A C R A S H C O U R S E I N C O M P U T I N G
on the Z80 Membership Card
by Crash Kernigan *

So you got it built; now you want to see it **WORK**! This "Quick Start" section will get you going fast. Don't worry about breaking anything. If it crashes, just turn it off and back on again, and everything will be fine.

To use the Z80-MC CPU card alone (no Front Panel card or SIO card), see <http://sunrise-ev.com/photos/z80/Z80MC-2017.pdf> for more details. Otherwise, plug the Front Panel card on top of the Z80 Membership Card. **BE SURE TO GET THIS RIGHT**! Plugging them together backwards or off by a pin is an expensive mistake! To make sure, there is a broad white stripe next to J2 and P2 on both boards.

Next, it needs power. Any well-filtered source of +5v to +9v DC that can supply about 250 milliamps will do. Connect power as shown below; Positive POWER to "+", and negative GND to "—" on the 6-pin header on the Front Panel card.

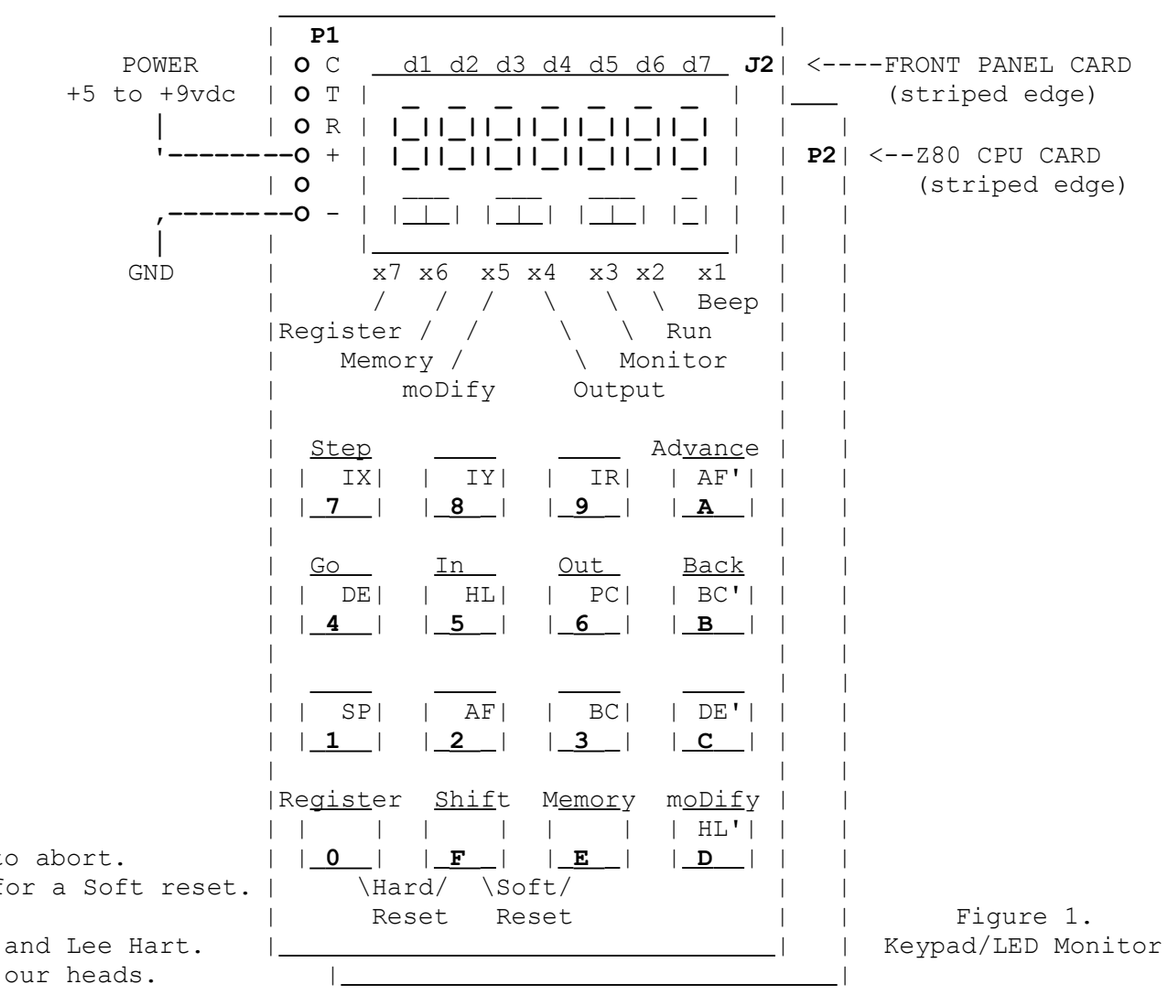


Figure 1. Keypad/LED Monitor

ZVM v1.5 Monitor -- Keypad/LED Command Summary

Key	Command	Key	Parameters
0	Register	1-D	select Register to view (1=SP, 2=AF, 3=BC, etc.)
4	Go		run Program at current PC
5	In	xx	read Input port xx
6	Out	xxyy	write yy to Output port xx
7	Step		Execute 1 instruction at PC
E	Examine	xxxx	Examine memory at xxxx
Once a Register, Memory, or I/O command starts:			
A	Advance		move Ahead one location
B	Back		move Back one location
D	moDify	xx	change current location
		or xxyy	2 or 4 digits based on destination size
F+E	Soft Reset		jump to Monitor (Warm boot)
F+0	Hard Reset		reset Z80 (Cold boot)

Notes: xx and yy are any hex digits 0-9, A-F.
Commands continuously show the last location accessed.
Incomplete commands time out in 3 seconds; so just wait 3 seconds to abort.
F is also a Shift key: Hold **F** down, then press **0** for a Hard, or **E** for a Soft reset.

--

* OK, true confession time. It's actually written by Josh Bensadon and Lee Hart. But this is how Crash would have done it, if he existed outside of our heads.

K E Y P A D / L E D M O N I T O R

On power-up, the display shows COLd 00. This means the system was COLD booted (reset). The 00 means it's the first time since power-up. This number will increase by one each time the Z80 is reset. Annunciator LED x3 is on because it is in *Monitor* mode.

There are two modes: **Monitor** and **Run**. **Monitor** lets you examine and modify RAM, CPU registers, and I/O ports. **Run** executes programs in memory.

There are nine keypad commands (plus two "oh shit!" Resets). Pressing a key starts that command. Most commands expect number(s) after the command. Once a command starts, enter numbers with **less than 3 seconds between keys**, or it times out and ends the command.

Examine memory	Press **E FFF1**	Display shows the address and its contents. Address FFF1 is the high byte of the "tick" counter, so the 23 will actually be counting up in real time.	FFF1 23
Advance to next	Press **A**	Advance to next address and show its contents.	FFF2 34
Backup to previous	Press **B**	Backup to previous address and show contents. Use **A** and **B** as many times as you like to move around.	FFF1 56
moDify RAM	Press **E**	First, **E**xamine the location you want to change.	
	Press **D**	LEDs x5 and x6 light to show you will mo**D**ify Memory.	
	Press 2 hex digits	Each digit pair is a byte that is written into RAM. The address then advances to the next location.	
	Press 2 more digits	To modify successive locations. Wait 3 seconds to end the command.	

RAM is located at 8000-FFFF. Obviously, you can't change ROM (Read Only Memory), which is at 0000-7FFF. Let's enter a simple program: JP SELF (Jump on Self... ouch!) The opcode for JP is C3. We'll enter it at address 8000, so the next two bytes are 00 and 80. The Z80 is a "little endian" CPU, which means it stores the lower byte in the lower memory address. (Don't change FA00-FFFF; they're used by the monitor.)

Press **E 8000**	to **E**xamine address **8000** (initially some random number nn).	8000 nn
Press **D**	mo**D**ify mode. LEDs x5 and x6 light.	
Press **C3**	Change 8000 to C3. Display advances to next.	
Press **00**	Change 8001 to 00. Display advances to next.	
Press **80**	Change 8002 to 80. Wait 3 seconds to end the command.	
Press **B B B**	Prove it worked by examining the contents of 8000-8002.	8000 C3
Press **A**	The easy way is to press **B** a few times to Backup to 8000,	8001 00
Press **A**	then **A** to Advance and see what's in 8001 and 8002 as well.	8002 80

To run a program, first we have to point the PC register (Program Counter) to it. Then we switch to *RUN* mode to run it!

Examine Register	Press **0 6**	to view Register PC (nnnn is the value of PC at the moment).	PC nnnn
Modify Register	Press **D 8000**	mo**D**ify it to 8000 (if it wasn't already there) Wait 3 seconds for the command to time out.	PC 8000

Run Mode Press **4** LED x2 turns on (Run mode) and x3 turns off.

It's now running your program! The PC is displayed; but not changing. That's because it's a 1-instruction program! Let's return to **Monitor** mode, and enter a bigger program so we can see it do something.

Monitor Mode Press and hold **F**, then press **E** X2 turns off, x3 turns on, and display shows SoFt 01

This switches back to **Monitor** mode. The display shows you got here with the keypad (a "soft reset"), and the number of soft resets since power-up. (If you're already in Monitor mode, pressing F-E shows F-E nn.) Here's the program we'll enter in the format used by an "assembler":

```
addr  data       assembler mnemonics   comments
8000  3C         HERE: INC   A         ; increment register A
8001  C2 00 80         JP    NZ,HERE   ; jump here (i.e. to 8000) if A is not 0
8004  03               INC   BC        ; increment register BC
8005  C3 00 80         JP    HERE      ; jump to "here" (so it repeats forever)
```

Enter this program:

Press **E 80 00** **E**xamine memory address 8000.

Press **D 3C C2 00 80 03 C3 00 80** mo**D**ify it to enter the program. Hint: Use **A** and **B** to check for mistakes.

Press **0 6** Check that Register PC is still at 8000. If not, set it to 8000 as described above.

Now, does it work? Let's see...

Single Step Press **7 7 7 7 ...** Go to **Run** mode, execute one instruction, then return to **Monitor** mode.
Each time you press Single-Step, one step of the program is executed. Hold it down to auto-repeat about twice per second. Since it is displaying the PC, you will see it step from 8000, 8001, 8000, 8001... Single-Step can trace programs in ROM as well as RAM. That's quite a useful trick!

Whatever you are viewing when you **Step** or **Run** will continue to be viewed after the step. This means you can watch any Memory location, Register, or Input port change as you **Step** or **Run** a program.

Press **0 2** To view Register AF (A and its Flags).

Press **7 7 7 7 ...** Every 2nd step, see A get incremented (as it executes the INC A instruction).

Press **4** Switch to *Run* mode. Now the AF display counts up too fast to read.

Press **0 3** View Register BC. It is counting up 256 times slower than A; but still very fast!

Examine Input Port Press **5 12** Read Input port 12. in12 78
(Note: There <u>is</u> no Input port 12, so this displays "air".)

Modify Output Port Press **6 12 34** Write 34 to Output Port 12. ou12 34
(Likewise, there <u>is</u> no port 12, so nothing visible happens).

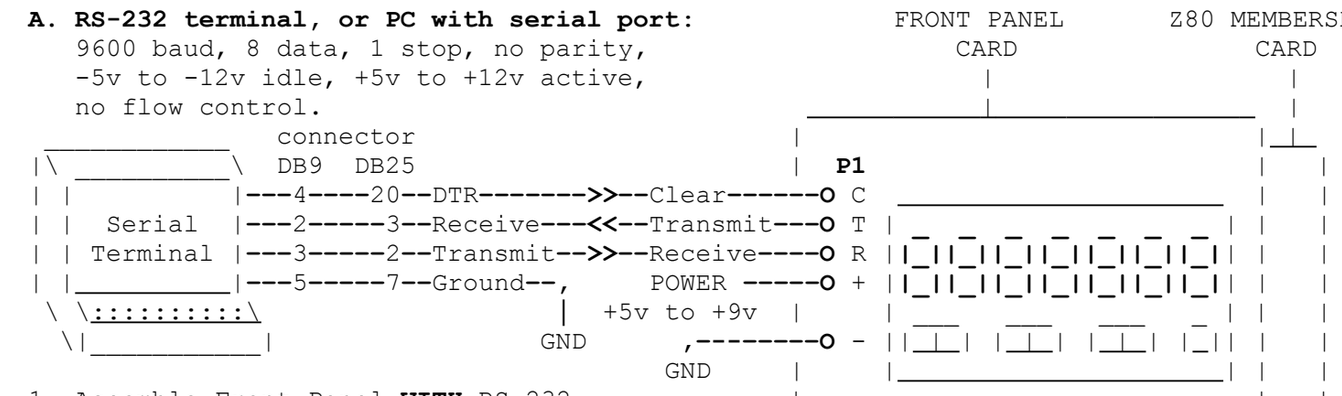
Hard Reset Press and hold **F**, then press **0** Reset Z80, and show # of resets. F-0 nn

This resets everything, just as if you had removed power, and powered it up again. It works even if the Z80 is not responding to interrupts or your program has crashed. (nn is just counting the number of times you hit F-0.

S E R I A L T E R M I N A L M O N I T O R

The ZMC ROM has a second monitor. It works with a serial terminal to give you a full-size keyboard and display. It works with just about any device that can send/receive serial data. You can use a real RS-232 data terminal; or a computer with a serial port; or a computer with a USB-to-serial adapter. If you're using a PC/Mac/Linux computer, you'll also need to run a terminal emulation program like HyperTerminal (Windows), MacTerminal (Mac), etc. There are two ways to hook it up (A and B):

A. RS-232 terminal, or PC with serial port:
9600 baud, 8 data, 1 stop, no parity,
-5v to -12v idle, +5v to +12v active,
no flow control.



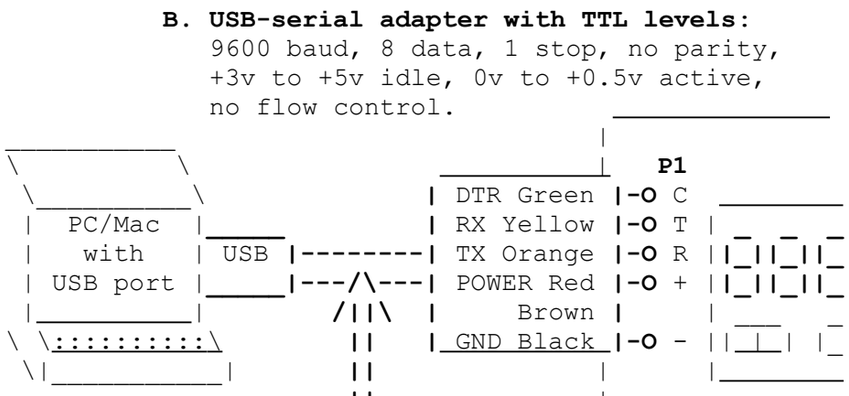
1. Assemble Front Panel **WITH** RS-232 option. **Install** Q1,Q4,D2,R3,C2 (page 16).

TERMINAL ACCESS COMMANDS

<**enter**> or **?**	display HELP screen
D xxxx yyyy	Dump memory from xxxx to yyyy
C xxxx yyyy	Continuous Dump (no pauses)
E xxxx	Edit memory starting at xxxx
M xxxx yy..zz	enter Many bytes yy thru zz into memory starting at xxxx
G (xxxx)	Go to PC (or optional address)
W	Watch execution (single step)
I xx	Input from port xx
O xx yy	Output yy to port xx
R rr (=xx)	Register (=xx change to xx)
L	Loop back test; echo what you type
T xx yy	Test RAM from page xx to yy
V	Version display
H	Load Intel HEX file
X U xxxx	XMODEM Upload to memory at xxxx
X D xxxx cccc	XMODEM Download cc blocks from xx
B	run BASIC (by Dave Dunfield)

extra commands with Z80-SIO card installed:
1 2 3 P Y U S (see Z80-SIO manual)

B. USB-serial adapter with TTL levels:
9600 baud, 8 data, 1 stop, no parity,
+3v to +5v idle, 0v to +0.5v active,
no flow control.



1. For example, the Sparkfun FTDI USB-serial cable plugs right in, and provides both power and serial data I/O. www.sparkfun.com/products/9718

2. Assemble your Front Panel with the **TTL Serial I/O** option (see page 16 in the Assembly Manual). Remove Q1 and Q4, and install jumper wires to short their base-to-collector.

3. Hint: I recommend removing pin 2 from P1, and plugging the hole for the brown wire in the USB-serial cable to act as a "key" so you can't plug it in backwards!

Figure 2.
Serial Monitor Summary

They say a picture is worth a thousand words (2k bytes); so here are screen dumps of the commands. The parts you type are **BOLD**. My comments are on the right.

Commands can be entered in upper or lower case. All numbers are typed and displayed in hexadecimal. Leading zeros are assumed (so typing address "55" is treated as "0055"). Only the last 4 digits are used; so mistakes can be corrected by simply typing the correct value after the wrong one (for example, mis-typing "220" is corrected to "0230" in the **Dump memory** command below).

```
Cold Start

Z80 MEMBERSHIP CARD MICRO-SD, V1.5beta July 23, 2017
HARDWARE: 01 FP
Main Menu >
```

Cold Start - The opening screen

Version number (and the output of the "Version" command)
This means the Z80 sees one extra card (the Front Panel)
ZMC prompt, when it's ready for your input

```
Main Menu >d 2200230  23f
M0230  0D 0A 43 6F 6C 64 20 53 74 61 72 74 0D 0A 00 0D ; ..Cold Start....
Main Menu >
```

Dump memory **D <StartAddr> <EndAddr>**
Dumps memory in ASCII format. Each line starts with M, then the address, up to 16 data bytes in hex, a semicolon, then the same bytes in ASCII (or a dot if it's not a printable character). Spaces are included for readability. **D** pauses and waits for a key after each page (so data won't scroll off the screen too fast to read). Press <Esc> to abort.

```
Main Menu >c 230  43f
M0230  0D 0A 43 6F 6C 64 20 53 74 61 72 74 0D 0A 00 0D ; ..Cold Start....
M0240  0A 53 6F 66 74 20 52 65 73 74 61 72 74 00 0D 0A ; .Soft Restart...
M0250  53 74 65 70 00 0D 0A (I aborted it at this point)
Soft Restart 02
AF=0045  BC=FF00  DE=0FFF  HL=02A4  AF'=0000  BC'=0000  DE'=0000  HL'=0000
IX=0000  IY=0000  IR=001F  PC=DAFF  SP=FF59
Main Menu >
```

Continuous dump memory **C <StartAddr> <EndAddr>**
C is the same as **D**, but does not pause. Use **C** to print or capture a Dump on tape or disk. The format is the same as the M command, so you can "play it back" to reload saved data back into memory without having to type the M command.

Press F, then 0 on the keypad to abort a long listing.
"Soft Restart" and the register contents will be displayed.

```
Main Menu >e 8000
8000 : 12 c3 C3
8001 : 34 00 00
8002 : 56 80 80
8003 : 78 <esc>
Main Menu >
```

Edit bytes in memory **E <StartAddr>**
The current address and its contents are displayed. Type the new value (C3 for example). The contents are then displayed again to see if it wrote, and the address is incremented. Note: C3 00 80 is the same as "JP 8000" used in the earlier example. Press <Enter> or the <Esc> key when done.

```
Main Menu >m8000 3c    ; inc a          Same little program as before.
m8001 c2 00 80         ; jp nz, 8000    Note that anything after a ";"
m8004 03               ; inc bc         is ignored, so you can even
m8005 c3 00 80         ; jp 8000        load an assembly listing.
```

Memory load **M <StartAddr> <1stByte> <nextByte>...**
Like **E**, but doesn't show memory contents before or after your entry. The **M** command automatically loads ASCII dumps created by the **C** or **D** command. The "Main Menu >" prompt is suppressed, but the monitor is ready for the next command after each line. Hint: If your terminal sends too fast, enable "pacing" (try 100 msec/line, 2 msec/char).

```
Main Menu >g 8000 PC=8000
```

Go execute program **G <Addr>**
Go to Run mode; your program is now running!

```
<Ctrl>-C 02
AF=0045  BC=0000  DE=D800  HL=2000  AF'=BFBD  BC'=BDFF  DE'=FFFF  HL'=FDBF
IX=AFFF  IY=FFFF  IR=0076  PC=8000  SP=FF5A
Main Menu >
```

control-C (Soft Reset) returns to Monitor mode, and displays all the registers so you can see what it was doing.

```
Main Menu >w
Step 01
AF=0045  BC=0000  DE=D800  HL=2000  AF'=BFBD  BC'=BDFF  DE'=FFFF  HL'=FDBF
IX=AFFF  IY=FFFF  IR=0076  PC=8000  SP=FF5A
Main Menu >
```

Watch execution **S**
execute ONE instruction at PC, shows number of steps so far, and displays registers so you can see what it did.

```
Main Menu >r
AF=FFFF  BC=BFBD  DE=FFFF  HL=FFFF  AF'=BFBD  BC'=BDFF  DE'=FFFF  HL'=FDBF
IX=AFFF  IY=FFFF  IR=001B  PC=8000  SP=FF5E
Main Menu >r b?
Main Menu >r BC
BC=BFBD
Main Menu >r BC=1234=1234
Main Menu >
```

Register examine or modify **R <Register>=<Value>**
R alone shows all registers.
R followed by a register name shows its contents.
Register names are CASE SENSITIVE,
so use "B", not "b".

Follow register name with "=value" to change it.

```
Main Menu >I FF 00
```

Input port read **I <port>**
Read Input port and show its contents (i.e. port FF is 00).
(Note: Since there is no port FF, this just displays "air".)

```
Main Menu >O FF 11
Main Menu >
```

Output port write **O <port> <byte>**
Output 11 to port FF. (This also does nothing, since there is no port FF.)

```
Main Menu >L Hello, world! <esc>
```

Loop back test **L**
Anything you type is simply echoed back to the screen and to the Front Panel LEDs. <Enter> starts a new line, and <Esc> ends the command. Use L to test your serial connection, or to see what ASCII looks like on a 7-segment LED display.

```
Main Menu >T 80 F9
TESTING RAM
RAM PAGE MARCH PASSED
RAM BYTE MARCH 1 PASSED
RAM BYTE MARCH 2 PASSED
RAM BIT MARCH PASSED
RAM SEQUENCE TEST PASSED
```

Test RAM **T <StartPage> <EndPage>**
Tests RAM from 8000-F9FF.
Don't test RAM pages FA00-FFFF; they're used by the monitor.
Patience, grasshopper: The entire battery of tests will take about 15 minutes to complete.

```
Main Menu >V
Z80 MEMBERSHIP CARD MICRO SD, v1.5 beta July 23, 2017
HARDWARE: 01 FP
Main Menu >
```

Version **V**
Shows the ZMC monitor ROM version number,
and what extra cards are plugged in.

HEX file transfers

Here's the easiest way to upload a program: Have your terminal send it as an Intel HEX file. These are produced by most assemblers. It includes the load addresses, the bytes to load, and checksums for error checking. It's just a simple ASCII text file, so it's easy to send. In HyperTerminal, click Transfer... Send Text File.

Intel Hex format starts each line with a colon. This colon is the Monitor's command to receive a hex file. You don't have to type this colon; just upload and watch. :-)

```
Main Menu >:188C840056D3F808F556D38EF63BD5F801F456F880AED38EFE3BD5F82D
:188C9C0001F556F80130D5F8FCA796B7E7F805BDF8ADF4F4ADF8F5A620
:188CB400E672AE93BC4DACDC4DACDC8E2656D4F800BC300BF801BCF82F
:188CCC00F5A6E672AE9BBFF0AFEF8EF3BE8EF23A1F15159C3A249E5FCE
:188CE400D49BBAD4455AE58AF4AA159A7C00BAD445A60A56302A45A686
:188CFC00065A302A2AD4F814AFF8005A1A2F8F3A45D40309010300015F
:188D140009020708090100030102030001020203030400FCFCFCFCFC1F
:188D2C00FCFC7222222247424272081525325 98EE0A0E004070217063D
:068D44003F0817040800BF
:00000001FF

HEX TRANSFER COMPLETE ERRORS=00
Main Menu >
```

Upload Intel HEX file **H**
Just send a hex file. It starts each line with a ":",
so you don't need to type anything.

This is normally the last line, which ends the command.
If it's missing, wait 10 seconds or press the <Esc> key.
no errors (good!)

XMODEM file transfers

HEX file transfers are easy, but slow. Here's a faster way to save and restore your work. XMODEM is a classic binary file transfer protocol that includes error checking. This version works with either Checksum or CRC error checking, and will auto-detect and auto-negotiate this.

```
Main Menu >x d 8000 fc

TRANSFER COMPLETE
```

XMODEM Download **X D <StartAddr> <#blocks>**
Send data **FROM** Z80 **TO** your terminal. Each block = 128 bytes.
This example sends a copy of all RAM from 8000 to FDFF.

This example sends all of RAM (8000 to FDFF) except for the top two pages (FExx and FFxx), which are used by the Monitor. It is pointless to save them, and it will crash the stack if you restore them (Cold Boot time...)! XMODEM sends 128-byte blocks. 8000-FDFF is 126 256-byte pages, which is 252 128-byte blocks, which is FC hex blocks.

To receive this download, your terminal needs a program that supports the XMODEM format. Luckily, just about every modem program made in the last 40 years has it. Here's an example using HyperTerm:

1. Type the command **x d 0 8000 fc** as shown above. The Z80MC is now ready to send data.

2. Within 2 minutes (before the command times out):

   - Click **Transfer... Receive File...**
   - Select **Xmodem** (not 1K Xmodem) from the drop-down list,
   - Select the filename to receive,
   - Then click **Receive**.

3. The transfer will complete automatically. Or to cancel it, type <Ctrl>-X.

```
Main Menu >x u 8000

TRANSFER COMPLETE
```

XMODEM upload **X U <StartAddr>**
Receive data **FROM** Terminal **TO** Z80.
This receives the above RAM image and loads it at 8000-7DFF.

This example receives the same blocks of data sent in the previous example, and loads them into RAM starting at 8000. You don't need to specify the number of blocks here, as it is set by the sending program (your terminal).

Here's an example for uploading this file using HyperTerm:

1. Type the command **x u 8000** as shown above. The Z80MC is ready to receive data.

2. Within 2 minutes (before the command times out):

   - Click **Transfer... Send File...**
   - Select **Xmodem** (not 1K Xmodem) from the drop-down list,
   - Select the filename to send,
   - Then click **Send**.

3. The transfer will complete automatically. To cancel it, type <Ctrl>-X.

Micro BASIC

Micro-BASIC is a fast integer BASIC interpreter by Dave Dunfield (included with his permission). BASIC commands must be CAPITAL letters (so press your CAPS LOCK key). The manual is at <http://www.classiccmp.org/dunfield/altair/d/basic.txt>.

```
Main Menu >B
MICRO BASIC COPYRIGHT 1983 BY DAVE DUNFIELD
?NO PROGRAM ERROR
READY
EXIT
```

Starts BASIC **B**
The sign-on message.
There is initially no program, so you get this message.
At the READY prompt, enter your BASIC commands or programs.
The EXIT command returns to the monitor.

If you use the CPU card "naked" (no Front Panel or Z80-SIO card), BASIC starts automatically at reset. You'll see a string of dots (….) as it looks for a BASIC program to run. To enter the Monitor instead, hit the **M** key.

--

There... this should get you going! Read the software manuals and source listings for more details on operation, and how to use the routines in the ZMC monitor for your own programs. Questions? Problems? Go to our user group at <https://groups.io/g/Z80MC>.

by Richie "Crash" Kernigan – last update 20 May 2025