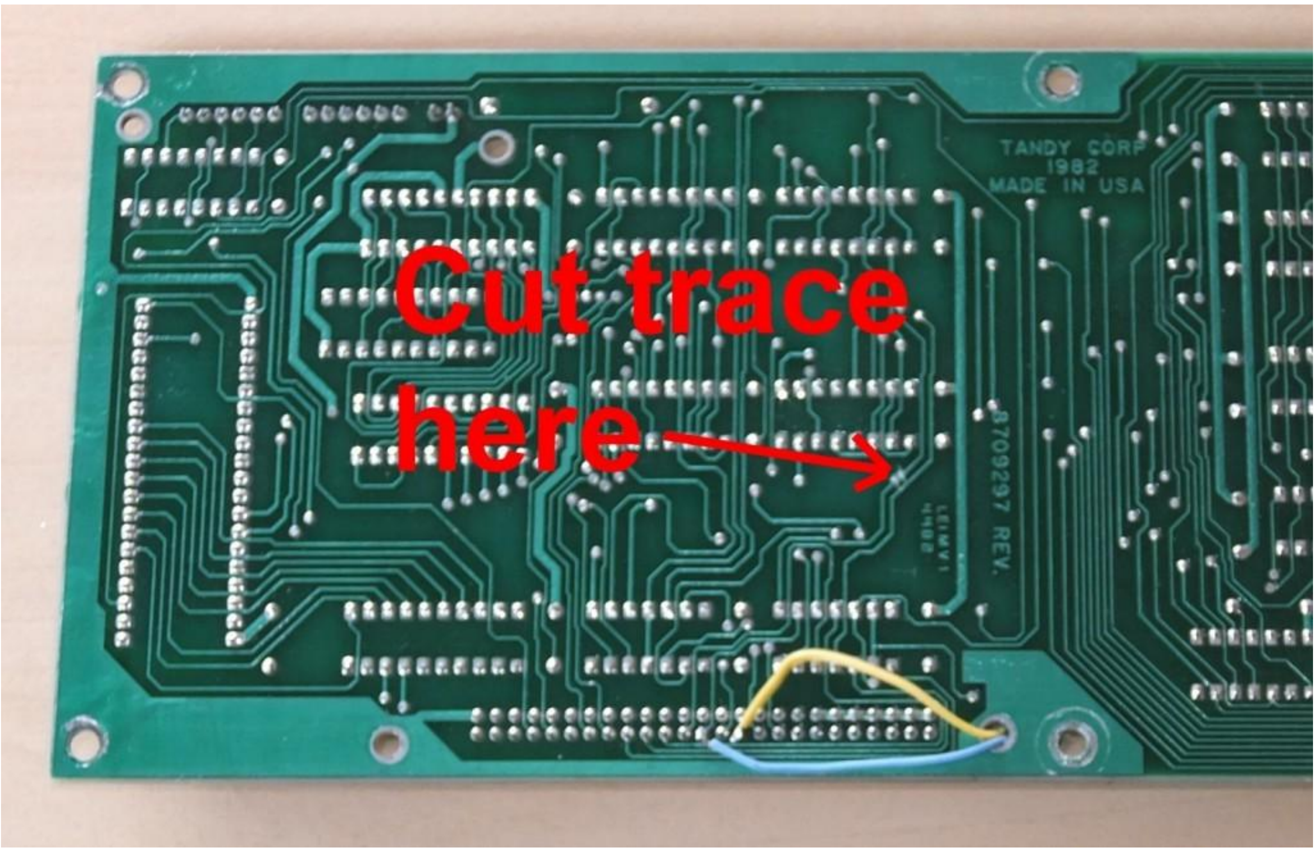
U22 is a 74LS32 and needs to have pin 12 disconnected from the motherboard and wires to pin 7. Getting at the pin is difficult so I just unsoldered the chip, installed a socket and performed the modification on a new chip. This way I can also go back to standard in the future if needed.



MODIFY THE HI-RES BOARD

U25 on the hi-res board gets a 74LS02 piggybacked onto it - only the power pins (7 and 14), and you then run three wires: Pin 8 to Pin 21 of the 50-pin I/O bus; Pin 9 to Pin 25 of the 50-pin I/O bus; and Pin 10 to Pin 11 of U8 (a PAL chip). Lastly you need to cut the trace which connected U25 Pin 8 to U8 Pin 11. With only one trace cut, I can also easily go back to standard if needed in the future by unsoldering the piggybacked 74LS02 and restoring the trace cut.





RE-ASSEMBLE YOUR COMPUTER

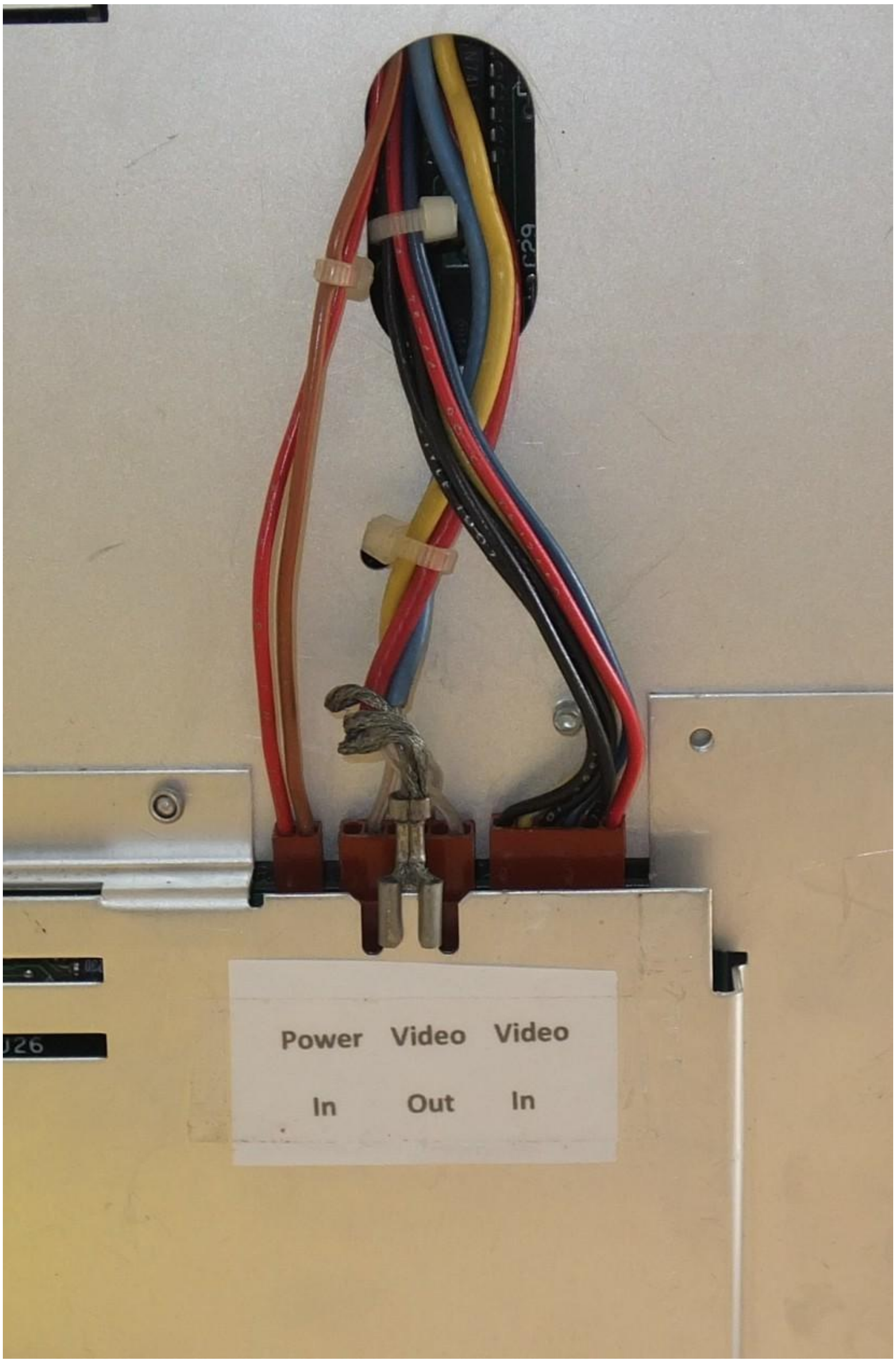
If you removed your hi-res board from a Model III then just connect it up in the same manner on your Model 4. If not, it's pretty easy to work out...

Connect the bus interconnect cable assembly to the Model 4 motherboard so that they "tail" hangs out the bottom of the computer (this is so external devices like a hard drive or FreHD can be attached to the computer), and the short female cable faces chips U73-76 on the motherboard. Screw in the eight screws which hold the motherboard in place, and plug in everything except the hi-res board. Test functioning of the system - it should all work as it did before.

Now connect the hi-res board into the bus interconnect cable and mount the whole aluminium chassis assembly in place where the previous RFI shield was. Your Model 4, particularly if it was an international version, may not have had an RFI shield fitted, if so examine the holes on the hi-res board's aluminium chassis and the Model 4 electronics chassis and it's pretty self explanatory where all six holes line up.

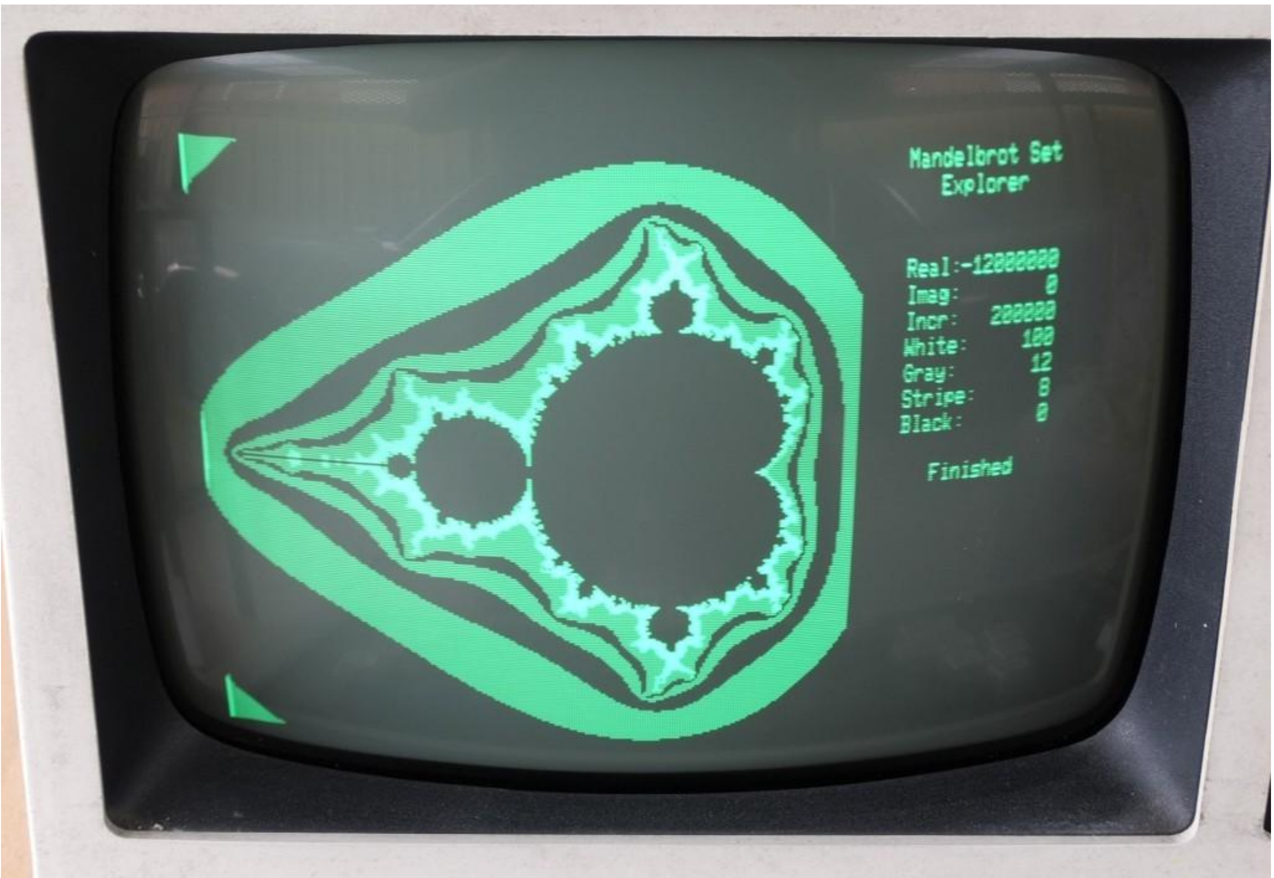
WIRE IT UP ALREADY!

On the back of the hi-res board you will see three connectors, a 2-pin one, and two 6-pin ones. Plug the two-pin power cable into the 2-pin connector. Motherboard video now runs via the hi-res board so the short 6-wire cable connects to the existing video output on the motherboard and the right connector on the hi-res board. Finally the old video cable which used to connect to the motherboard is connected to the middle connector on the hi-res board, along with it's grounding strap (see photo).



TESTING

Booting the system up it should run normally, and you can test it out with hi-res programs. Fractals by Micro-Labs Inc[3]. is a Model 4 program which is one of my favourites, it boots quickly and will confirm the hi-res board's operating status in a few minutes. Remember also to program in hi-res BASIC, called "BASICG" you need copies of the special versions of TRSDOS 1.3 from the 26-1125 kit and the TRSDOS 6.x disk from the 26-1126 kit.



HISTORICALLY SPEAKING...

I wondered why Radio Shack itself went to the trouble of releasing a tech bulletin about making the Model III hi-res board work on the Model 4... surely anyone with a Model 4 would buy the new-design 26-1126 hi-res board, as it was $100 cheaper than the Model III board. It made no economic sense to Model 4 owners to buy a Model III board to install in a Model 4. Then I remembered Radio Shack offered the 26-1123 Model III -to-4 upgrade[4], which replaced the Model III motherboard with a Model 4 one. I can see a predicament where an owner of a hi-res Model III wanted to upgrade to a Model 4 but re-use the hi-res board.

SOURCES OF HI-RES SOFTWARE

While not supported as extensively as the hi res graphics on the Apple][, IBM PC, or even the Coco, there is enough software to make interesting use of the hi-res board if you know where to find it. The BASICG language is similar to GW-BASIC on the IBM PC and Extended Colour Basic on the Coco, with commands like LINE, CIRCLE and PAINT much software can be easily modified to use on the Model III and 4. 80micro regularly ran articles with hi-res software from the years 1985-87.

In a future article I may outline some of the better software for the hi-res board and where to find it.

Ian Mavric

ianm@trs-80.com

References:

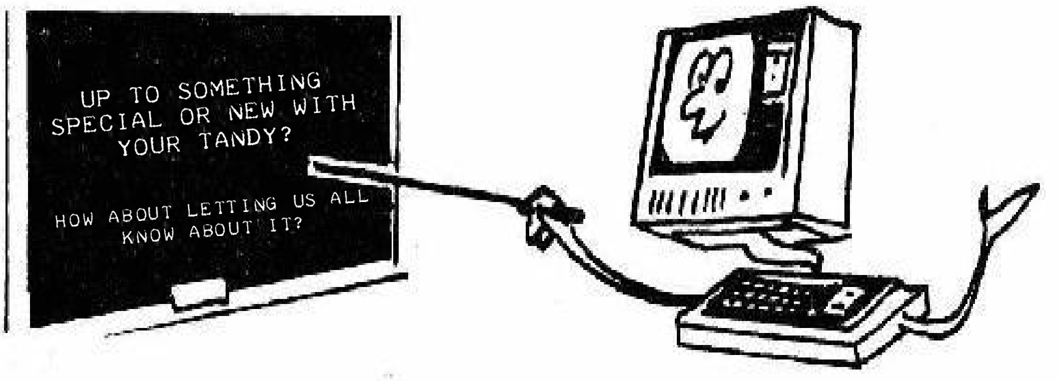
1. TRS-80 Catalogue RSC-9, September 1983, page 28.

2. http://www.trs-80.com/wordpress/zaps-patches-pokes-tips/hardware/#m3graphics

3. http://www.trs-80.org/model-4-grafyx-solution/

4. TRS-80 Catalogue RSC-9, September 1983, page 20.

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.



8-BIT COMPUTER EVENT
SATURDAY-1ST NOVEMBER 2014

WE HAD A SUPER SUNNY DAY FOR THIS, THE FIRST EVER 8-BIT COMPUTER EVENT, A REAL HELP WHEN LOADING AND UNLOADING!

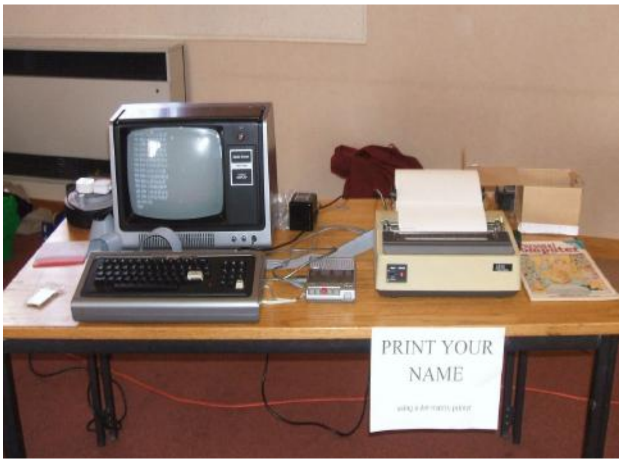
DUE TO ILLNESS AND OTHER MISHAPS, A COUPLE OF CLASSIC COMPUTER FANS WERE UNABLE TO ATTEND AND THEIR APPLE IIE'S AND SPECTRUM'S WERE SADLY MISSED. HOWEVER, JUST UNDER 20 PEOPLE CAME THROUGH THE DOOR DURING THE DAY, WHICH I WAS PLEASANTLY SURPRISED WITH.
PETER STONE CAME ALL THE WAY FROM WOLVERHAMPTON BY PUBLIC TRANSPORT AND ALTHOUGH, BY NECESSITY, HIS TIME WITH US WAS LIMITED, IT WAS REALLY GREAT TO MEET UP WITH HIM, FACE TO FACE, SO TO SPEAK, AS WE HAVE BEEN EMAILING EACH OTHER WITH ALL THINGS TANDY FOR NEARLY 8 YEARS.

THERE WAS A GOOD SUPPLY OF TANDY COMPUTERS NEEDLESS TO SAY. I TOOK MY 2 MODEL 1'S, A MODEL 3 AND A MODEL 102.
THE FIRST MODEL 1 WAS RUNNING A BASIC PROGRAM TO PRINT OFF 'YOUR NAME', USING MY MICROLINE-80 PRINTER WITH ITS KEYBOARD PARALLEL PORT ATTACHMENT (SEE THE ARTICLE ON PAGE 3).
THE SECOND MODEL 1 HAD AN ACULAB FLOPPY TAPE SET-UP, BUT, OF COURSE, NO WORKING WAFERS!
THE MODEL 3 WAS SHOWING OFF ITS AUTO-BOOTING FREHD (AS THERE ARE CURRENTLY NO 5 1/4 FLOPPIES ATTACHED).
TO ROUND OFF THE TANDY CONTINGENT, MY MODEL 102 WAS UP AND RUNNING, ATTACHED (FOR THE FIRST TIME, BY USING THE BBC MODEL B'S PRINTER LEAD) TO (YET ANOTHER IRRESISTIBLE EBAY BARGAIN), A CGP-115, 4-PEN COLOUR GRAPHIC PRINTER.
ALL THE HARDWARE WAS SWITCHED ON FOR THE WHOLE TIME WE WERE OPEN AND EVERY PIECE PERFORMED FAULTLESSLY.

OTHER HARDWARE ON SHOW WAS A MODEL B, BBC, AN EPSON HX20 WITH AN EPSON RX80 PRINTER, A SINCLAIR QL (THE WAFER-DRIVE MODEL) AND JUST FOR THE 'GAMERS' A COUPLE OF OLD PC LAPTOPS RUNNING MATTHEW REEDS EMULATOR. JUST IN CASE YOU'RE WONDERING WHAT IF ANY WERE THE MOST POPULAR PROGRAMS RUNNING ON THE EMULATOR? JUST AS BACK IN THE 1980'S IT WAS ANDROID NIM (LEO CHRISTOPHERSON) AND ATTACK-FORCE (BIG FIVE SOFTWARE) WITH THE HIGH-SCORE LIST BEING WELL FOUGHT OVER!

DEE AND LORNA KEPT AN EYE ON THE KETTLE THROUGHOUT THE DAY, SO THERE WAS ALWAYS A GOOD SUPPLY OF TEA OR COFFEE AND FIONA (A COLLEAGUE FROM THE ANTIQUE CENTRE I HELP OUT AT), PROVIDED A VERY LARGE SUPPLY OF 'OATIE' BISCUITS, JUST ABOUT ENOUGH FOR THE WHOLE DAY!

MANY THANKS TO EVERYONE WHO ATTENDED AND HELPED THROUGHOUT THE DAY. I APPRECIATE LOUTH IS NOT ONE OF THE EASIEST PLACES TO TRAVEL TO, SO PERHAPS, FOR THE NEXT 'DO' IT SHOULD BE HELD SOMEWHERE CENTRAL WITH BETTER COMMUNICATIONS. ANYONE ANY THOUGHTS ON THE IDEA?

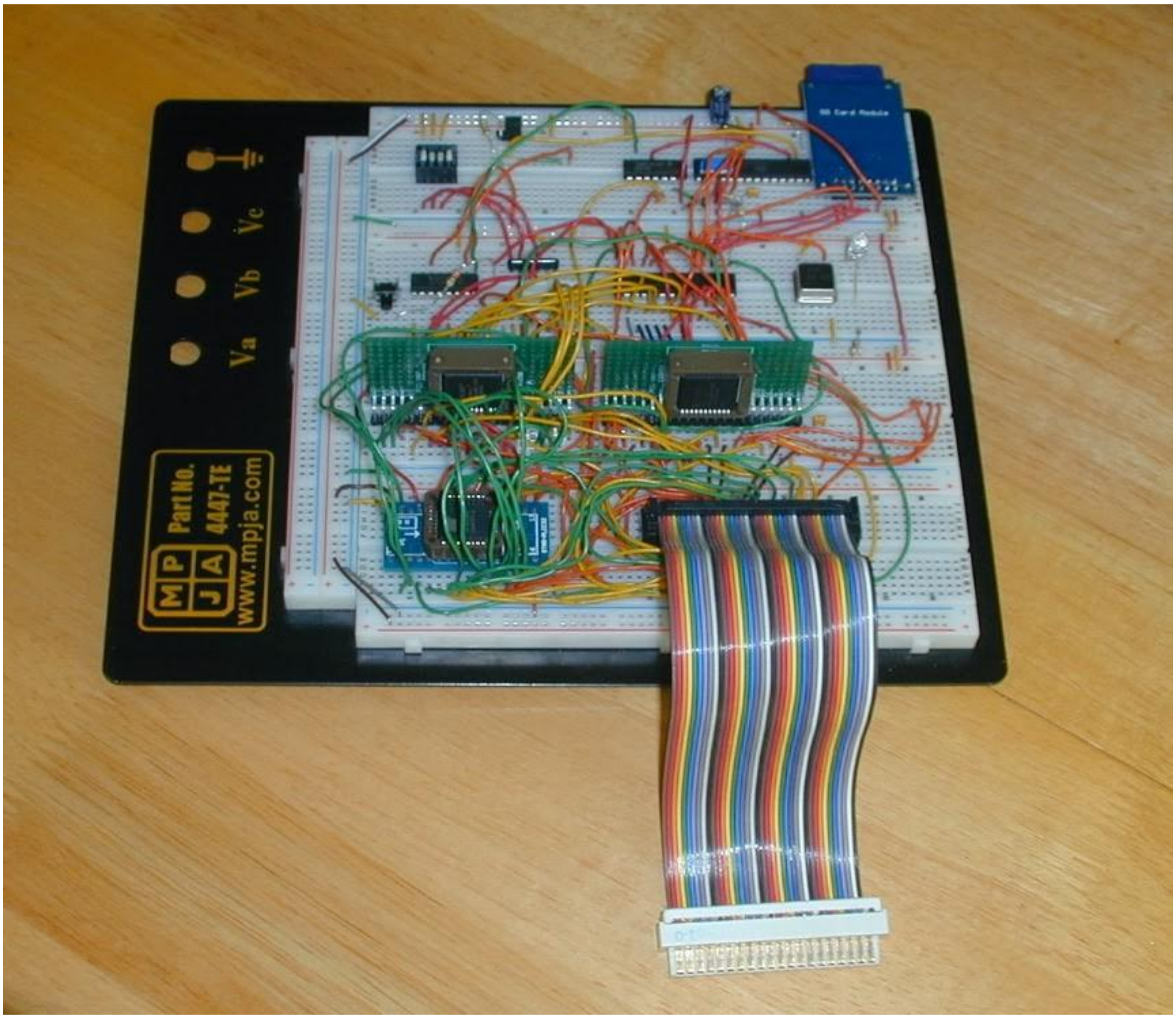


INTENSE EFFORT REQUIRED TO ENTER THE ATTACK FORCE HIGH-SCORE TABLE - JUST AS EVER!

MANY THANKS TO PETER STONE FOR ALL THE PHOTOS.

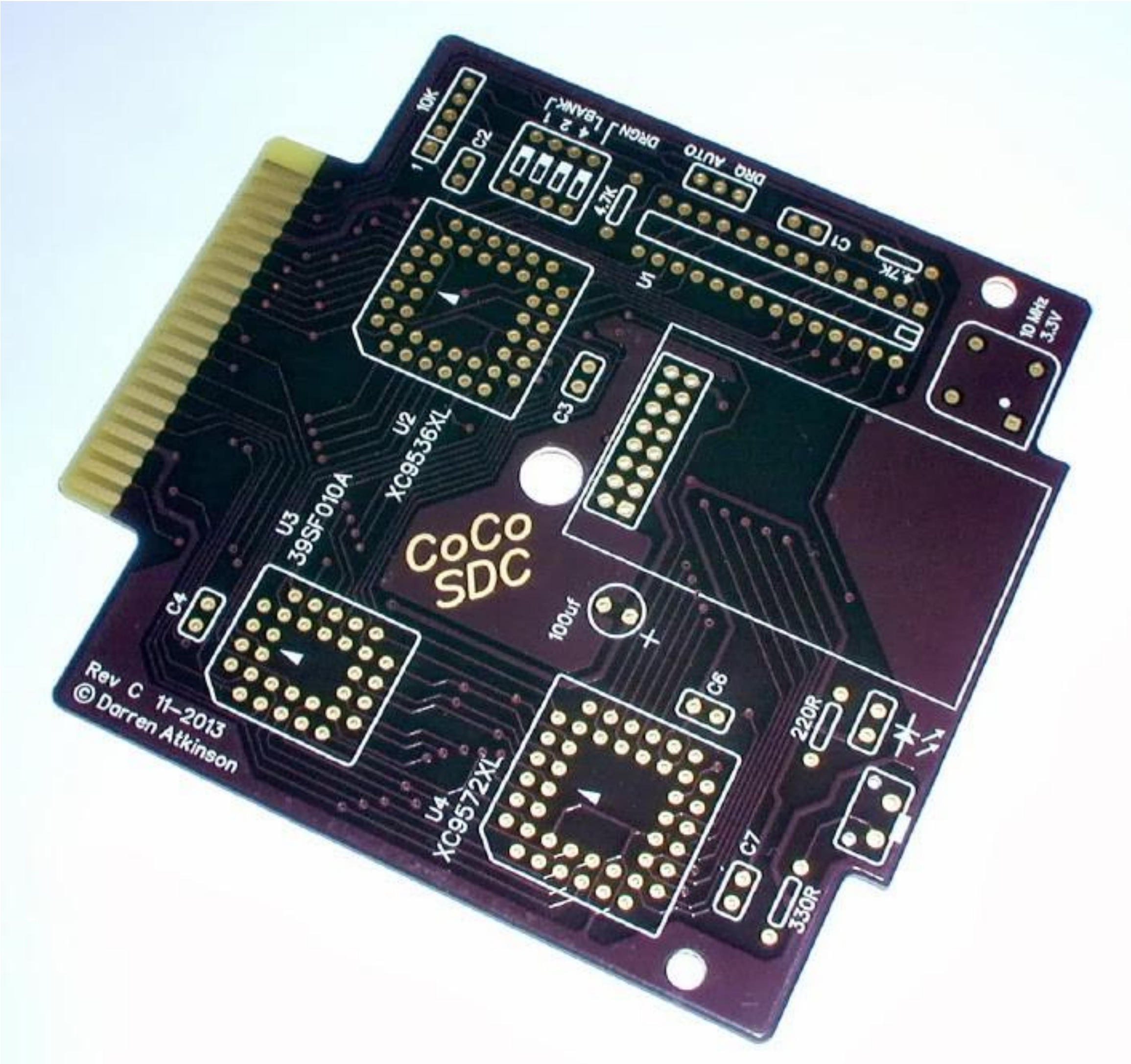
The CoCo SDC Project

Full details from -
http://cocosdc.blogspot.com.au/2014/01/introduction.html



The CoCo SDC is a home-brew project for the TRS-80 Colour Computer (CoCo). It has been in various stages of development since 2009. The original plan was to provide floppy controller emulation which worked in conjunction with the DriveWire server. That idea was eventually scrapped in favour of a self-contained system using an SD card.

A number of high capacity storage solutions have previously been developed for the CoCo, including a MicroSD card interface, a handful of IDE and SCSI interfaces and the very popular DriveWire server. One drawback of these offerings has been that they aren't compatible with software that was written to interact directly with a floppy disk controller. This isn't so much a problem if you are primarily using the CoCo for Basic programming or running OS9 software. There are however a number of titles (mostly commercial games) that fail to work with those other systems.



The CoCo SDC aims to solve the compatibility problem by combining the traditional "software hook" approach with a robust emulation of the floppy controller in hardware. This dual mode implementation provides excellent performance for the majority of software which "plays by the rules" while adding a high degree of compatibility with those titles that employ floppy-based copy protection schemes or simply choose to roll their own floppy drivers.

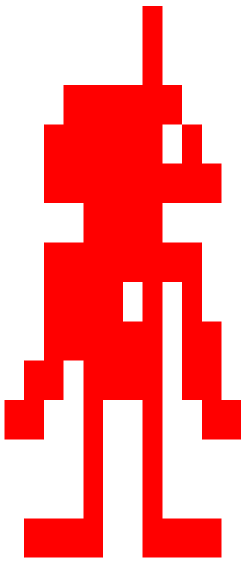
An enhanced LBA access mode has also been incorporated into the firmware, allowing the CoCo SDC to go beyond simply emulating floppy disks and interface with virtual hard disk images as large as 2 gigabytes. Two separate disk images (floppy or hard disk) contained on the same SD card may be "connected" simultaneously.

Also onboard is 128K of Flash memory which is divided into 8 banks of 16K. These 16K banks are both hardware and software selectable and occupy the cartridge ROM space from $C000 to $FEFF.

One bank of the Flash memory is used to hold the SDC-DOS code which is yet another patched version of Microsoft's Disk Extended Colour Basic 1.1. Included in SDC-DOS are additional commands to mount disk image files on the SD card, program the Flash and execute ROM images contained in the Flash. DriveWire disk support is also included in SDC-DOS.

Features and Specifications

- Atmega 328P AVR micro controller @ 10Mhz
  Custom 512 byte boot-loader allows firmware to be updated by the CoCo
- 128K In-System-Programmable Flash
- Accepts SD/SDHC cards formatted with FAT16 or FAT32 file system
- Emulates a Tandy Floppy Disk Controller
  Future firmware planned to emulate Dragon DOS floppy controllers
- LBA access mode for virtual hard disk support
- Extensions to Disk Basic in SDC-DOS for disk image manipulation
- DriveWire disk protocol with auto-speed configuration for CoCo 1, 2 or 3
- "Disk Switch" button to support multi-disk programs
- PCB can be mounted in a Tandy FD-502 enclosure



(: WARNING :)
THE NEXT PAGE IS CLASSIFIED

MEMORY EATING BUG!

WITH APOLOGIES TO LEO CHRISTOPHERSON - ANDROID NIM, STILL ONE OF MY ALL-TIME FAVOURITE PROGRAMS, WRITTEN FOR THE MODEL 1.

IT WAS NEWS TO ME, BUT A LONG-TIME CONTRIBUTOR TO TRS8BIT, THE WELL KNOWN AND PUBLISHED AUTHOR, E.T. FONEHUME, INFORMED ME THAT THE MEMORY LOCATION TAKEN UP BY THE BASIC PRINT POSITION 182 CAN BE A GATEWAY INTO RAM WHEN CERTAIN BINARY CHARACTERS ARE ISSUED BY BAMBORISMUS, THE EMPEROR OF THE NIM ANDROIDS!

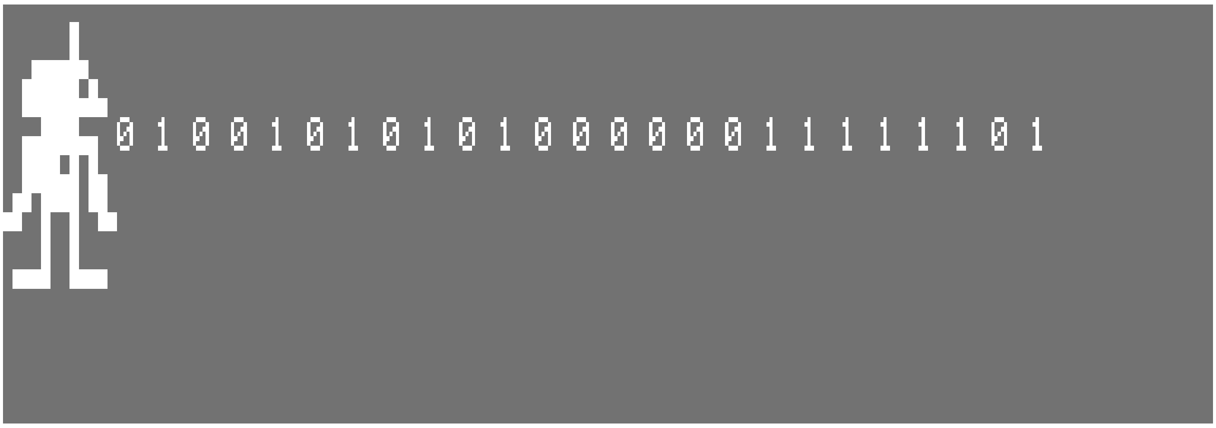
WITH EYEBROWS RAISED IN A TRULY SCEPTICAL MANNER, I ENQUIRED WHAT WAS SO SPECIAL ABOUT PRINT POSITION 182.
IT'S A SIMPLE EXPLANATION I WAS TOLD. IF YOU ADD TOGETHER 1+8+2 WHICH MAKES 11, THEN ADD 1+1 = 2. 2, IN BINARY, IS 10 AND THIS IS PART OF THE SECRET VIRAL CODE NEEDED FOR BAMBORISMUS TO 'SUCK-OUT' ALL YOUR RAM MEMORY, ONE BIT AT A TIME!
JUST TRY IT OUT IF YOU DON'T BELIEVE ME.

WELL.... WHO CAN RESIST ANY CHALLENGE LIKE THAT EH!

JUST TO MAKE YOUR LIVES EASIER, I'VE MANAGED, AFTER CONSIDERABLE RESEARCH I MIGHT ADD, TO FIND THE NECESSARY CODE TO PERFORM THIS, UP UNTIL NOW, SECRET ROUTINE.
I'VE EVEN MANAGED TO FIT IT ALL INTO ONE LINE OF BASIC CODING AND PLACED IT ON THE DOWN-LOADS PAGE, AS A '/BAS' FILE, JUST TO SAVE YOU TYPING IT ALL IN!

MERRY CHRISTMAS EVERYONE!

```
1 DATA128,128,128,168,12,128,190,191,191,185,144,12,128,188,191,
175,172,144,12,184,135,159,175,138,181,12,160,176,149,170,176,14
4,12:CLS:FORX=1TO32:READY:PRINTCHR$(Y);:NEXT:CLEAR200:FORZ=0TO1S
TEP0:X=RND(2)-1:P$=P$+STR$(X):P$=RIGHT$(P$,50):PRINT@133,P$;:NEX
T
```



IN MAV'S FreHD WORKSHOP
by Ian Mavric

HOW TO CONVERT TRS-80 MODEL III/4NGA TO FreHD AUTO BOOT

Last issue I started with the biggie, installing the auto-boot EPROM to the Model I. It involved lots of wires to be soldered, and a relatively expensive EPROM (27C256). This time I discuss how to add the EPROM to the Model III and 4NGA, which is positively a walk in the park. The EPROM can be produced by downloading the BIN file from the downloads page from my web site, or if you don't have your own EPROM programmer, I can burn and send you the EPROM for a small cost.

Only one part is needed, a 2716 EPROM which holds a modified version of the code which was originally in ROM C. Once installed you will lose the ability to CSAVE tapes at 1500 baud, as the FreHD auto-boot coding resides in the place the CSAVE routine used to like. Fred Vecoven - the designer of the auto-boot EPROM and indeed the FreHD itself - figured this was a fair trade-off. Not many people have the need to record 1500 baud tapes anymore. And I am in full agreement.

On the Model III:

System requirements: 48K Model III, disk or diskless system. You also need a FreHD.

Modification summary: Remove old ROM C and install new EPROM.

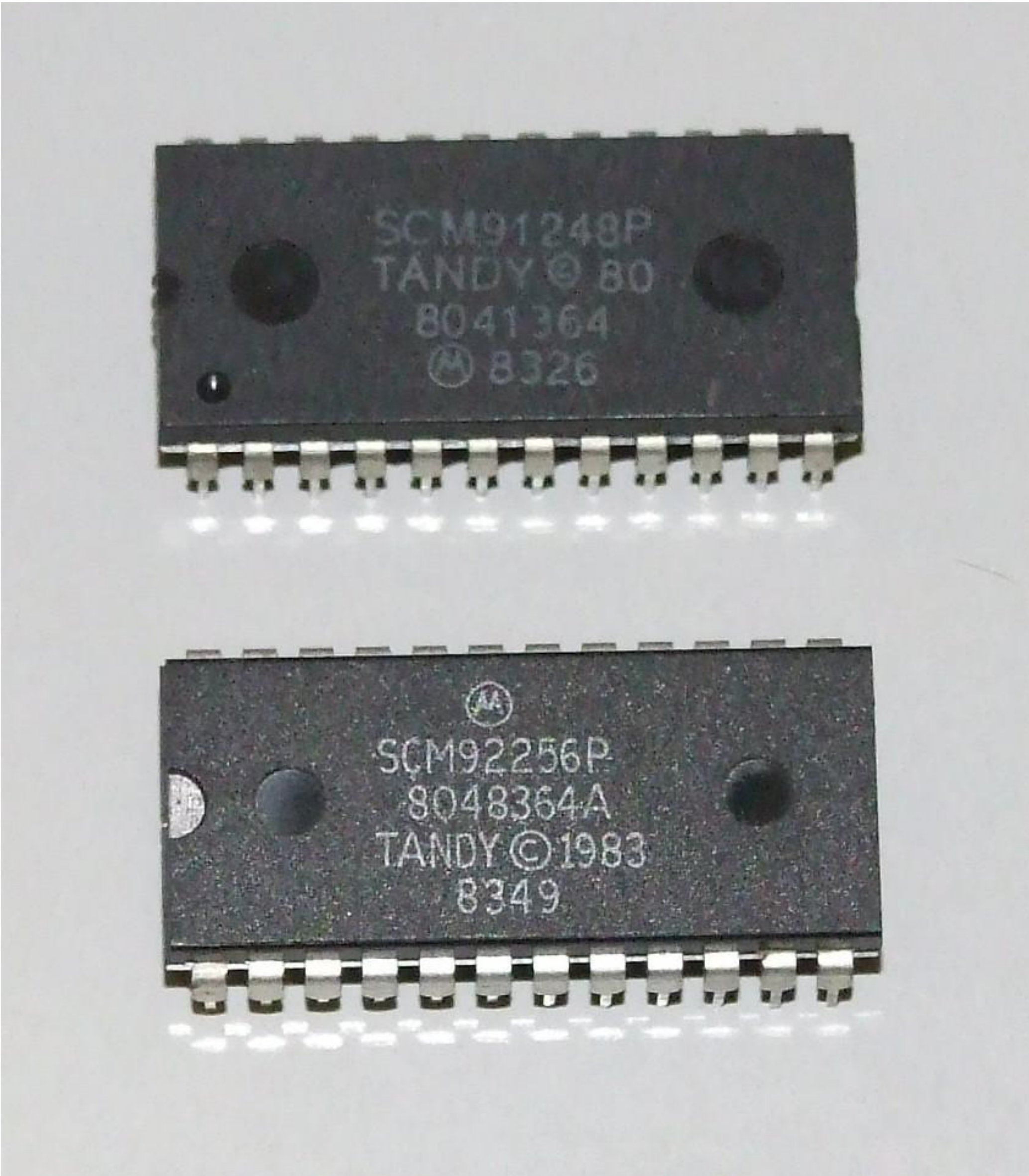
1. Open your Model III the usual way and remove the RFI shield covering the motherboard (if fitted). Locate U106 (it's in the lower right-hand corner of the motherboard) and remove the chip already installed. Burn the file http://ianmav.customer.netspace.net.au/downloads/model3rom-C2.bin to a 2716 EPROM and insert it into U106. Re-assemble computer and make sure it runs properly before connecting FreHD.

On the Model 4:

System requirements: 64K Model 4, disk or diskless. You need to have a Non-Gate Array motherboard. You also need a FreHD.

Modification summary: Determine which Rom version you have, remove old ROM C, and install new EPROM.

1. Non-Gate Array motherboards came with two distinct ROM sets which we call GEN1 (installed on 80% of Model 4s) and GEN2 (installed on the other 20%). To determine which one you have, either download ROMID/CMD (a program by Dean Bear) from my web site: http://ianmav.customer.netspace.net.au/downloads/romid.zip Run it on your machine and it will tell you which ROM version you have. The other way is to look at U68 and if it has Tandy code 8041364 on it then it's a GEN1 ROM set installed, and if U68 has 8048364 or 8048364A then you have GEN2 installed. See picture:



Picture: U68 GEN1 ROM (top) and GEN2 ROM (bottom)

2. Burn the file http://ianmav.customer.netspace.net.au/downloads/model4rom-C.bin (GEN1) or http://ianmav.customer.netspace.net.au/downloads/model4romgen2-C.bin (GEN2) to a 2716 EPROM and insert it

into U70. Re-assemble computer and make sure it runs properly before connecting FreHD.

SET UP YOUR SD CARD:

Auto-boot FreHD systems require a special image file (at this stage) and fortunately we have put all the popular Model III/4 operating systems into a single ZIP file you can download from: http://ianmav.customer.netspace.net.au/downloads/FreHD%20Self%20Boot%20Start%20Files.zip LS-DOS 6.3.1, CP/M 2.2, LDOS 5.3.1 and Newdos/80 V2.5 (obviously only the last two work on the Model III) and a small file called frehd.rom which is the menu system.
CONCLUSION:

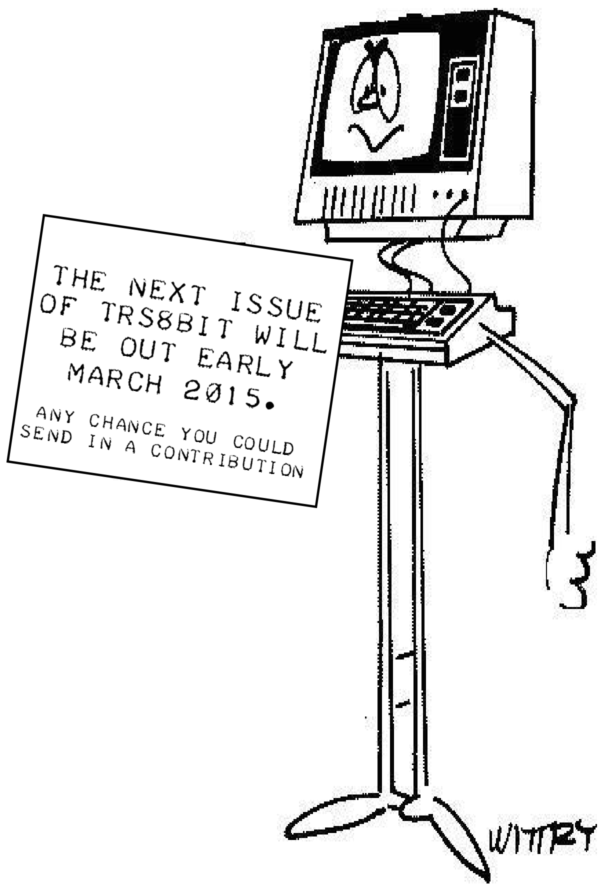
This modification gives you a reliable system which boots up quickly without a floppy disk. As diskettes become more scarce and unreliable, this system makes using the TRS-80 much more pleasurable since you no long need to worry if your favourite boot disk will still work.

I'd like to mention Dean Bear for all his hard work on the ROMID/CMD program as it makes checking ROM version without opening the TRS-80 quick and painless. Thanks buddy.

Next time: Upgrading the Model 4GA/4D with FreHD auto-boot.

Ian Mavric

ianm@trs-80.com



```
       AS             YX
    YXMASME       MERRYXMA
   RYXMASME       MERRYXMAS
  RRYXMASME  YXMAS ERRYXMASM
  RRYXMASM  RYXMAS ERRYXMASME
 ERRYXMAS   RYXMASM RRYXMASME
 ERRYXMAS   RYXMASM  RYXMASME
MERRYXMASM   YXMAS  RRYXMASMER
MERRYXMASME  YXMAS ERRYXMASMER
MERRYXMASME RYXMASM RRYXMASMER
MERRYXMASMERRYXMASMERRYXMASMER
MERRYXMASMERRYXMASMERRYXMASMER
MERRYXMASMERRYXMASMERRYXMASMER
 ERRYXMASMERRYXMASMERRYXMASMER
 ERRYXMASMERRYXMASMERRYXMASMER
 ERRYXMASMERRYXMASMERRYXMASMER
  RRYXMASMERRYXMASMERRYXMASMER
  RRYXMAS ERRYXMASMERR XMASMER
  RRYXMAS  RRYXMASMER    ASMER
   RYXMAS  RRYXMASMER    ASMER
   RYXMA   RRYXMASMER    ASMER
   RYXMA   RRYXMASMER     SMER
   RYXM    RRYXMASMER     SMER
   RYX    ERRYXMASMERR      ER
   RY     ERRYXMASMERRY      R
   R      ERRYXMASMERRY
         MERRYXMASMERRY
         MERRYXMASMERRY
         MERRYXMASMERRY
        SMERRYXMASMERRY
        SMERRYXMASMERRYX
        SMERRYXMASMERRYX
       ASMERRYXMASMERRYXM
         MERRYXMASMERRY
```

```
10 REM XMAS0804/BAS
20 CLEAR500
30 M$="MERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRYXMASMERRY
XMASMERRYXMASMERRYXMASMERRYXMAS"
40 GOSUB 220
50 READ S
60 IF S=0 GOSUB 160
70 READ L
80 T$=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 16K 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$(81,32)
230 RETURN
240 DATA 8,2,23,2,0,5,7,19,8,0,4,8,19,9,0,3,9,14,5,20,9,0
250 DATA 3,8,13,6,20,10,0,2,8,13,7,21,9,0,2,8,13,7,22,8,0
260 DATA 1,10,14,5,21,10,0,1,11,14,5,20,11,0,1,11,13,7,21,10,0
270 DATA 1,30,0,1,30,0,1,30,0,2,29,0,2,29,0,2,29,0,3,28,0
280 DATA 3,7,11,12,24,7,0,3,7,12,10,26,5,0,4,6,12,10,26,5,0
290 DATA 4,5,12,10,26,5,0,4,5,12,10,27,4,0,4,4,12,10,27,4,0
300 DATA 4,3,11,12,29,2,0,4,2,11,13,30,1,0,4,1,11,13,0,10,14,0
310 DATA 10,14,0,10,14,0,9,15,0,9,16,0,9,16,0,8,18,0,10,14,0
320 DATA 99
```

WE HOPE YOU LIKE THE XMAS ANGEL.

IT'S JUST TO WISH YOU ALL,
READERS AND CONTRUBUTORS, A VERY

MERRY CHRISTMAS
AND A HAPPY NEW YEAR

DUSTY & DEE

INNOVATIVE

TRS 80-GENIE SOFTWARE

from the professionals

QUICKPROPLUS

Quickpro Plus is a Basic program generator. That is to say you tell it the type of program you want and it writes it for you. The most widely publicised of such program generators is The Last One and it is, therefore, inevitable that Quickpro Plus will be compared with it.

There are two approaches that one can take in writing software like this. Either one can set out with a very broad brush and try and make the generator capable of producing a wide variety of data handling software or one can restrict it to some extent, to simply producing file handling programs. The Last One seeks to go the first route, Quickpro Plus goes the second. There is a great paradox in this software if one thinks about it. Obviously, if a person is at least a semi-skilled programmer then he does not need a program generator. They are really for people who are not skilled in programming and want that chore taken off their hands. The paradox is that programs like The Last One, by being all things to all men are also complex in use and one therefore gets the position of a program aimed at a beginner, but actually requiring some skill to use it.

It was because of this apparent paradox that Quickpro Plus came into being. It is written for somebody with little or no knowledge. You will find no mention of flow-charts and little mention of fields, records and other technicalities. It was written so that a person could sit down in front of his computer, answer a few questions and have a program produced for him, and this is exactly what Quickpro Plus does. The other side of the coin is that it concentrates entirely on producing file handling programs. Within that context the program which you have generated will run on the computer like any other Basic program. You will be able to add file records, in other words items of information in your file. You will be able to search for and locate records, and retrieve these records, as and when you wish; you will be able to up-date and change the records, indeed you can delete them altogether. In the program generation process you will be able to design your own screen layout. Co-ordinates appear on the screen and you simply say where you want questions and statements to be inserted. You will, of course, be able to define whatever part of the record you will wish to use as a search key. These fields may be restricted if you wish to just numeric data and, of course, you may name the data file and indeed the program as you desire.

An added feature is that you may carry out various calculations on any of the numeric fields and if you want to you can change this numeric data. Up to fifty separate computations can be carried out on these fields. The program will report the calculations to you in various arrangements using any of the normal mathematical functions.

Quickpro Plus supports a full print report facility. Indeed within minutes you can design a new report with any column names that you choose, with any calculations that you might want and for many selections of records in your total file. A report will be produced within seconds. This can have been built into the program or you can re-arrange matters so that you get a one time reporting. The same file is thereby manipulated in many different ways. Computations are done and results printed all from the same file which your program has produced.

Quickpro Plus is available for the Model I, Model II and Model III Tandy machines, together with the original Video Genie, the Genie I and Genie II. A version for the Genie III will be available towards the end of 1982. Quickpro Plus is, of course, disk orientated and has no application for cassette users. It is supplied on a protected disk, but Molimerx have masters from which they can repair any damaged disks and hence retain their reputation for support.

The Last One is a Registered Trademark of D.J. A1 SYSTEM LTD.

Quickpro Plus Tandy Model I, III & Genies £98
Tandy Model II £108
Both prices plus V.A.T. Receipted Parcel shipping £1.50.

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.

AMERICAN EXPRESS Cards Welcome

BARCLAYCARD VISA

Buy it with Access

ASK MAV ABOUT THE TANDY TRS-80 MODEL 4D MICROCOMPUTER

by Ian Mavric

1.

Q. What is a Model 4D? I've never heard of it...
A. In late 1985, in readiness for the 1986 Radio Shack Computer Catalogue RSC-15 Tandy introduced the Model 4D as a face-lifted better-value version of the Model 4 Gate Array computer. While I'm sure test units were sent to countries like Australia, UK and France, Tandy decided to only sell the 4D in the USA and Canada, and to let the stocks of Model 4s in other countries just sell out. Evidence of this is in the Australian 1986 Tandy Electronics catalog only featuring the Model 4 Gate Array and the 1987 Computer and main catalogues in Australia has no Model 4 on offer at all. I'm sure similar bad news was noted in countries other than the USA and Canada. To summarise a 4D is pretty easy, its a Model 4 Gate Array with an extra key (backspace), dual double-sided twist latch disk drives (actually 1/2 height drives with oversized face plates), and new nameplate, and Model 4 Deskmate software thrown in for free. Initially it was thought that the D stood for double-sided disk drives but Radio Shack was adamant it stood for Deskmate.

Deskmate was a simple and easy to use integrated suite of popular applications at the time including word processing, data manager, terminal program, spreadsheet etc. Tandy made Deskmate for various other computers including the Tandy 1000 and 2000, and the Colour Computer 2 and 3. It did it's job OK and probably helped with Tandy 1000 sales so Tandy figured the same would stimulate Model 4 sales.

2.

Q. Did Tandy advertise the computer? I don't recall any ads anywhere...
A. They must have figured the computer would sell itself because Tandy invested virtually nothing in promoting the machine. I can think of only one advert made for the 4D which appeared in magazines like 80micro (so essentially preaching to the already converted masses):

During it's long life (that's not to say it sold in huge numbers) it never dropped in price from $1199 and was still a regular feature in the 1990 catalog. People from 80micro speculated at the time that Tandy only did a single production run and it took that many years to sell them all, or that the profit margin per machine was so good that they didn't want to reduce the price on a computer which was selling 50-100 units per month. Conspiracy theorists speculated that Tandy wanted to keep a computer which was historically related to their first computer on the market due to the main competitor Apple still selling a computer (called the IIgs) which was still related to their first successful computer - the Apple][. I don't really buy that explanation because while the Apple IIgs was fairly current by 1990 (having a mouse, GUI, and 16-bit architecture) the Model 4D was just hopelessly outdated in every aspect.

3.
Q. I own a Model 4 that works well, is there any point in buying a 4D?
A. Not really. The 4D's double-sided disk drives are handy but double sided disk drives or 3.5in disk drives can be easily added to any Model 4. The 4D runs at the same speed and has the same memory sizes (64K up to 128K) as the Model 4. The green screen is quite nice to have if you own an early Model 4 with the B&W screen but basically that's it.

4.
Q. Does the 4D have any problems one should be aware of?
A. It's actually a very well sorted machine and rarely gives any problems. Early 4Ds had a membrane keyboard which felt dead to type on and was difficult to repair so if that keyboard fails you more or less throw it away and replace it with another one. Later keyboards were by ALPS and as well as being much nicer to type on, you can also repair individual key switches when they fail.

The disk drives are by Japanese manufacturer TEC and are type FB-503 and are not a bad unit, quiet and dependable with a direct-drive motor no belts to wear out. But there is one problem which affects most drives I have seen, which is the electrolytic capacitors tend to leak and cause corrosion damage the logic board. The caps which cause most problems are C14, C16, C17, and to a lesser extend C26 and C28. If your disk drive fails unexpectedly check around these capacitors for unexplained corrosion, and you may be able to replace them and replace damaged PCB tracks. This problem doesn't seem to relate to how much or little work the drive has done. I have seen drives with leaky capacitor which have only done a few hours work in their life. Inspect yours today!

5.
Q. The screen is very sharp, how did they get such a clear picture?
A. Upon opening the 4D I noticed they used the Model 4P video board, which is noted for it's very sharp picture. Time-wise this makes sense, Tandy stopped making the 4P around the time they started making 4Ds so probably had a large stock of 4P video board so just used them in the new computer. All 4Ds come with a high-quality Phillips 12VCMP31 picture tube which was also used in the high-end business computers like the Model 12, 16B and 6000. It's much sharper than the television grade picture tube used in TRS-80 Model I/III/4 computers.

6. How do I install a hi-res graphics board?

The Radio Shack graphics board (cat. no. 26-1126) from the regular Model 4 just fits plugged into it's own special connector the same way as the earlier machine. However due to the differing refresh-rates of the Model 4D and Model III, the Model III graphics board into Model 4 installation won't work, so don't try it. I have also used a Grafyx Solution board from Micro-Labs and it works fine on the 4D.

Next time: I answer questions about the Tandy TRS-80 Model 2000.

ianm@trs-80.com

Business Time with Kev

I my last article I introduced the second of my Model 16's. Thank you for all the positive comments as we (including me) re-discover these majestic machines.

An issue aside, my unpacking woes have got worse as we removed the remaining machines from the original storage area to move cars back in there. Moving three AS/400's into an already cramped and messy space just doesn't help but such is our lot in collecting life.

Now before you get to some glorious photos of the first Model 16 I acquired a little history is quite relevant here.

You might recall from my last article that the Model 16 featured came from a bakery. Here's what's interesting, so did my first one but not the same bakery. It came from a bakery in Mallala in South Australia. For those who are motor racing enthusiasts this town might sound familiar due to the presence of a very famous motor sport track.

Now for us Australians, Minlaton, where my other Model 16 came from, and Mallala are not that far away from each other – about 151km (or just down the road as we say).

I was talking to fellow columnist Ian Mavric about the significance of this or even it was significant. Back in those days a Model 16 was very expensive hardware and is it just a coincidence that two machines surface that are both from bakeries. By the way they didn't surface at the same time – I can't recall how far apart it was but I seem to recall it was about two years. Ian seemed to think that back in those days some states and Tandy stores were a little more entrepreneurial than others and maybe it's a case that someone saw a market in bakery software or just technology of some type in bakeries – we may never know. The first Model 16 I introduced is currently out for power supply repairs – we're hoping that the internal HDD is still viable on it and we may be able to learn from its contents what the machine was actually used for. The Model 16 that I'm featuring here did come with an external 15MB HDD but like most of them, it is unserviceable.
Now there's a little story to how I got his one. The one I featured in my first article was an eBay acquisition. For the machine featured in this article it was part of a trade. The owners of the bakery (at that stage it hadn't been operational for many, many years) were interested in a rotary hoe I had for sale so we did a deal which included the Model 16. From my perspective a very nice way to engineer a rescue because it was very unloved where it was.

So here it is in all its glory – unfortunately the keyboard for this one has not surfaced yet – like a lot of stuff it's still in a box somewhere.

And another angle…

One of the immediate differences, when compared to the first machine I wrote about, is that this one has two 8" floppy drives – these are the Thinline type the same as the first machine. You'll also notice that the case is far more yellowed than the first machine.
The catalogue number of this machine is 26-4005. Now for those who have jumped straight to Ira Goodling's catalogue number search you'll see this pop up as a "Model 12 - Computer - 2 Disk Drives" yet I've constantly made mention of having two Model 16's. To add to the confusion, and for those who have already zoomed in on the front of the machine, the front badge on the machine is for a Model 12.
Like the first Model 16, this one has a little upgrade sticker on the back. I tried to take some photos of the sticker but unfortunately I couldn't get them to turn out that well. You can see the sticker above the Arcnet port hole in the photo of the rear ports later in this article but unfortunately I couldn't get a "legible" shot. What it says is this:
"This unit contains Model 16 upgrade (Cat: No. 26-6010)"
Back to Ira's catalogue number search and this pops up as a" Model II - Upgrade - Model 16 Enhancement Option"

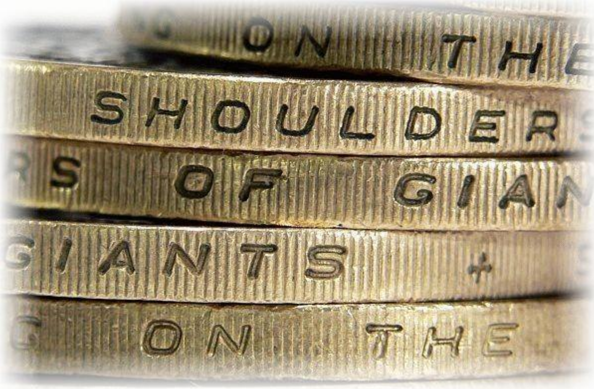
I grabbed this from Ira's web site – thank you Ira. At $1,499USD in the 80's I'd imagine that number was quite frightening. I'd also hazard a guess and say that this machine is possibly quite rare, rarer than a genuine 12 or 16, given the insertion of an upgrade kit.

At this stage I'm guessing from the upgrade stickers that the machine has 128K of RAM and was not optioned out to 256K – it will be some time before we know exactly. Unlike the first machine I introduced this one appears to be a mess inside. I flipped the cage fan off and its quite dusty inside – it did come from a bakery and referring back to how I got this machine, the rotary hoe looked cleaner. You might recall the very clean cage photos from the other machine – it appears to have led a charmed life (that's why I'm hoping the HDD is still viable). This machine has some additional cards – I can't be certain of their exact nature without extracting them at this stage – are they an add-on or are they part of the Model 16 upgrade?

In terms of rear ports this machine has the same ports. I have yet to compare the configuration of these ports to see if they are exactly the same as the Model 16b.

Well it sounds like there's plenty of fodder for future articles as I work towards getting these back up and running. As I write this article the first machine featured is out getting one of the power supplies repaired. It uses a Mode III power supply to run the drives and I opened the case about a week or two ago and it had a popped cap on the Model II supply. The main power supply appears to be quite tidy and intact.
For this machine, currently I'm describing it as a box of dust with some electronics inside.

KEVIN PARKER



USING NEWTON'S QUOTE "STANDING ON THE SHOULDERS OF GIANTS" AS INSPIRATION, I OFFER ANOTHER IN THE SERIES OF SOME OF THE KNOWN, AND LESSER-KNOWN NAMES WHO INSPIRED MUCH OF WHAT WE, TODAY, TAKE FOR GRANTED.

J LYONS & CO LTD

Lyons' Share

Commercial computing in Britain has its beginnings in a rather unexpected place

Electronic Office
Unlike all previous computers, which were designed for scientific or military applications, the LEO 1 was designed to perform only simple arithmetic operations, but on thousands of items or transactions per day

In 1947 a pioneering decision was taken to attempt to build a computer that could be used to automate clerical office work. This was to be the world's first commercial business computer. The imaginative decision came from a rather surprising source: J Lyons, the corner house teashop company. Lyons' business involved a large number of small transactions, and in order to make such a business operation profitable it was necessary to keep the paperwork under strict control. Even after the devastation of the Second World War the company employed more than 1,000 clerks for sorting out the receipts from the teashops.

Because of these considerations Lyons had a long tradition of innovation in business methods: they introduced calculating machines into their teashops as early as 1896 and by the 1930's they were experimenting with microfilm records of transactions. At the same time they set up the first business research centre to study operating methods.

Every few years Lyons would send representatives abroad to investigate new developments that might be useful to them, and in 1947 two employees were sent to the United States to look into the new 'electronic brains'. The most useful discovery they made was that a computer was being built much nearer home, in Cambridge.

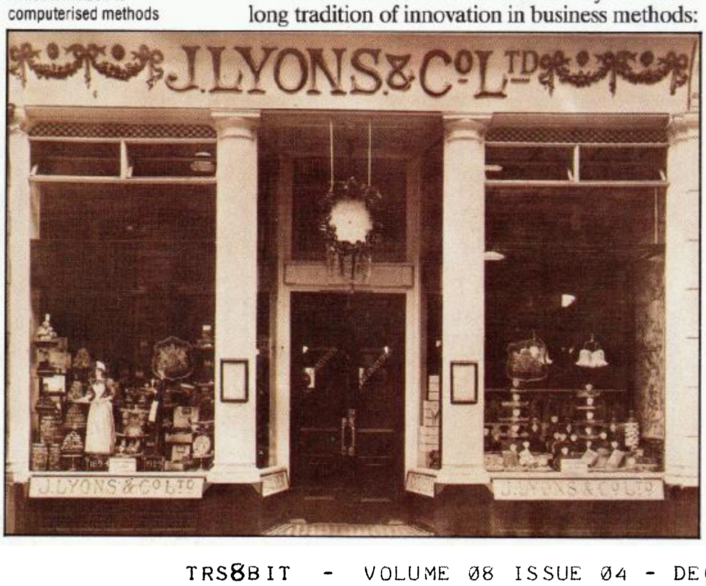
Lyons' board of directors commissioned a feasibility study to consider the possible development of the company's own computer. The estimate suggested that a computer could be constructed for £100,000 and that it would save £50,000 a year. Consequently, in October 1947 Lyons began work on the project. The enterprise was all the more daring because at that time the Cambridge computer was only at the design stage. Lyons gave a grant of £3,000 to Cambridge University to help build what became known as EDSAC (Electronic Delay Storage Automatic Computer). The grant was used to buy up government surplus valves. In 1949, EDSAC successfully completed its first job — to calculate a table of prime numbers.

Lyons analysed the problems their computer would have to solve, sketching out the routines that it would need. These studies became the blueprints for the first programs and helped determine the design of the hardware. But it soon became apparent that a business computer was fundamentally different from a university research machine. Whereas EDSAC was designed to execute long and complicated mathematical operations on an input consisting of just a few numbers, the business computer had to solve the opposite kind of problem. Mathematical operations were minimal — just adding and multiplying — but the amount of data processed was enormous.

LEO (the Lyons Electronic Office) became operational on 9 February 1954, and was used to calculate the weekly payroll for 1,700 members of staff. It performed in one and a half seconds an operation that had previously taken a clerk eight minutes.

LEO was a great success for Lyons, and they soon realised they needed more than one machine. Other companies expressed an interest and Lyons set up a company to use the skills they had learnt in manufacturing and marketing computers. Leo Computers was highly successful and went on to produce a series of improved versions of LEO that were used by industry, government and business. The company was bought up in 1963 by the English Electric Company.

Electronic Teashop
The traditional Lyons teashop seems an unlikely venue for the first major commercial application of computers, but it was precisely this kind of business, with its considerable number of small transactions, which lent itself to computerised methods



TRS-80 Emulators . com

TRS32: A Model I/III/4/4P Emulator For Windows

written by Matthew Reed

Unregistered Shareware Version:

- Works under all current versions of Windows
- Full Windows application — no low-level hardware conflicts!
- Model I, Model III, Model 4, and Model 4P emulation
- Four floppy disk drives (with optional realistic disk drive sound)
- Cassette tape drive with graphical on-screen controls
- Exatron Stringy Floppy emulation
- Printer support
- Serial port for RS-232 communications
- Joystick support (using a Windows joystick — TRISSTICK and Alpha Products joysticks are emulated)

Registered Version:

- All features included in the shareware version
- Built-in emulation of an Epson FX-80 dot matrix printer (including graphics and control codes)
- High resolution graphics (Radio Shack and Micro-Labs)
- Up to 1 megabyte of additional memory in Model 4 and 4P modes
- Hard disk support
- Orchestra 85/90 music generation

Actual Dotwriter output — emulated FX-80
click for larger sample

Interested?

- Read the TRS32 emulator documentation
- Download the shareware version
- Register online

IN MAV'S WORKSHOP

by Ian Mavric

UNDERSTANDING THE MODEL 4P GATE-ARRAY MOTHERBOARD

(NOTE: no, it's not deja-vu, this article borrows a lot from last issue's article on the M4GA motherboard, just updated for 4P application.)

When designing the Model 4 in late 1982/early 1983, Radio Shack took notice of what other computer makers like Kaypro, Osborne and even IBM were doing and decided that a relatively portable version of the Model 4 was just what busy managers needed. The result was a 10kg cream plastic box we know as the Model 4P. Looking for ways to cut production costs and improve reliability (that was not to say the 4P was unreliable) advances in electronics in the early 1980s led to an entirely new motherboard for the 4P around mid-1984. Using everything learned from the Model 4 Gate-Array motherboard, a new reduced chip count board combined arrays of TTL logic chips into single chips called gate-arrays. We (and Radio Shack, for that matter), call these Gate-Array motherboards, hereafter known as M4PGA boards. You can tell one of these straight away as the computer catalog number is 26-1080A.

A LOOK AT THE M4PGA BOARD

We see it has a number of large 24, 28, or 40 pin chips which are logically laid out and while the board may look imposing, especially when repairs are needed, if you know what each part does you can logically track down problems fairly quickly.

U45: Z80A CPU - Needs to be a Z80A, I have seen boards which have a plain Z80 which is not fast enough to run at 4MHz in Model 4 mode for any length of time. If your computer won't start check for CLK signal on Pin 6.

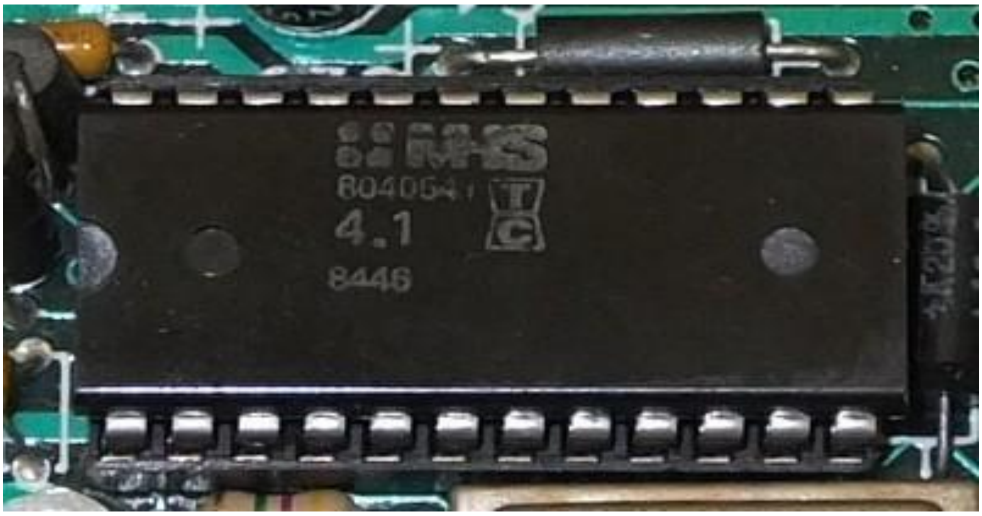
U69 (ROM): Boot functions, memory test, RS232 Boot. When we upgrade a M4PGA to FreHD auto-boot, U69 is replaced with a programmed EPROM.

U106 8040542 Gate Array "4.2":

Address Decoder - if the computer won't get past a junk screen, and you know the memory chips are all OK and that there is a CLK signal on Pin 6 of the Z80A, suspect this chip.

Y1 and U146 (Crystals): Y1 is for Model III mode (2.02MHz) and U146 is for Model 4 mode (4MHz). I have seen systems where Model III mode works but Model 4 mode doesn't - replace U146. If Y1 doesn't work then the computer will not start at all.

U148 8040541 Gate Array "4.1":



Clock switchover chip controls which frequency is used in which mode. If this chip doesn't work then the computer may run slow (2MHz) in Model 4 mode, or the system will just not start up.

U42 8040045: CRT Controller - controls the video sync on the screen - if you have lost horizontal or vertical sync that can't be adjusted by the controls on the Video PCB, then suspect this chip.

U82 SRM2016C: Video RAM chip - if you see spelling errors on the screen or characters is missing but the software otherwise works properly, suspect this chip.

U102 8040543 Gate Array "4.3":

Video Gate Array - works in conjunction with U42, U82, and U103 to display information on the CRT. When this chip fails, nothing will be visible on the CRT, even though the computer still runs and responds to commands.

U103 8049007 Character Generator: forms all the letters, numbers, or symbols you see on the screen. If the characters are malformed or missing dots, suspect this chip.

U73 WD8116: Baud Rate generator - if the serial port doesn't work or if some of the baud rates work and others don't, suspect this chip. But check other RS232 problems below on chips U31, U33 and U70 (below).

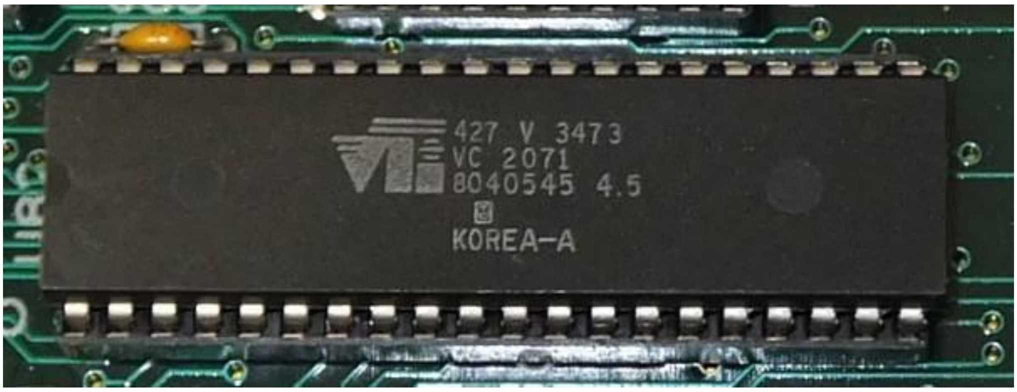
U17 WD1773 Floppy Disk Controller chip: industry standard double density floppy disk controller. If the computer boot known good disks from known good disk drives (as tested on another computer) then suspect this chip or U18.

U18 8040544 Gate Array "4.4":

Disk Drive Interface Oscillator - critical timing array to make U17 work. If U17 is known good (as tested on another computer) but the disk drives still don't work right, then suspect this chip.

Y2 (Crystal): needed for floppy disk system (U17, U18) to work. If it doesn't work then neither U17 nor U18 will do anything.

U31 8040545 Gate Array "4.5":



RS232 support gate array, controls the interface between the RS232 serial port (U70, U33, U11, U12, U13) and the processor. If diagnostic software reports no RS232 port installed, suspect this chip.

U33 TR1865PL: UART - Universal Asynchronous Receiver Transmitter which is the heart of the RS232 port. Easy to blow this chip by plugging in poorly designed RS232 devices or any device which sends uncontrolled voltages into the RS232 port.

THE LEGACY OF THE GATE-ARRAY MOTHERBOARD

The M4PGA motherboard was a great improvement over the M4PNGA board which had issues synchronising the Model 4 mode 80x24 display (which was achieved with a trimmer capacitor which tended to drift off over time). It was a pity that just as Radio Shack perfected the 4P machine, and included a decent green screen CRT to make a really professional looking machine, they decided to go in the direction portable computing was taking with the Model 100 and quietly dropped the 4P from production. It says something about the good design that many M4PGAs come up for sale on places like eBay and almost always work perfectly.

Ian Mavric

ianm@trs-80.com

USEFUL PROGRAMS FROM THE PAST.. STILL IN USE TODAY

ZEN EDITOR ASSEMBLER - LAURIE SHIELDS SOFTWARE

NOW I KNOW WE ALL HAVE OUR 'FAVOURITE' WHEN IT COMES DOWN TO IT, BUT THERE WERE 3 MAIN REASONS I TOOK TO ZEN FROM THE OFF. THEY WERE -

1. THE AUTHOR.
A LONG-TIME FRIEND (AND HERO) OF MINE, LAURIE SHIELDS. ZEN WAS ALWAYS A 'WORK IN PROGRESS' AND UPDATES AND IMPROVEMENTS WERE ALWAYS READILY AVAILABLE, IF NOT FOR FREE THEN FOR A VERY MODEST CHARGE. WE ALL WAITED WITH BAITED BREATH, LOOKING FORWARD TO HEARING WHAT NEW 'GOODIES' LAURIE HAD GOT FOR US THIS TIME. LAURIE SPENT MANY HOURS LECTURING ON ASSEMBLER PROGRAMING TO THE NATGUG USER GROUP AND WOULD ALWAYS BE HAPPY TO ASSIST WITH ANY PROBLEMS YOU WERE ENCOUNTERING.

2. THE ACTUAL WORKINGS OF THE PROGRAM.
ZEN MADE LIFE SO MUCH EASIER FOR ME WITH MY CASSETTE BASED 16K MODEL 1, LEVEL 2 SYSTEM. ZEN MEANT THAT I COULD LOAD IN OR WRITE A PROGRAM AND ALL MY ASSEMBLER WORKINGS (MEAGRE AND BUG-RIDDEN, AS THEY USUALLY WERE) WOULD ASSEMBLE, ALLOW FOR DEBUGGING (WITH BREAK-POINTS) AND OUTPUT AND LOAD INTO MEMORY WITHOUT HAVING TO MESS ABOUT WITH THE TAPE RECORDER, UNTIL I WANTED TO SAVE EITHER THE SOURCE OR OBJECT CODE. IT ALSO GAVE ME THE OPTION, AT THIS POINT, TO VERIFY ANY SAVES I HAD MADE. (WORTH IT'S WEIGHT IN GOLD FOR THAT REASON ALONE, WHEN USING MY CRT-81 TAPE RECORDER!). IF I REMEMBER CORRECTLY, IT COULD ALSO READ EDTASM AND MON3 SYMBOLIC DUMP TAPES AND CONVERT THEM TO ZEN FORMAT AUTOMATICALLY.

AND FINALLY -

3. @ZEN.
LAURIE PRODUCED A VERSION FOR THE ACULAB FLOPPY TAPE. I REMEMBER, AT THE TIME, THINKING IT WAS THE BEST £19.50 I'D SPENT FOR AGES! IN THE 1980'S, WHAT MORE COULD YOU ASK FOR, OTHER THAN, @PENCIL AND @ZEN. (WITH ONLY @CALC MISSING. I DON'T THINK LAURIE EVER GOT AROUND TO CONVERTING VISICALC FOR THE FT!)

FOR ANY ASSEMBLER PROGRAMING I DO TODAY, IT HAS TO BE ZEN! IF YOU WOULD LIKE TO TRY THE LATEST, ZEN85, VERSION ON YOUR EMULATOR, (TRSDOS, NEWDOS82 AND LDOS COMPATIBLE), THERE IS A DOWNLOAD AVAILABLE ON THE WEBSITE.

*** IF YOU HAVE A FAVOURITE PIECE OF SOFTWARE, WOULD *** YOU LIKE TO TELL US ABOUT IT?


ZEN
VERSION 3.0


ZEN85
EDITOR -:- ASSEMBLER
(C) AVALON SOFTWARE
1985 LAURIE SHIELDS
A ASSEMBLE
B BYTE FIND
C COPY MEMORY
D DOWN # LINES
E ENTER EDITOR
F FILL MEMORY
G GLOBAL REPLACE
H HOWBIG SOURCE
I INCREASE S.T.
J JUMP TO MEMORY
K KILL FILE
L LOCATE STRING
M MODIFY MEMORY
N NEW LINE (EDIT)
O RETURN TO DOS
P PRINT SOURCE
Q QUERY MEMORY
R READ SOURCE
S SORT S.T.
T TARGET LINE #
U UP # LINES
V VERIFY CASSETTE
W WRITE SOURCE
X XAMINE REGISTERS
Y TOGGLE DISPLAY
Z ZAP # LINES
DI DIRECTORY
RI READ & INSPECT
RL READ LIBRARY
REF REFERENCE LIST
*** TOM BOURKE ***
*** VERSION 6.2 ***