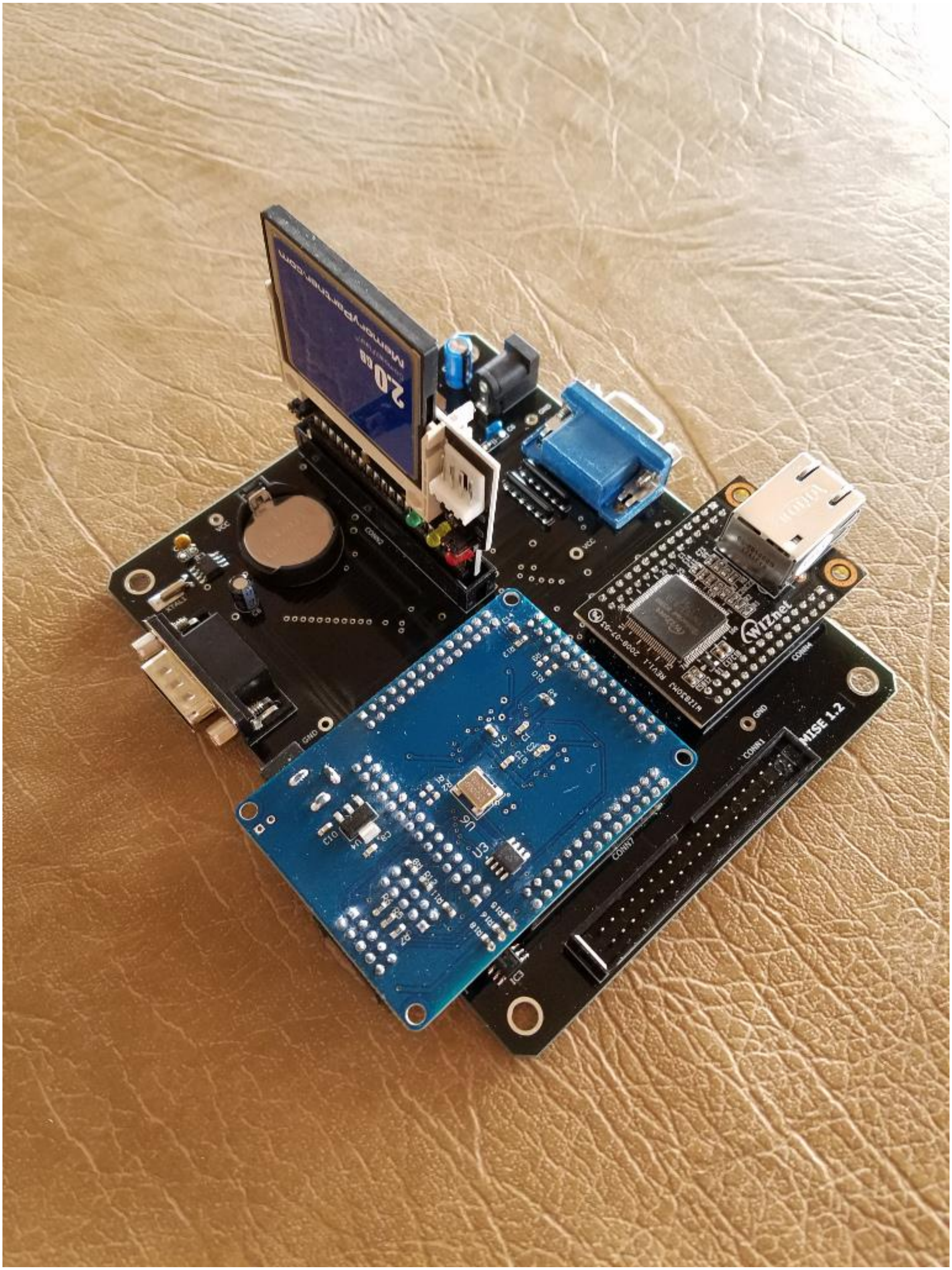
2.0GB
MemoryPartner.com
WIZnet
MISE 1.2
CONN1
CONN4
CONN7
GND
VCC
XTAL1
U3
U6

# Table of Contents

**Setup Instructions**

MISE comes partially disassembled for shipping. No electronic knowledge or experience is necessary to put it together. No soldering is required.

1. **Important:** Before opening the anti-static bags containing the MISE components, ground yourself with an antistatic wrist strap to avoid damaging any of the semiconductor devices. If possible, place the MISE on an antistatic mat. At the very least, touch a grounded surface (such as a metal PC case) to drain off any static electricity before proceeding.

2. Place the MISE on a non-hard surface. If not pre-installed, insert the CR2032 battery into the holder. The positive (+) side (the one with the writing on it) should face upward.

3. If not pre-installed, insert the FPGA module (the blue board) into the four 28-pin headers, making sure to align the pins properly. The outermost 10-pin programming box header should extend off the side of the board that has the 9-pin joystick connector. Press the module down until there is no gap on any of the four sides.

4. Mount the WIZ830MJ Ethernet module on the two receptacles, pressing it down until there is no gap. The RJ45 connector should extend off the side of the board.

5. If the Compact Flash card is not already inserted into the adapter, do so after reading the warning here. The card goes on the side of the adapter with the three LEDs, with its label facing outward. If you have a secondary CF card you'd like to use for system backup, insert it carefully into the socket on the back side of the connector. Insert the adapter into the 40-pin box header (the one that does not run along the edge of the PCB) being careful to align the bump on the adapter to the notch on the header.

6. Double-check that all submodules (FPGA, Ethernet, IDE/CF adapter) are inserted properly. It's very easy to mistakenly insert them one pin spacing off. Despite the notch, the IDE/CF adapter can easily be inserted backwards.

7. Push the four plastic board feet through the holes at the corners of the MISE PCB from the solder side.

8. Clean any oxidation off all the Model I keyboard and Expansion Interface edge connectors using DeoxIT or a nonabrasive pencil eraser (aka deoxit classic.)

9. If you are using a TRS-80 Expansion Interface, connect the ribbon cable's card edge connector to the Expansion Interface's buffered expansion port. The ribbon cable should exit from the bottom of the connector, and the colored stripe along the cable should be toward the back. If you are connecting the MISE directly to the Model I keyboard, the colored stripe should be on the right with the keyboard facing you. Plug the other end of the ribbon cable assembly into the 40-pin box header that runs along the edge of the MISE PCB, being careful to align the bump on the connector to the slot on the header. Press down firmly.

10. Connect your VGA and Ethernet cables to the MISE. Connect your Atari 2600 compatible joystick to the 9-pin DSUB connector.

11. Connect the AC adapter to the MISE power jack. The AC adapter should not be powered on until you are ready to use the MISE. It is strongly recommended that you plug the AC adapter, your TRS-80, Expansion Interface and peripherals, and your VGA monitor into a single power strip that is switched off when the TRS-80 is not in use. If this is not possible, unplug the AC adapter when not in use rather than just disconnecting it from the MISE.

12. Turn on your power strip, or power up the MISE and floppies/other peripherals. Then switch on the Expansion interface. If you wait until your VGA monitor warms up, you'll see a MISE splash screen. Switch on the keyboard.

The MISE should now boot LDOS. If the video is shifted, blank, or doesn't look right, try using your monitor's "auto" or "sync" function. Sometimes this a button, sometimes it's in a menu.

If you are using a widescreen VGA monitor, the video will probably appear to be "stretched" horizontally. You will need to figure out how to set your monitor to 4:3 display mode to reproduce the original TRS-80 aspect ratio.

13. If you are using MISE attached directly to the keyboard (no Expansion Interface), read the instructions here.
14. Prepare for system recovery:
    - Make a set of system floppies. ○ If you have a second CF card, make a backup of the primary hard disk image and put it somewhere safe.
    - If you have a CF card reader/writer device, dump a raw image to a file on your PC using a tool like this.

Don't become overly obsessed with this stuff--as far as I know, no MISE customer has needed system recovery yet. If your CF becomes corrupted, it's easily fixed or replaced.

15. At this point you might try typing DO AUTOEXEC to configure the VGA colors and network module. If this works properly, you can type the command AUTO DO AUTOEXEC to run this script on every boot. Edit the file AUTOEXEC/JCL if you wish to modify its commands. It is strongly recommended that you always at least run DHCP at boot, to set up the network adapter. Hold down <ENTER> while booting to bypass any AUTO command.
16. If your Model I has a lowercase modification (you can tell by whether lowercase letters appear in the date displayed when you booted) try running LOCASE/CMD to enable LDOS's lowercase driver. If this works properly (you see lowercase letters in "LDOS Ready", for example) use the command SYSTEM (SYSGEN) to store the new configuration. If instead you see digits and punctuation where there should be lowercase letters, run UPCASE/CMD to disable the lowercase driver. See the section on using the VGA display for more information.
17. System powerdown sequence: Remove all floppy diskettes. Switch off the keyboard, then the Expansion Interface. Finally, power down the MISE and other peripherals.

**Notes**

- The Compact Flash cards are **not** hot-pluggable. Do not insert or remove CF cards with the MISE or TRS-80 powered on. Doing so will corrupt the filesystems.
- The Compact Flash filesystems are native TRS-80 and therefore not compatible with PC filesystems. If you plug them into a PC card reader you will not be able to see the files. However, you can make TRS-80 image backups on your PC by reading them in raw mode. To transfer files between the PC and TRS-80, use the Ethernet connection.
- When powering up or rebooting the MISE, you might notice that attached floppy drives spin briefly. This is an intentional result of the MISE boot sequence initializing the floppy control logic in the Expansion Interface.
- To boot from floppy, bypassing the hard drives, hold down <SHIFT> while switching on the keyboard or pressing the reset button. To boot directly to Level II BASIC, hold down <BREAK> instead.
- The Compact Flash to IDE adapter has two jumpers (labeled JP1 and JP2) which are correctly configured when shipped. Should they ever be moved, their proper positions are both to the left with the 40-position connector facing you.

- There's an Ethernet activity LED on the underside of the FPGA module. Yes, it's hard to see. Another LED illuminates when the joystick's "fire" button is pressed, and the third LED when the reset button on the FPGA module is pressed.
- When disconnecting the ribbon cable assembly from the MISE, be very careful to pull *straight upward* on the entire width of the cable. Pulling up one side or the other can break the side walls of the box header and bend its pins.
- For international customers who can't use the US-plug power supply, contact me and I will exclude it from your order and credit you my cost on that item. You will need a power supply with a 2.1mm ID, 5.5mm OD, centerpositive female plug. The ones shipped with MISE are 5V switching supplies that provide 2A. Higher voltages, lower supply currents, and unregulated power supplies would probably work, but have not been tested with MISE. The supply feeds a 3.3V linear voltage regulator which powers everything on MISE.

## Network Configuration

### MAC Address

The MISE requires an Ethernet hardware address, known as its MAC (Media Access Controller) address. This address consists of six hexadecimal bytes separated by colons. It is configured in the hardware section of the NETWORK/CFG file:

```
        [Hardware]
        # Bit 1 of 1st byte set -> locally administered address
        # Other 5 bytes can be anything locally unique
MAC Address = 02:00:00:00:00:03
```

As shipped, the MISE uses a *locally-administered* MAC address, which is indicated by setting the second least significant bit of the first byte. This address must be unique on your local network. If you wish to continue using a locallyadministered MAC address and the one supplied is not unique, simply modify the file so that it is.

If you wish to use a *globally-unique* MAC address, the GETMAC utility will display the address assigned to your MISE. Enter this address into the NETWORK/CFG file. This step is normally not necessary if you have a home router, which replaces the MAC addresses of devices on your local network with its own before sending packets to your Internet Service Provider. If, however, you have the temerity and capability to attach your TRS-80 directly to the Internet, you might need the globally-unique address.

Either way, the MAC address is read, and the Ethernet module is configured, by the DHCP/CMD utility. This applies whether or not you have a DHCP server. See the section on setting your IP address for details.

Whenever you make changes to your NETWORK/CFG file, you should copy the new file to your MISE4 system floppy.

## IP Address and DHCP/CMD

The MISE requires a Internet Protocol (IP) address for all TCP or UDP communication such as FTP or Telnet. The address is configured in the Ethernet module using the DHCP/CMD utility:

```
DHCP [-Q] [-F] [-N]
```

If you have a Dynamic Host Configuration Protocol (DHCP) server available (usually your home router or your Internet Service Provider) simply typing DHCP will configure the MISE and show you the results. This will also create a file called DHCP/CFG with all the information obtained from the server. The -Q (Quiet) switch will prevent displaying the results. The -F (Force) switch will, if DHCP negotiation fails, use the values from the existing DHCP/CFG file.

If you do *not* have a DHCP server or do not wish to use one, you can manually set the IP address, subnet mask, and gateway address in the [IPv4] section of DHCP/CFG. Then invoke DHCP/CMD with the -N (No server) switch. This will read the file and configure the Ethernet module.

Once you've successfully run DHCP/CMD, you can test that it's working properly by typing "PING <ip address>" in a Windows command prompt or Linux terminal window.

It is recommended that you set up your local router to use a static IP address for your MISE. See your router's manual for instructions on doing so. A static address will allow you to give the MISE a name that can be used from your PC instead of needing to know its currently-assigned IP address. Under Windows, you can edit the file:
C:\WINDOWS\System32\drivers\etc\hosts

to look something like this:

```
127.0.0.1       localhost
192.168.1.1     router
#192.168.1.2    PC
192.168.1.3     TRS-80
192.168.1.5     printer
```

after which you can access the TRS-80 by name. The Linux version of the file is /etc/hosts. The same file can be used on your Model I; see here for more information.

Note that the IP address of my PC, the one with this hosts file, is commented out with the # character. Windows 7 (and probably some other OSes) get confused when they see their own IP address in the hosts file, possibly resulting in long startup times and failure of some services (most notably the Windows 7 FTP server.)

There are some DHCP options that can be modified in the NETWORK/CFG file:

```
[DHCP Client]
Config File = "DHCP/CFG"
Hostname = "TRS-80"
Timeout = 2
Max Retries = 1
```

The Hostname variable is reported to the server during DHCP negotiation.

Here's a sample DHCP/CFG file.

Whenever you make changes to NETWORK/CFG or DHCP/CFG, you should copy the new file to your MISE4 system floppy.

## HOST/TXT file for static addresses

Your MISE comes with a file called HOSTS/TXT:0 that contains an example list of static IP addresses:

```
      127.0.0.1       localhost
      192.168.1.1     router
      192.168.1.2     PC
      192.168.1.3     TRS-80
      192.168.1.5     printer
```

127.0.0.1 is called the *loopback* address and shouldn't be changed. The addresses starting with "192.168" are all local network addresses, assigned by a local router. If you browse your router's URL (in my case http://192.168.1.1) you should be able to set up static IP addresses for your local devices. Consult your router's manual for specific instructions. This will allow the MISE to address all such devices by name using DNS, rather than by their currently-assigned IP address.

Whenever you make changes to your HOSTS/TXT file, you should copy the new file to your MISE4 system floppy.

With one important exception, this file can have the same contents as the one used by your PC. See the section on setting your IP address for more information.

The contents of the HOSTS/TXT file are used by the dnshost assembly module, which you can link to your own custom applications. See the sections on DNS/CMD and DNS2/CMD for example applications.

## Sample NETWOTK/CFG file

```
[Hardware]
# Bit 1 of 1st byte set -> locally administered address
# Other 5 bytes can be anything locally unique
MAC Address = 02:00:00:00:00:03

[TFTP Server]
Timeout = 2
Max Retries = 7

[DHCP Client]
Config File = "DHCP/CFG"
Hostname = "TRS-80"
Timeout = 2
Max Retries = 2

[DNS Client]
Hosts File = "HOSTS/TXT"
Max Retries = 3

[FTP Server]
MOTD File = "MOTD/TXT"
Timeout = 900
Floppy = Yes
System = No
Invis = No
Group = "users"

[FTP Client]
Max Retries = 6
Login = "anonymous"
Directory = "/"        # default GET/PUT logical drive
```

```
[SNTP Client]
Server = "pool.ntp.org"
#Server = "time.nist.gov"
#Server = "time.windows.com"
Timeout = 4
Max Retries = 3
Offset = -6     # -7 IS MST; MDT is -6
```

### Network Applications FTPD/CMD and FTP1D/CMD, the File Transfer Protocol servers

FTPD/CMD and FTP1D/CMD are full-featured File Transfer Protocol (FTP) servers. When running, they provide Ethernet file transfer capability on the network.

```
FTPD [-L <log_level>] [-I] [-S]
FTP1D [-L <log_level>] [-I] [-S]
```

FTPD allows up to four simultaneous connections from clients, and requires a 48KB Model I. FTP1D is the same as FTPD, but only allows a single connection. This permits FTP1D to be used on a Model I with only 32KB of RAM.

Press <BREAK> to exit FTPD.

When connecting to the server, the root directory "/" contains numbered virtual directories, one for each TRS-80 hard drive partition or floppy disk drive. Files cannot be written to the root directory. Within each drive's virtual directory, all files are "owned" by whatever name you logged in with, and the group specified in the NETWORK/CFG file. The slash character (/) between filenames and filename extensions is converted to a dot (.) by FTPD.

In order to access password-protected TRS-80 files, log in using their password or the LDOS system password RS0LT0FF (zeros not the letter 'O') which can be used to access any file.

Within text-based FTP clients, filename wildcarding is done LDOS style--without using asterisks. For example, to list all filenames starting with the letters "ba", the command "dir ba" is used. To download all such files, "mget ba". To list all executables, "dir .cmd".

The file MOTD/TXT (Message Of The Day) is sent to FTP clients when connecting to the server as a login greeting. Some clients display its contents, some do not.

The -L argument specifies the verbosity of log messages displayed on the screen. Level 0 displays almost nothing and level 3 displays pretty much everything that happens. The default log level is 1.

The -I and -S switches specify that the server should by default show Invisible and/or System files in directory listings. These switches override the corresponding settings in the NETWORK/CFG file.

You can connect to the server using Windows Explorer and many other applications. In the address box, type in "ftp://<your MISE IP address>", where the IP address is specified in your DHCP/CFG file. If you've correctly set up your PC's hosts file, you could also use something like "ftp://trs-80", or with an account and password, "ftp://username:rs0lt0ff@trs-80".

Be aware that text file line endings are encoded differently on the TRS-80 than the PC. Using a command-line FTP client, you can specify ASCII mode and the software will perform the translation. Windows Explorer doesn't provide this functionality, so if you download a TRS-80 text file to your PC and look at it in Notepad, it'll look wrong (although Emacs handles it properly). Some clients like Filezilla perform Netascii translation based on the filename extension.

**Important note:** If you use Filezilla (and this probably applies to some other client software) you must turn off the transfer timeout, because the TRS-80 is far slower than the client expects. Under Edit->Settings->Connection, set the timeout value to 0 to disable it.

PC network Firewall software can sometimes interfere with FTP communication. If you have problems, make sure that TCP ports 20 and 21 are unblocked. I've noticed that under Norton 360, blocking these ports seems to be the default. To override this, you need to navigate to Settings->Firewall->Traffic Rules and Add a new rule to allow traffic on TCP ports 20 and 21.

## FTP/CMD, the File Transfer Protocol client

FTP/CMD is a full-featured FTP client which allows you to connect to FTP servers in order to store and retrieve files. At least 32KB of TRS-80 RAM is required.

```
FTP [host-name [IP-port]]
```

A DNS lookup of *host-name* will be performed if required. *IP-port* defaults to the standard FTP control port number 21. If no arguments are specified, FTP enters command-line mode.

The MISE FTP client behaves very much like a standard UNIX-style version. Doing a web search for "man ftp" should provide enough information to get started. The commands *?* and *help* display a list of implemented commands. You can use *help <command>* to get a brief description of a specific command.

Local TRS-80 file and disk drive names are converted to UNIX-style pathnames. A virtual root directory "/" contains a numbered subdirectory for each TRS-80 disk drive, and slashes in TRS-80 filenames are converted to periods. For example */0/readme.txt* refers to the TRS-80 file README/TXT:0. Use the *lcd* command to set the current local directory, for example *lcd /5*. If you need to access password-protected TRS-80 files, set the local password with the *lpass* command.

When using the single *get* and *put* commands, you can specify a different local or remote filename (and/or path) respectively as a second argument. This allows flexibility in dealing with LDOS's restricted filenames.

When using the multiple-file commands *mget* and *mput*, you can alter individual filenames if you are in prompting mode (see the output of the *status* command.) When asked something like *mget longfilename.longext?* during an *mget* command for example, you can answer "*y <local-filename>*". Likewise for *mput* where you can specify a different remote filename after the *y*.

Pattern matching, for example with the multiple-file commands *mget*, *mput*, and *mdelete*, always uses the method implemented on the machine where the files exist. So if you are connected to a UNIX-like server, you can probably use regular expressions like *mget m*.cmd* to retrieve all remote files whose names start with *m* and end with *.cmd*. When referring to TRS-80 local files, however, LDOS-style pattern matching is used. You would therefore type *mput m.cmd* (no asterisk) to prospectively store all local files whose names start with *m* and end with *.cmd*. Since this could cause some misunderstanding about exactly which files you wish to store, you can turn off local filename "globbing" with the *glob* command.

As with any FTP client, be careful to properly set *ascii* or *binary* mode before transferring files!

Here are the settings available in your NETWORK/CFG file (with their default values):

```
[FTP Client]
Max Retries = 6
Login = "anonymous"
Directory = "/"          # default GET/PUT logical drive
```

## SNTP/CMD, the Simple Network Time Protocol client

SNTP/CMD is a Simple Network Time Protocol (SNTP) client. It queries the SNTP server specified in your NETWORK/CFG file and displays the result. This is *not* a full implementation of NTP, but it's close enough for a TRS-80.

```
SNTP [-S] [-H] [-Q]
```

The -S switch causes the Software (LDOS) clock to be set.

The -H switch causes the Hardware clock (RTCC) to be set.

The -Q switch suppresses displaying the date and time obtained from the server.

Routers and PC network Firewall software can sometimes interfere with SNTP communication. If you have problems, make sure that UDP port 123 is unblocked.

SNTP/CMD does not deal with time zones or Daylight Savings Time. Instead, the NETWORK/CFG file contains an entry in the [SNTP Client] section called "Offset". You should set its value to your offset from Greenwich Mean Time (GMT). You'll need to correct it by an hour each time Daylight Savings Time begins or ends.

## TELNET/CMD, the Telnet client

The Telnet client allows you to logon to remote servers from your TRS-80. A minimum of 32KB RAM is required.

TELNET [-Q] [HOST [PORT]]

If HOST is omitted, TELNET will enter command-line mode rather than terminal mode. Type ? or HELP in command-line mode for a list of commands.

You can enter command-line mode from terminal mode by pressing <BREAK>. While connected to a server, simply press <ENTER> to reenter terminal mode from command-line mode.

The control key combination is <SHIFT><DOWN-ARROW>. For example you can enter CONTROL-C (interrupt) with <SHIFT><DOWN-ARROW>C. Be prepared for carpal tunnel syndrome.

It is recommended that you enable the advanced LDOS keyboard driver KI/DVR when using TELNET, to allow entering additional characters such as <CLEAR><ENTER> for underscore. For TELNET this applies whether or not you have a lowercase modification.

It is also recommended that you enter 80-column display mode prior to running TELNET, since many Telnet servers assume that as a minimum display width.

**DIRE WARNING:** This Telnet client does not implement credentials authorization or secure transmission of data. More precisely, when receiving a DO-AUTHENTICATION Telnet option, this client responds with WONT-AUTHENTICATION, thus rejecting the opportunity for secure communication. Otherwise Kerberos authentication would have to be implemented on the Model I, a total impracticality and likely impossible. What this means is that all data received and sent using TELNET, including account names and passwords, is transferred as plain text. This information can easily be intercepted and used by malicious third parties to access your accounts. You will have to hit <ENTER> when starting TELNET following a disclaimer to confirm your understanding of and agreement to this. You may use the -Q (Quiet) switch to bypass the warning once you are familiar with it. It is ***imperative*** that you not use TELNET to access servers outside your secure LAN for anything but trivial purposes, using accounts you don't care about, and using passwords that are not shared with any of your non-trivial logins.

Known issues:

- Insecure data transmission.<br>
  Terminal line mode not implemented, only character mode.<br>
  Missing command-line mode commands: logout, display, mode, send, set, unset, status, toggle, slc.<br>
  Many ANSI escape sequences not implemented.<br>
  Many Telnet options (out-of-band communication sequences) not implemented.

The default TCP timeout is a bit long at 31.8 seconds, appears to hang unless you wait long enough when connecting to or disconnecting from an unresponsive server.

## TFTPD/CMD, the Trivial File Transfer Protocol server

TFTP is a simple file transfer protocol that uses UDP, not TCP communication like FTP. In order to use it to communicate with your TRS-80 you'll need client software such as the TFTP program that's included with some versions of Windows.

```
TFTPD [-O]
```

The -O switch allows TFTPD to overwrite TRS-80 files. Press <BREAK> to exit TFTPD.

The slash character (/) between filenames and filename extensions is converted to a dot (.) by TFTPD. If you wish to specify a TRS-80 drive number when accessing a file via TFTP, prepend the drive number like this:

```
TFTP> get /0/readme.txt
```

Be aware that text file line endings are encoded differently on the TRS-80 than the PC. Using a command-line TFTP client, you can specify ASCII mode and the software will perform the translation.

Network Firewall software can sometimes interfere with TFTP communication. If you have problems, make sure that UDP port 69 is unblocked.

## DNS/CMD and DNS2/CMD, Domain Name Service client lookup demos

The DNS/CMD demo program performs Domain Name Service (DNS) lookups and inverse lookups.

```
DNS <host name>
DNS -I <IP address>
```

Without the -I switch, DNS/CMD attempts to find IP addresses associated with the given host name. First it checks in the HOSTS/TXT file using the **dnshost** assembly module. If no matching entry is found, the DNS server listed in DHCP/CFG is queried using the **dnslook** assembly module, and all matching IP addresses are listed.

The -I switch performs an *inverse* DNS lookup. That is, it attempts to find a local or fully-qualified domain name for the given IP address.

DNS2/CMD is a simplified version of DNS/CMD. It simply returns the *first* matching IP address found. The assembly code for interfacing with the supplied assembly modules is extremely simple.

```
DNS2 <host name>
```

DNS2 does not support inverse DNS lookups.

Source code for both DNS demos is included. To build new versions of the programs using the MRAS assembler, enter these commands:

```
    ATTRIB DNS/CMD.UTILITY:0 (PROT=FULL)
    MRAS DNS
DO DNS
```

The "DO" command runs a linker script that outputs the executable program file. Replace "DNS" with "DNS2" to build that demo instead.

## TELSRV/CMD, a Telnet server demo

TELSRV/CMD is a demonstration of using the Telnet protocol to communicate with the TRS-80 Model I through MISE. To use it you'll need telnet client software such as the TELNET program included with some versions of Windows, or PuTTY. Up to eight clients can connect to the server simultaneously.

TELSRV listens on IP port 1978 (a good year for the Model I) not the default port 23. This port must not be blocked by firewall software. The command to connect might look something like this:

```
> telnet trs-80 1978
```

After a brief greeting, clients can enter the following commands:

HELP or ? : list the commands

WHO : list users' socket numbers and IP addresses

SAY : send a message to all users

QUIT : disconnect from server

Source code is included, allowing easy modification of the server's command set. The table of commands and their subroutines are all found near the end of TELSRV/ASM. To build a new version of the program using the MRAS assembler, enter these commands:

```
        ATTRIB TELSRV/CMD.UTILITY (PROT=FULL)
        MRAS TELSRV
DO TELSRV
```

The "DO" command runs a linker script that outputs the executable program file

## Making MISE system floppies

It is critical to make system floppies for your Model I in case hard disk :0 becomes unbootable for some reason.

### Making the floppy disks

MISE comes with four JV1-format floppy images on drive :0, called MISE1/JV1...MISE4/JV1. These are for single-sided, single-density floppies to allow their use with any Model I that has at least one disk drive attached.

First format four single-density, single-sided, 35-track floppy diskettes. Then dump the images onto them using the JV1COPY utility:

```
JV1COPY MISE1/JV1 :6
...
JV1COPY MISE4/JV1 :6
```

and label the floppies.

- MISE1 (bootable) is the first of the 2-diskette LDOS system set, updated with the "year 2012" patches.
- MISE2 (non-bootable) is the second LDOS system floppy.
- MISE3 (bootable) contains utilities and drivers to set up the CF hard drives.
- MISE4 (non-bootable) contains utilities and files needed to run FTP.

If you have a double-density adapter, you can combine these four floppies into one bootable double-density, doublesided diskette.

## Making hard disk boot floppies

A hard disk boot floppy allows access to the hard disk system while booting from a floppy diskette. This may be useful in situations where hard drive :0 becomes unbootable.

Make another copy of the MISE1 LDOS system diskette, and then write the hard disk configuration to it:

```
JV1COPY MISE1/JV1 :6

SYSTEM (SYSGEN,DRIVE=6)
```

You can now boot this floppy by holding down <SHIFT> during reset. However, this floppy will not be bootable on a nonMISE system, or a MISE system with a different drive configuration.

### Lowercase Effects

When booting from a floppy that lacks CONFIG/SYS--generated by the SYSTEM (SYSGEN) command--LDOS auto-detects lowercase hardware modification and sets its display driver appropriately. Booting with CONFIG/SYS present, however, LDOS restores the display driver that was in effect when SYSTEM (SYSGEN) was executed. This applies to both hard and floppy disks. See the section on Using the VGA Display for information on selecting the proper display driver.

## Compact Flash Adapter

The Compact Flash (CF) adapter supplied with MISE is inserted into an IDE box header; you could use an IDE cable and electromechanical hard drive instead if you wanted. You'd need an IDE cable with the "key" pin (pin 20) unblocked-many cables have that pin blocked to ensure the cable is inserted correctly, but MISE uses it to power the CF adapter. You'd need an external power supply for the IDE drive.

The adapter has 3 LEDs on the front side: red (power), yellow (hard drive activity), and green (CF card inserted).

The adapter has two CF sockets, for Primary and Secondary devices. The Primary CF device (the one that LDOS boots from) is inserted into the CF socket on the side with the LEDs. The Secondary device (if any) is inserted on the other side of the adapter.

The jumper labeled CF1 on the adapter will, if shifted to the rightmost position, swap the Primary and Secondary devices.

With two CF cards inserted into the adapter, you may notice that the yellow LED is illuminated when reset to the MISE is removed (usually when the Expansion Interface is first switched on.) This is normal--the signal used on the IDE interface to indicate the presence of the secondary device is shared with the activity indicator. Once the MISE is booted, the yellow LED will function normally.

**Important:** The pins of the CF sockets are extremely delicate. If you feel any resistance when trying to insert or remove a CF card **stop immediately**. Check carefully for bent pins. Trying to force a CF card in or out can cause the pins to become unseated from the back of the socket! If the socket is simply too tight, a **gentle** rocking motion can sometimes help to insert or remove a CF card. The connectors for the Secondary cards tend to be particularly tight.

## Hard disk organization Logical Organization

LDOS reads and writes raw hard drive data in 256-byte *sectors*. When addressing a sector, software provides a 10-bit *cylinder* and 8-bit sector number. Thus each logical hard drive can have up to 1024 cylinders of up to 256, 256-byte sectors each, totalling as much as 64MB.

*Files* are allocated in groups of 32 sectors known as *granules* or *clusters*, each of 8KB size. So each cylinder can have up to 8 granules, and must have a number of sectors that's a multiple of 32. Each LDOS hard disk directory can hold up to 256 files.

As shipped, MISE's disks are organized as follows:

Drive :0 -- Hard disk, 170 cylinders 8 heads, 10.625 MB
Drive :1 -- Hard disk, 170 cylinders 8 heads, 10.625 MB
Drive :2 -- Hard disk, 170 cylinders 8 heads, 10.625 MB
Drive :3 -- Hard disk, 170 cylinders 8 heads, 10.625 MB
Drive :4 -- Hard disk, 170 cylinders 8 heads, 10.625 MB
Drive :5 -- Hard disk, 174 cylinders 8 heads, 10.875 MB
Drive :6 -- First floppy drive
Drive :7 -- Second floppy drive

Note that the entire 64MB accessible by LDOS is used. If you wish to reorganize the drives--for example to access more than two floppy drives--be aware that LDOS attempts to double the size of each granule on disks with more than 203 cylinders; this is highly discouraged and may cause LDOS and other software to fail. Furthermore it's advisable to have as many hard drives as acceptable due to the limit of 256 files per logical drive.

### Physical Organization

Although solid-state disks like MISE's CF cards don't contain physical read/write heads or rotating disk platters, the original terminology is retained. A disk drive has some number of data cylinders. A cylinder consists of one or more tracks, each corresponding to a drive head. A track/head holds some number of data sectors. In the case of LDOS, each head or track corresponds to one 8KB granule of 32 sectors.

Compact Flash memory cards each contain a controller that emulates an IDE (Integrated Drive Electronics) electromechanical disk drive, also known as PATA (Parallel AT Attachment). IDE, when addressed in C/H/S (Cylinder/Head/Sector) mode, appears very similar to the MFM controllers that LDOS can access using the RSHARD driver package from Misosys. Similar, but not identical; there are some important differences.

MFM uses an 8-bit data bus, but IDE is 16 bits wide. Although the MISE FPGA can access both bytes of the data bus, sector data is only written and read on the LSB, thus wasting half the space on the CF card. Furthermore, LDOS uses 32 sectors per track, whereas IDE can have up to--and CF cards usually have--63 (not 64) sectors per track. MISE uses only the first 32 sectors of eack track, again wasting nearly half (31/63) of the remaining disk space. This wasted disk space is not a problem with modern CF cards. The 2GB card supplied with MISE, for example, can hold up to six complete 64MB hard disk images even though so much of the space is unused.

MFM drives can have up to 8 heads, but IDE allows for 16--and CF cards usually emulate 16 heads. Rather than waste this space, MISE's hard disk adapter contains a register that allows heads 0-7 or 8-15 to be selected. The same register can be used to address cylinders beyond the first 1024.

## Hard disk adapter Adapter Registers

The following registers are implemented within the MISE FPGA:

Port 0xC0 (RO): Adapter Status

Bit 7 - Drive 0 hardware write protect (always 0)
Bit 6 - Drive 1 hardware write protect (always 0)
Bit 5 - Drive 2 hardware write protect (always 0)
Bit 4 - Drive 3 hardware write protect (always 0)
Bit 3 - Not an autobooting system (0 in MISE)
Bit 0 - IDE interrupt status

Port 0xC1 (R/W): Adapter Control

Bits 7:5 - No function
Bit 4 - Software reset (not implemented)
Bit 3 - Device enable (not implemented)
Bits 2:1 - No function
Bit 0 - IDE interrupt enable

Port 0xC2 (RO): Adapter Presence Detect

Bits 7:0 - Constant 0x01

Port 0xC3 (R/W): IDE Data MSB

Bits 7:0 - IDE data most significant byte

Port 0xC4 (R/W): Hard Disk Image Select

Bits 6:1 - Cylinder set select (IDE cylinder number MSB bits 7:2)
Bit 0 - Head set select (IDE head number bit 3)

### IDE Device Registers

The following registers are, except as noted, implemented within the IDE devices--usually CF cards, in the case of MISE:

Port 0xC6 (R/W): IDE Alternate Status (Read), Device Control (Write)
Port 0xC7 (RO): IDE Drive Address
Port 0xC8 (R/W): IDE Data LSB
Bits 7:0 - IDE data least significant byte

Port 0xC9 (R/W): IDE Error Status (read), Features (write)

Port 0xCA (R/W): IDE Sector Count

Bits 7:0 - IDE sector count

Port 0xCB (R/W): IDE Sector Number - 1

Bits 7:0 - IDE sector number - 1

When writing to port 0xCB, the MISE adds one to the value sent to the IDE device. When reading port 0xCB, MISE subtracts one from the value presented to the TRS-80. This provides maximum compatibility with the RSHARD MFM hard disk drivers from the RSHARD1 package, since MFM sector numbers start at 0 but IDE sector numbers start with 1.

Port 0xCC (R/W): IDE Cylinder Number LSB

Bits 7:0 = Cylinder number least significant byte

Port 0xCD (R/W): IDE Cylinder Number MSB

Bits 7:2 - See below Bits 1:0 - Cylinder number most significant byte bits 1:0

When writing to port 0xCD, bits 7:2 of the value sent to the IDE device are taken from port 0xC4 bits 6:1. This allows LDOS to access multiple hard disk images with IDE cylinder numbers beyond 1023. When reading port 0xCD, the cylinder number MSB returned by the selected IDE device is passed intact to the TRS-80.

Port 0xCE (R/W): IDE Drive/Head

Bits 7:5 - Should read constant 0x5 (IDE ECC/sector size parameters)
Bit 4 - Select secondary (slave) drive
Bit 3 - See below Bits 2:0 - Head number bits 2:0

When writing to port 0xCE, bit 3 of the value sent to the IDE device--bit 3 of the head number--is taken from port 0xC4 bit 0. This allows LDOS to access multiple hard disk images with head numbers in the 8-15 range. When reading port 0xCE, the IDE drive/head value returned by the selected IDE device is passed intact to the TRS-80.

Port 0xCF (R/W): IDE Status (Read), Command (Write)

**IDE Interrupt**

The IDE interrupt signal is routed to the TRS-80, allowing interrupt-driven hard drive access. Before enabling the interrupt, you must have an interrupt service routine in place. Since the network interrupt appears as bit 5 in the Interrupt Status Register at memory address 37E0H, the interrupt vector should be placed at address 4057H. To enable the interrupt, set bit 0 of I/O port 0xC1. The service routine must clear the interrupt source (normally by reading the IDE Status register at port 0xCF) before returning.

Alternatively, IDE interrupt status can be polled at port 0xC0 bit 0.

## Formatting hard disks

Use the IDEFORM utility to format hard disk partitions:

```
IDEFORM :d (MPW="<password>",NAME="<name>",SYSTEM,VERIFY,NOSTOP)
```

All command-line arguments and parameters are optional.

:d is the logical drive number to be formatted. It will be prompted for if omitted.

MPW specifies the drive's Master Password. It will be prompted for if omitted.

NAME or N specifies the drive's volume label. It will be prompted for if omitted.

SYSTEM or S causes the directory and boot sector to be initialized, but doesn't do any formatting.

VERIFY or V causes a long format verification process. This is not recommended for Compact Flash disks.

NOSTOP skips checking whether the disk already contains data and asking for its master password if so.

In order to format a disk without using the NOSTOP parameter, you must supply its Master Password. The Master Passwords for all Compact Flash partitions supplied with MISE are originally "PASSWORD".

## Reconfiguring the hard drives

As shipped, the MISE's hard and floppy drives are organized as described here. Some users may wish to reconfigure this arrangement, to have more or fewer hard drives, hard drives of different sizes, or a different number of floppy drives.

Before attempting any of these steps, make a backup of your CF card, or use a CF card reader/writer to dump an image to your PC. That way if something goes wrong you can recover your system.

When creating a hard drive, its size will be: cylinders * heads * 8KB.

### Removing and creating hard drives

In order to re-arrange your hard drives you will first need to remove one or more from the system. If you wish to remove **all** hard drives and start from scratch, skip this section.

Let's assume you want to remove drive :5. Start by disabling the drive:

```
SYSTEM (DRIVE=5,DISABLE)
```

Disabling the drive does not remove its assignment in LDOS. If you ENABLE the drive at this point, nothing will have been changed. To permanently remove the drive (not necessary if it's a floppy drive):

```
IDECLEAR :5
```

This will free the drive's disk space allocation for new drives to be created. The drive, if re-enabled, will become the lowest-numbered unused floppy.

Proceed to remove drives :4, :3, etc. in the same manner until you reach the desired arrangement. It's recommended that you remove the highest-numbered drive first and proceed downward, to leave the maximum amount of contiguous free space on the disk without any "holes". Do not remove drive :0, which would render the hard disk system unusable. Skip to the next section if you really want to start from scratch.

Next, if desired, create new hard drives to replace the ones you've removed, starting with the lowest-numbered:

```
SYSTEM (DRIVE=3,DRIVER="IDEROM/CMD")
```

This will generate several prompts. The "drive address <1-2>" is 1 for the Primary and 2 for the Secondary IDE device. "F/R" means fixed vs. removable. Use Fixed.

The partition's number of heads controls the number of heads assigned to the logical drive. It's recommended that you always use the maximum of 8 heads.

The number of cylinders controls how many will be assigned to the drive. The default value shows the maximum number that can be used. It is **strongly advised** that you use 203 or fewer cylinders because LDOS doubles the size of its file allocation units for any larger number. In addition to wasting disk space, this can cause problems with software including LDOS itself.

If the geometry of the new drive doesn't match the drive that previously occupied its disk space, or if the previous drive was corrupted, you will be advised that the new drive appears to be unformatted. **Important:** If you receive this message, you must *immediately* format the new drive, because certain LDOS commands will corrupt its Drive Code Table (DCT) entry.

If you don't see the message, it means that your new drive appears to match the physical geometry of an old drive. If this is not what you intended, it's advisable to re-format the drive.

See here for information on formatting hard drives.

Continue creating hard drives until you run out of drive numbers or reach the maximum of 1024 cylinders per physical IDE disk. Finish by executing SYSTEM (SYSGEN) to save your changes to disk.

**Starting from scratch**

If for some reason you really want to remove and reconfigure all the hard drives, read on. This is not something most MISE users will ever need to do. There are two main approaches.

**One way: Re-initializing the primary CF card from floppy diskettes**

1. Create MISE system floppies.
2. Insert and boot the MISE3 floppy by holding down <SHIFT> during reset. The following instructions assume that your first hard drive will be created as :4, allowing up to 4 floppy drives, :0 through :3, to be accessed during this procedure. Only one floppy drive is assumed, though.
3. Create the boot partition. Enter SYSTEM (DRIVE=4,DRIVER="IDEROM/CMD"). Answer the prompts as described in the previous section. Immediately format the drive after creating it.
4. Remove the MISE3 floppy and insert the MISE1 floppy. Then BACKUP :0 :4 (S,I).
5. SYSTEM (DRIVE=4,SWAP=0) to make the hard disk partition drive :0.
6. Remove the MISE1 floppy and insert the MISE2 floppy. Then BACKUP :4 :0 (I).
7. Remove the MISE2 floppy and insert the MISE3 floppy. Then BACKUP :4 :0 (S,I).
8. COPY HDSYS0/SYS.SYSTEM:4 SYS0/SYS.SYSTEM:0.
9. Remove the MISE3 floppy and insert the MISE4 floppy. Then BACKUP :4 :0 (I).
10. Create additional partitions as described in the previous section. Use the DEVICE command to determine drive assignments, and use SYSTEM (DRIVE=...,SWAP=...) to move hard and floppy drives where you want them.
11. SYSTEM (SYSGEN) to save your changes to disk.
12. Reboot to test the new configuration.

The basic system created by this procedure does not contain all the files that came with MISE. You can download additional files from this website, and copy them to MISE using FTP. Once you have all the files, run DO ACCESS to set the correct passwords and permissions on all the files. This script will also help you detect files that are missing.

**Another way: Initializing a second CF card**

This assumes you have the default logical drive number setup (hard drives :0 through :5, floppies :6 and :7).

1. Power down the TRS-80 and MISE.

2. Insert a second CF card into the Secondary IDE slot on the back of the IDE/CF adapter, after reading the warning here.

3. Powerup and boot from your Primary IDE disk.

4. Select a drive to disable, to temporarily free a drive number for your new partition. Choose either a hard drive or an unused drive number. Using hard drive :5, SYSTEM (DRIVE=5,DISABLE,DRIVER="IDEROM/CMD").

5. Answer the various prompts, entering "2" for the drive address. This selects the Secondary IDE device. The new drive doesn't have to be the same size as the existing one, but it should be large enough to hold the entire current contents of drive :0.

6. IDEFORM :5.

7. BACKUP :0 :5 (S,I). This backs up your LDOS system onto the new drive.

8. SYSTEM (DRIVE=5,SWAP=0). This makes your new drive :0.

9. At this point you can create more drives, or copy files to your new drive(s). However it is **IMPORTANT** that before you continue, you DISABLE and then IDECLEAR any remaining drives on the Primary CF card. If you do not, additional drives you create after moving the Secondary CF card to the Primary slot (see below) may have incorrect starting cylinder numbers. The following command will disable and clear all logical hard drives that are not on the same physical drive as :0:

   ```
   IDECLEAR -A
   ```

10. SYSTEM (SYSGEN). Note that these last two steps do not affect the configuration of the CF card in the Primary slot since drive :0 is now assigned to a logical drive on the Secondary card, and the new CONFIG/SYS file is written to drive :0.

11. Power down the TRS-80 and MISE. Remove the Primary IDE CF card. Move the Secondary IDE CF card into the Primary slot.

12. Power up and boot MISE. The ROM driver automatically detects that the boot drive was created in the secondary position, switches its DCT entry to the primary position, and disables drives that were in the primary position.

13. Create additional partitions as described in the previous section. Use the DEVICE command to determine drive assignments, and use SYSTEM (DRIVE=...,SWAP=...) to move hard and floppy drives where you want them.

14. SYSTEM (SYSGEN) to save your changes to disk.

15. Reboot to test the new configuration.

Or to simplify matters, just make a mirror-image backup to the second card:

```
IDEBACK
```

The secondary CF card need not even be formatted. See the backup utility page for more details.

### Backing up hard disks

MISE includes a utility for performing backups of an entire hard disk image:

```
IDEBACK (DEVICE=d,IMAGE=i)
```

The device number is 1 for the Primary IDE device (CF card) and 2 for the Secondary (the default). The image number ranges from 0 (the default) to the maximum allowed on the specific device, as determined by its number of cylinders and heads. Run IDEBOOT with no arguments to see the devices' configurations.

If you have logical drives that you do not wish to backup, disable them with SYSTEM (DRIVE=d,DISABLE) before running IDEBACK.

If you just want to do a simple backup of your primary CF card image to your secondary CF card, disregarding all this "alternate image" tomfoolery, just type IDEBACK with no arguments and hit <ENTER> in response to the dire warning.

Each dot printed while IDEBACK runs represents a copied cylinder.

IDEBACK performs a granule-by-granule backup of all logical drives on the current image on the primary IDE device to the specified image slot. If you have multiple images on the primary device that you wish to back up onto the secondary device, you must enter the IDEBACK command for each image after booting it. See the section on using multiple hard disk images for more information on booting alternate images.

Note that **MISE does not support hot-plugging of CF cards**. MISE and the TRS-80 must be powered down when inserting or removing CF cards in either device position.

## Using multiple hard disk images

Although LDOS hard disk images are limited to 64MB in size, MISE is capable of accessing alternate images on an IDE/CF disk. The 64MB restriction results from LDOS' inability to address more than 1024 cylinders or 8 heads on a disk. IDE, however, permits as many as 65536 cylinders and 16 heads.

To see how many images your CF card can hold, run the IDEBOOT utility without any arguments. This will display the active image number, a number of complete images and a number of partial images. Complete images consist of 1024 cylinders, regardless of how many are actually used by your logical disks. Partial images can use up to the number of cylinders shown. The minimum 2GB CF cards shipped with MISE can usually hold six complete images and two partial images.

MISE comes with only the default image, number 0, populated. Before you can boot an alternate image you must first populate it. There are two approaches.

### Creating an alternate hard disk image using floppies

You'll need MISE system floppies. Place the MISE3 bootable floppy in the first floppy drive. Holding down <SHIFT> while pressing <ENTER>, issue the command:

```
IDEBOOT <n>
```

where <n> is the number of the image you wish to populate. When the floppy finishes booting, proceed with the instructions given here for initializing a hard drive from scratch. Once you've created the bootable hard drive, you can populate additional drives on the image. Afterward, IDEBOOT <n> will boot the image directly. IDEBOOT 0 or resetting the TRS-80 will return to the default image.

### Copying an existing hard disk image to an alternate image

Use the IDEBACK utility to copy the entire current image to another image:

```
IDEBACK (DEVICE=d,IMAGE=i)
```

The device number is 1 for the Primary IDE device (CF card) and 2 for the Secondary. The image number ranges from 0 to the maximum allowed on the specific device, as determined by its number of cylinders and heads. For example, this command:

```
IDEBACK (DEVICE=1,IMAGE=1)
```

copies the current image on the primary device to the second image on the primary device, after which you can IDEBOOT 1 if the current image is bootable. And the command:

```
IDEBACK (DEVICE=2,IMAGE=0)
```

copies the current image to the first position on the secondary IDE device, making that CF card bootable if it is moved into the primary IDE slot. You cannot directly boot images on the secondary IDE device.

The two images must be the same size (that is, both complete or both identical-sized partial images). Be aware that doing this will overwrite the entire contents of the destination image.

The time required to perform an image copy varies with how much disk space is used. Allocation units (grans) that are not part of any file are not copied.

After populating an alternate image on a CF card (that is, IDEBACK to image number > 0) it is **strongly** recommended that you rename (change the volume label) of drive :0 of the new image. This is because alternate images are initially exact duplicates of image 0, and it can be difficult to tell which image is currently active. For example:

```
IDEBACK (D=1,I=1)
IDEBOOT 1
ATTRIB :0 (NAME="IMAGE1")
```

Now when you boot the alternate image, the new volume label "IMAGE1" will be displayed on the LDOS boot screen, with the FREE command, and in directory listings of drive :0.

## Using the VGA display

The MISE's VGA controller produces 800x600 @ 60Hz SVGA output. The screen is organized as a 768x576 VGA pixel display area which captures and shows the contents of the TRS-80 screen memory, with an improved character set that includes lower case and full descenders (g, p, etc.) and punctuation that extends above and below the alphanumeric character baseline. A border area of 12 VGA pixels top and bottom, and 16 pixels left and right, surrounds the TRS-80 display. The Model I problem of small, horizontal black lines appearing briefly on the monitor while accessing the screen memory has been eliminated.

Note that using the MISE with a wide-screen (16:9 aspect ratio) VGA monitor may cause the video to be stretched horizontally, appearing differently than it would on an original Model I CRT monitor. However, most such monitors have a mode for displaying a 4:3 image in its original format.

### Lowercase Letters

Support for lowercase letters is limited for Model I keyboard units without lowercase modification. This is because the 7bit display memory in the keyboard doesn't store bit 6 of character codes without such a mod. Instead, the generated bit 6 is set only when both bits 5 and 7 are 0. This has the effect of changing lowercase letters to digits and punctuation on the original Model I video monitor.

MISE stores all eight bits of characters written to video memory. Reads of video memory come from the Model I keyboard, and will therefore vary depending on whether a lowercase mod is installed. While MISE is capable of displaying lowercase characters on VGA whether or not a lowercase mod is installed, it's unlikely that any existing software uses this capability since without a mod lowercase letters written to screen memory are converted to digits and punctuation on the original Model I video monitor.

MISE is shipped with LDOS's lowercase driver disabled. In order to enable the lowercase driver, run the LOCASE utility. To disable the driver, run UPCASE. After running either utility, enter the command SYSTEM (SYSGEN) to store the changes to disk.

When the lowercase driver is enabled, caps lock is on by default. This is consistent with what you'll see on the Model I CRT monitor, and is meant to provide maximum compatibility with non-lowercase systems. You can type <SHIFT>0 (zero) to switch caps lock mode off or back on. LDOS also provides an advanced keyboard driver with additional features:

```
SET *KI KI
```

This driver allows entry of nearly all 256 possible binary character codes. See the LDOS manual for details. You can SYSTEM (SYSGEN) to save the driver setup to disk so that it will be reloaded on every boot, or alternately you can set up KI/DVR in your AUTOEXEC/JCL script. To disable this driver,

```
RESET *KI
```

and then SYSTEM (SYSGEN) again. Note (using the MEMORY command) that this does not remove the KI/DVR driver code from high memory; it just sets the current keyboard driver to the LDOS default. If you wish to remove the driver from high memory,

```
COPY CONFORIG/SYS.CCC:0 CONFIG/SYS:0
```

then reboot, reload any other high-memory drivers you had installed before, then SYSTEM (SYSGEN) again.

See the manuals page for the LDOS manual, which describes KI/DVR in detail.

**Changing Colors**

After powerup or reset, MISE's VGA controller defaults to white characters on a black background with a black border. The easiest way to change these colors is with the SETCOLOR/CMD utility:

```
SETCOLOR (FOREGROUND="color",BACKGROUND="color",BORDER="color")
```

See the table below for the color names, which can be abbreviated to three or more letters. Any color specification that is omitted is not changed.

FOREGROUND can be abbreviated as FORE or FG. BACKGROUND can be abbreviated as BACK or BG. BORDER can be abbreviated as BORD or BD.

The colors can also be changed by accessing two I/O ports:

Port 0xF1 (R/W):

Bit 7 - Use border color
Bits 6:4 - Background color
Bits 2:0 - Foreground color

Port 0xF2 (R/W):

Bits 2:0 - Border color

Setting a border color in port 0xF2 does not immediately change the screen's border color. Setting bit 7 in port 0xF1 causes the border color to take effect. This allows all three colors to be changed simultaneously. Writing port 0xF1 with bit 7 cleared sets the border color the same as the background color. This eliminates the need to write to port 0xF2 if you want the border and background colors to be the same. The following table shows the values to use for each color:

| **Color** | **Foreground** | **Background** | **Border** |
|---|---|---|---|
| **White** | 0 | 7 | 7 |
| **Yellow** | 1 | 6 | 6 |
| **Magenta** | 2 | 5 | 5 |
| **Red** | 3 | 4 | 4 |
| **Cyan** | 4 | 3 | 3 |
| **Green** | 5 | 2 | 2 |
| **Blue** | 6 | 1 | 1 |
| **Black** | 7 | 0 | 0 |

A BASIC program, COLORS/BAS is included which cycles through all the foreground and background color combinations.

**80-Column Mode**

*Note: 80-column display mode is only available in MISE units shipped with or updated to FPGA firmware dated June 8, 2015 or later. 48KB memory is required.*

The MISE's VGA output is capable of displaying characters as an 80x25 array instead of the native TRS-80's 64 columns by 16 lines.

To enter 80-column mode:

```
SET *DO C80
```

To exit 80-column mode:

```
DO C64
```

which runs the commands RESET *DO and OUT 0XF0 0. See the description of register 0XF0 below. Note that leaving 80column mode does not remove the driver from memory.

This mode was primarily developed to enhance the usefulness of the Telnet client. Many LDOS commands assume 64 columns, and their output is not readable in 80-column mode. Such commands include DIR, FREE and HELP. Some BASIC features like PRINT @ do not work in 80-column mode.

The 80-column driver, C80/DVR, installs itself as three drivers starting at a fixed memory location, &HE7F7, as can be seen using the MEMORY command:

- DSP is the 2,000 byte display memory, with a 9-byte driver descriptor prepended. You can alter the display output by POKEing to the &HE800-&HEFCF address range.
- C80 is the driver software.
- NIL is unused high memory space. It is intended to show how much free space was available for installing highmemory drivers before 80-column mode was entered.

The set of 127 display characters does not include six-cell graphics like native TRS-80. To see the entire character set, enter this BASIC program:

```
10 CLS
20 FOR X = 0 to 127 : POKE &HE800+X, X : NEXT
30 GOTO 30
```

Do not save your configuration using SYSTEM (SYGEN) while the 80-column driver is in memory. If you mistakenly do so, you will not be able to see the LDOS prompt after rebooting if 80-column mode was active at the time. Blindly enter DO C64 to get your prompt back, then restore the original driver configuration as described regarding KI/DVR above.

**VGA Test Mode and Interrupt**

The VGA controller contains a third register with some additional functions:

Port 0xF0 (R/W):

```
        Bit  7  - 80-column mode, see above
        Bit  5  - VGA test mode
        Bit  4  - Vertical blanking status (read only)
        Bit  1  - Interrupt status/clear
Bit  0  - Interrupt enable
```

**VGA Test Mode**

Setting bit 5 in port 0xF0 causes the VGA controller to enter a test mode in which foreground and background colors on the display are set to a specific test pattern. The utility program VGATEST/CMD enters this test mode with a repeating pattern of all 256 characters shown. Hit <BREAK> to exit VGATEST.

**Vertical Blanking Interrupt**

A vertical blanking interrupt and status bit allow applications to restrict their screen updates to times when the display isn't being drawn. The actual VGA vertical blanking duration is 739.2 microseconds. However, the interrupt and status bit are asserted sooner--when the bottom border region is reached--providing an additional 316.8 microseconds at both the bottom and the top of the screen during which CPU writes to screen memory will not be seen.

Before enabling the vertical blanking interrupt by setting bit 0, you must set up an interrupt service routine. Since the vertical blanking interrupt appears as bit 4 in the Interrupt Status Register at memory address 37E0H, the interrupt vector should be placed at address 4055H. The service routine must clear bit 1 in port 0xF0 to reset the interrupt.

Instead of using the interrupt, bit 4 of the register can be polled to determine when vertical blanking is in effect.

**High Resolution Display**

The VGA controller emulates the vintage PCG-80 and 80-Grafix devices for the Model I, which provided high-resolution graphics (384x192 pixels) by using programmable character sets. For both devices, the characters are programmed by entering special modes that capture writes to screen memory, 3C00H-3FFFH. Programming is done in banks of 64 characters at a time. The first 16 bytes of screen memory correspond to the first character, the second 16 bytes to the second character, and so on. Only the first 12 bytes of each character's screen memory are actually used, corresponding to the 12 vertical rows of TRS-80 pixels in a character. Bits 6:1 of each byte correspond to the horizontal pixels in a row, with bit 6 leftmost and bit 1 rightmost.

**Orcim Systems PCG-80 Mode**

Up to the entire 256 characters can be programmed, and displayed in 128-character banks.

Port 0xFE (WO)

Bit 7 - Display programmable characters 128-255 (if 0, display normal TRS-80 characters 128-255)
Bit 6 - Enter programming mode
Bit 5 - PCG operation
Bit 3 - Display programmable characters 0-127 (if 0, display normal TRS-80 characters 0-127)
Bits 1:0 - Programming bank select

Writes to this register have *no effect* on PCG-80 emulation unless bit 5 is set.

**Programma International 80-Grafix Mode**

Up to 64 characters can be programmed, and displayed in a single bank.

Port 0xFF (WO)

Bit 7 - Display programmable characters 128-191 (if 0, display normal TRS-80 characters 128-191)
Bit 6 - Enter programming mode
Bit 5 - 80-Grafix operation
Bit 3 - 32 characters per line "wide" display mode

Writes to bits 7:6 of this register have *no effect* on 80-Grafix emulation unless bit 5 is set. This I/O port is shared with the control register inside the Model I keyboard unit, which controls the cassette recorder and "wide" character display mode on the regular TRS-80 monitor.

## Using the real-time clock/calendar

MISE includes a real-time clock/calendar (RTCC) which is compatible with the vintage Alpha Products Newclock-80 device. As shipped, MISE uses a driver in its ROM extension which updates LDOS's software clock from the RTCC once per second as long as interrupts are enabled.

The simplest way to set the RTCC is automatically via the SNTP utility program. To instead manually set the date and time, follow these steps:

1. Run the STOPRTCC/CMD utility. This disables the ROM driver and enables LDOS's software clock.

2. Set the software calendar and clock using the DATE and TIME commands.

3. Run the SETRTCC/CMD utility. This sets the hardware clock from the software clock.

4. Run the USERTCC utility. This re-enables the ROM driver.

To disable the RTCC such that it will not be re-enabled by reboot, issue the command "SYSTEM (SYSGEN)" after running STOPRTCC. The software clock/calendar will still be set from the RTCC at reboot unless you physically remove the RTCC chip from its socket on the MISE circuit board.

Here's an ad for the original Newclock-80. See the manuals page for data sheets of both the emulated vintage RTCC chip and the modern chip actually used by MISE.

**I/O Ports**

Timekeeping registers within the RTCC chip are double-shadowed by the MISE FPGA. They are all read from the chip approximately 15 times per second. The values seen through TRS-80 I/O ports are all updated simultaneously when they are not being accessed.

Interlaced with the burst reads of the RTCC registers are time slots for writing to the RTCC chip if any of the shadowed date/time values have been modified by the TRS-80. Once a value is modified, it will not be updated until the new value has been written to the RTCC chip.

Below are listed the I/O ports and bit assignments used by the RTCC. Values read from unlisted bits should be disregarded.

Port 0xB0 (R/W*):
Bits 3:0 - Seconds ones; seconds range is 00-59

Port 0xB1 (R/W*):
Bits 2:0 - Seconds tens

Port 0xB2 (R/W):
Bits 3:0 - Minutes ones; minutes range is 00-59

Port 0xB3 (R/W):
Bits 2:0 - Minutes tens

Port 0xB4 (R/W):
Bits 3:0 - Hours ones; hours range is 00-23 or 01-12 (see below)

Port 0xB5 (R/W):
Bit 3 - 24-hour time format (not AM/PM)*
Bit 2 - Time is PM (not AM)
Bits 1:0 - Hours tens

Port 0xB6 (R/W):
Bits 2:0 - Day of week; 0=Sunday, 6=Saturday

Port 0xB7 (R/W):
Bits 3:0 - Day of month ones; day of month range is 01-31

Port 0xB8 (R/W):
Bit 2 - Leap year*
Bits 1:0 - Day of month tens

Port 0xB9 (R/W):
Bits 3:0 - Month ones; month range is 01-12

Port 0xBA (R/W):
Bit 0 - Month tens

Port 0xBB (R/W):
Bits 3:0 - Year ones; year range is 00-99

Port 0xBC (R/W):
Bits 3:0 - Year tens

***Notes:**

- Writing to either port 0xB0 or 0xB1 sets the RTCC's seconds value to zero by default. This is for compatibility with the Newclock-80; it allows a full minute to update the other fields before rollover can occur. Setting bit 1 in port 0xBF bypasses this compatibility mode and allows the seconds to be directly controlled. Both SETRTCC and SNTP bypass Newclock-80 compatibility.
- LDOS uses 24-hour time format.
- The leap year bit does not affect RTCC operation. Leap year status is generated by hardware. This is different from Newclock-80 operation.

## Using the joystick port

The MISE's joystick port is designed to work with a wide range of vintage Model I games. It is software-compatible with the vintage Alpha Products joystick. You'll need an Atari 2600-compatible joystick to use it. Plug the joystick into the 9pin male D-SUB connector on the MISE.

The majority of vintage games expect the joystick to operate in "5-bit" mode. That is, there are separate I/O port bits for each joystick switch: up, down, left, right, and the "fire" button. A few games, however (notably those produced by Adventure International) require "4-bit" mode, in which the joystick button does not have a separate bit; instead, pressing the button causes the "up" and "down" bits to be seen simultaneously. To enter 4-bit mode, set bit 1 of output port 0xBE. The command "OUT 0xBE 0x02" should do the trick. The Alpha products joystick used a physical switch for selecting the mode.

For maximum compatibility, the joystick switches can be read at three different I/O ports: 0x00, 0x13, and 0xF7. The bit positions are as follows (note that the bits read 0, not 1, if the corresponding switch/button is activated): **5-Bit Mode**

Bit 4 - 0 if "fire" button pressed.
Bit 3 - 0 if joystick handle pushed right
Bit 2 - 0 if joystick handle pushed left
Bit 1 - 0 if joystick handle pushed down
Bit 0 - 0 if joystick handle pushed up

**4-Bit Mode**

Bit 3 - 0 if joystick handle pushed right
Bit 2 - 0 if joystick handle pushed left
Bit 1 - 0 if joystick handle pushed down or "fire" button pressed
Bit 0 - 0 if joystick handle pushed up or "fire" button pressed

The state of unlisted bits should be disregarded.

When pushing the joystick handle diagonally, two switches will be activated simultaneously, so for example you'll see bits 1 and 2 active (low) when the handle is pushed to the lower left.

## 256-byte general-purpose EEPROM

MISE contains a 256-byte EEPROM that can be used to store data to be preserved when MISE and TRS-80 are powered off. Three demonstration programs with source code are included which can be used as examples when developing your own applications that use the EEPROM:

```
EEWRITE <address> <data byte>
```

```
EEREAD <address>
```

```
        EEDUMP
```

All data and address parameters may be specified as decimal or C-style hexadecimal numbers (e.g. 34 or 0x22).

EEWRITE writes the data byte to the specified EEPROM address.
EEREAD reads the byte from the specified address and displays it in C-style hexadecimal format.
EEDUMP reads and displays all 256 EEPROM data bytes as bare hexadecimal. Hit any key to exit EEDUMP.

The source code (e.g. EEWRITE/ASM) can be found on hard drive :1. To rebuild one of the utilities:

```
        ATTRIB EEWRITE/CMD.UTILITY:0 (PROT=FULL)
        MRAS EEWRITE
DO EEWRITE
```

The "DO" command runs a linker script that outputs the executable program file.

## Using the ROM-based double-density floppy disk adapter driver Installing drivers

MISE ships with floppy drives :6 and :7 set up for single-density operation. If you have a double-density adapter installed, you can load its drivers after booting MISE. For example LDOS's FDUBL utility installs high-memory drivers for the Tandy and Percom/Aerocomp adapters.

In its ROM extension MISE includes a driver for any of these three types of double-density adapters, allowing its use without consuming high memory. In order to enable this driver, use the DDEN/CMD utility. To disable the ROM driver execute SDEN/CMD, enter SYSTEM (SYSGEN), **and then reboot**.

*Note*: MISE units sold starting in 2018 ship with DDEN installed. In order to use FDBUL instead you will first have to enter SDEN, SYSTEM (SYSGEN), then reboot.

If you have a double-density adapter but don't wish to use the ROM-based driver, issue the command:

```
        FDUBL (PERCOM)
```

or

```
        FDUBL (TANDY)
```

to load a driver into high memory. Use (PERCOM) for an Aerocomp adapter since they are software-compatible. In either case, you must enter SYSTEM (SYSGEN) to save your driver changes to hard disk.

### Reverting drivers

If you ever wish to revert to MISE's original driver configuration, enter the command:

```
        COPY CONFORIG/SYS.CCC:0 CONFIG/SYS:0 then
```

reboot.

## MISE feature control registers

MISE contains a pair of registers whose bits control various features. The "disable" bits are included to enable the use of other peripherals that might conflict with MISE I/O port addresses.

Port 0xBE (R/W):

Bit 7 - Disable auto-boot
Bit 6 - Disable VGA registers (display stays active)
Bit 5 - Disable network module access
Bit 4 - Disable ROM extension

Bit 3 - Disable IDE/CF access
Bit 2 - Disable joystick port
Bit 1 - Joystick 4-bit mode

Port 0xBF (R/W):

Bit 2 - Expansion Interface detected (read only)
Bit 1 - Allow RTCC seconds to be set nonzero
Bit 0 - RTCC register write protect

## Utility programs

Several custom utilities are included with MISE. They don't require MISE to operate, but they have proven useful during its development.

DISKDATE :d <date>

DISKDATE/CMD allows changing a logical drive's date field originally set by FORMAT or IDEFORM. :d is the logical drive number, and <date> can be any 8-character string. Enclose the date string in double quotes if it contains spaces.

Note: If a hard drive's date string does not have slashes ('/') as its third and sixth characters, re-installing its driver using IDEROM will produce an error message about the drive appearing to be unformatted. However, the drive will install and function normally.

JV1COPY <source> <destination>

JV1COPY/CMD copies JV1-format images between floppy diskettes and files on the hard drives. Only single density, single sided images are supported. The source and destination arguments can be filenames or drive numbers.

OUT <port> <byte>

OUT/CMD sends a data byte to the specified I/O port. Either argument can be specified in decimal or C-style hexadecimal notation. For example, the following two commands are equivalent:

```
OUT 0xF1 0x90 OUT
241 144
```

REBOOT

REBOOT/CMD reboots the Model I, reducing wear-and-tear on the long-suffering reset button. This is similar to LDOS's BOOT command, but runs much more quickly and reliably.

SETSYS <filename> (SYSTEM=ON/OFF)

SETSYS/CMD sets or resets the System bit in a file's directory entry. The default is ON.

WHICH <filename>

WHICH/CMD searches for the specified file sequentially across all logical disks. It returns the full name, including drive number, of the matching file on the lowest-numbered drive.

If no filename extension is specified, /CMD is assumed. Thus WHICH determines which executable will be run when its name is entered on the LDOS command line.

## Installing FPGA updates

Occasional updates to the MISE FPGA configuration will become available on this website. Make sure that any programming file (with a .pof extension) you download is specifically compatible with your revision of MISE.

To install an FPGA update, you must obtain free programming software from Altera. You will have to create an account at http://www.altera.com. What you should download is the Quartus II Web Edition software. If you don't want to install the entire, very large application, you can just download the standalone programmer portion.

You will need a programming cable: either a ByteBlaster II that attaches to a PC parallel port, or a USB blaster. These (and clones) can be obtained inexpensively on eBay. The cable terminates in a 10-pin (2x5) receptacle with a raised bump on one side. Insert this connector into the 10-pin box header that hangs off the edge of the MISE, being careful to align the bump with the slot in the header. Some programming hardware lacks the polarizing bump. In this case, make sure that the cable's red "pin 1" stripe is toward the 9-pin D-SUB joystick connector.

If you download the standalone programming software to use with a USB Blaster, you'll also need a driver from Altera. Here's the current link:

https://www.altera.com/content/dam/altera-www/global/en_US/others/support/kdb/downloads/rd06212010_676/usb-blaster.zip

The MISE can be left in place, connected to your TRS-80, during programming, or you can disconnect it if you need to move it closer to the programming PC. Only the power supply connection is required.

Power up the MISE, but do not switch on the TRS-80 if still connected. Run the programming software, install your hardware, and set it for Active Serial programming. Add the supplied .pof file. Select the check-boxes for Program/Configure and Verify. Click the Start button. The FPGA will be programmed, and you should see a "success" message.

Power down the MISE and disconnect the programming cable. Reprogramming is complete.

**Note:** Some feedback from a MISE user on installing FPGA updates under Ubuntu Linux:

I just finally got around to installing this update too. (well some weeks ago)

2 difficulties I encountered along the way.

* Took me a while to figure out that ubuntu updates had broken the usb device permissions and that you need to install some udev rules to fix that, (or run the quartus app as root).

* I don't know where it's documented that the program to actually execute is ~/altera/13.1/qprogrammer/bin/quartus_qpgmw out of all the other equally likely looking scripts and binaries in several directories. That took me a while to find too, and I never did find it from any actual documentation like a readme or the altera web site. Googling turned up some forum post where someone mentioned it in passing.

But finally, fpga update installed fine and I ftp'd the new software to my drive :0 fine and everything is working fine afterwords.

**Note:** There is a bug in many versions of the Altera programmer software when used with a ByteBlaster II (parallel port) programmer, that you might have to work around. If the software informs you that your programming hardware is not compatible with Active Serial programming mode, you have hit the bug. First, select JTAG programming mode. Then load an .sof file, here are one for MISE 1.0 and one for MISE 1.1. Then hit "Start". The programming sequence will fail. Switch back to Active Serial mode, load the .pof file, select the Program/Configure and Verify boxes, and then hit Start. Your FPGA should now be reprogrammed successfully.

**Note:** If you ever for some reason replace the MISE's EP2C5 FPGA module--these are currently quite easy to find on eBay--you may have to remove four 0-ohm surface-mount resistors (shunts) labeled R1, R2, R9, and R10 on the module before plugging the module into the headers on the MISE. This is because the same module can be used with the EP2C5 FPGA's bigger brother, the EP2C8. The four shunts connect power and ground to the EP2C8 FPGA, but the corresponding pins on the EP2C5 are functional I/O pins used by MISE. For some reason all the module vendors I've used have populated these four shunts, unnecessarily and dangerously--if you power on the MISE with the shunts in place, your FPGA could be damaged.

**Warning:** There is at least one newer revision of the FPGA module that does not work with MISE without modification! You can tell you've got one of these if pin 17 is on the inside of the ring of pins; that is, the number 17 is printed closer to the FPGA chip than the number 18. The reversed pins are not a problem since neither is used by the MISE PCB. The real problem is that the power-on reset circuit was botched on this design. One side of R13 should be hooked to +3.3V and C14 should be hooked to ground, to provide a momentary active-low power-on reset signal to FPGA pin 73. Unfortunately the PCB has it backward. You can fix the problem by swapping R13 and C14 with a soldering iron and a bit of courage.

## MISE serial number

Each MISE unit has a unique, 16-byte serial number. It can be obtained using the GETSER/CMD utility:

```
GETSER

MISE Serial Number: XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX
```

This allows for about 3.4x10^38 MISE units to be produced. I'd better get back to soldering.

## MISE Hardware Errata MISE 1.0 PCB

1. The unit will not boot without the WIZ830MJ Ethernet module in place.

   **Explanation:** The Ethernet module drives an active-low interrupt signal to the FPGA, where it is combined with other system interrupts to generate the interrupt signal to the TRS-80 CPU. Without the Ethernet module in place, its interrupt signal is undriven. The FPGA pin used by that signal is not capable of having an internal pullup, so the TRS-80 may see a constant interrupt, preventing it from booting.

2. The pushbutton on the FPGA module should not be used.

> **Explanation:** The pushbutton (located on the underside of the module) was intended to provide a hardware reset to the MISE. Due to a board layout error its FPGA pin is allocated to an output signal. Pressing the pushbutton will short this signal to ground, potentially damaging the FPGA.

The reset button in the Model I keyboard unit provides a hardware reset to MISE.

**MISE 1.1 PCB**

No known issues.

## NOTES: