Kaypro Emergency Monitor ROM
May 25, 2023

Table of Contents

Introduction

The purpose of the monitor is to help debugging a broken Kaypro. Replace the normal boot/BIOS ROM with a EPROM created from this monitor image. The Kaypro will now interact using the serial port (“Serial I/O” on early models, “Serial Data” on later models). No other hardware will be initialized, which means the display and keyboard will not work, nor will the disk. The display will usually be filled with random characters, or at least will not be cleared.

There is also a emergency memory test ROM available, for cases when the monitor ROM won’t even work.

The ROM Image

The monitor binary image may be downloaded here: http://sebhc.durgadas.com/kaypro/monitor.bin. Note that this is a 2K image, the same size as the 2716 EPROMs used for the Kaypro II and Kaypro IV (2/83, 4/83) models. It may also be put in larger EPROMs (2732 or 2764) for use with the Kaypro 10 and */84 models. When using larger EPROMs, it should not be necessary to replicate the image throughout the extra space (this needs to be confirmed). Note that the Kaypro IV/83 model (may also apply to some later Kaypro II) has a jumper-selectable ROM socket that takes either 2716 or 2732. Be certain to set the jumpers appropriate for whatever EPROM is being installed. The Kaypro 10/83 has a socket wired exclusively for the 2732. The */84 models should work with either 2732 or 2764 without changing any configuration.

Note that the AT28C16 EEPROM (FLASH) should work in place of the 2716 in Kaypro circuits. The AT28C64 should work in place of the 2764.

Using The Monitor

The monitor ROM does not depend on any hardware besides the SIO1 channel A serial port (and associated baud generator). However, it does provide a “heartbeat” indicator on the drive select LEDs on */84 (and 10) models – provided that the sysport is operating correctly. On */83 models, this heartbeat will be in the form of the first character on the display alternating between ‘A’ and ‘B’ – provided that the video is working. The heartbeat ceases after the first command is entered.

The monitor prompt is the colon character (‘:’). When the system is RESET the monitor will initialize the serial port and print a signon string and then prompt for input. Input lines may be edited using the Backspace key. Ctrl-C will abort the input and go back to the prompt. Upon pressing RETURN, the input line is then parsed by the monitor. Pressing RETURN at the prompt (empty input line) will just re-prompt.

Note that while the ROM is enabled by the sysport, Low memory addresses contain the ROM contents and are not writeable. The exact boundary between ROM and RAM depends on the model. Also, the Kaypro II and IV models map the video RAM into addresses 3000-3BFF. In order to access the RAM at low addresses, a special program must be used to transfer control into high memory and update the sysport to turn off ROM (and video RAM). On */83 models, the sysport is implemented in a Z80-PIO which must be initialized before it can be used.

The command ‘?’ will print a brief list of commands with some help text.

The following commands are implemented:

D *start end*

Dump memory in hex. Both *start* and *end* are required, and are interpreted as hexadecimal. Up to 16 bytes are displayed per line, with ASCII representation following the hexadecimal byte values.

S *start*

Substitute memory content interactively. The current address (initially *start*) and byte are displayed in hex and a single character of input is accepted. Pressing RETURN will go to the next address (+1) without changing the current byte. Pressing dash (‘-’) will go to the previous (-1) address without changing the current byte. Pressing period (‘.’) will end the command and return to the monitor prompt. Entering hex digits and RETURN will replace the current byte with the new value and advance to the next address (+1).

G *start*

Go (jump) to *start* and begin executing the code there. If the code executes any RST instruction, control will be returned to the monitor, where the message “*** RST” and the calling address (after the RST instruction) will be printed. No registers are saved and the code may not be resumed.

F *start end data*

Fill memory with the byte *data* repeated throughout the range.

M *start end dest*

Move a block of memory. If *start end* overlaps with *dest* then the results are undefined.

I *port* **[** *num* **]**

Input from *port* xxx and display the value in hex. If *num* is given, then the same port is read that number of times, and each value displayed. Both *port* and *num* are hexadecimal. There is no delay between inputs, other than the time it takes to print the value in hex.

O *port data* **[** ... **]**

Output *data* to *port*. If more than one data bytes are given, then those values are successively output to *port*. There is no delay between outputs, other than the time it takes to parse the next value.

N *hw*

iNitialize a hardware component. *hw* may be:
KB83 – Initialize the keyboard SIO channel on a */83 model.
KB84 – Initialize the keyboard SIO channel on a */84 or 10 model.
CRTC – Initialize the CRT controller chip on a */84 or 10 model.

T *hw*

Test a hardware component. *hw* may be:

KBD – Run a simple test for receiving codes from the keyboard. Waits for keys to be pressed on the keyboard and prints the hexadecimal code that is received. To end the test, type any character on the monitor console.

CRTC – Run a simple test for CRTC update status. Tries to detect the status change three times, and prints “Update” for each. The test may be aborted, if hung, by typing any character on the monitor console.

VRT – Run a simple test for CRT vertical retrace activity. Gathers up to 10 transitions of the VRT bit in the status register, then displays results. Output is similar to that for FLPY, but the 6845 status register is being watched instead. Normal VRT rate seems to be about every 20 milliseconds (50Hz).

CRTR – Run a pass/fail integrity test for CRTC register access. Since the only CRTC registers that can be both read and written are the cursor position hi/low registers, this test runs through all possible values 0000-3FFF and checks that the written value can be read back. If fail, prints the expected and actual values for the cursor position of the first failure (test stops).

FDRD – Do a floppy disk read of one sector into 8000h. User must first turn on motor, select a drive, select DD, select a side, and must have a formatted floppy diskette in the drive. See **Testing FDC** below.

FLPY – Run a test of the FDC system. User must first turn on motor and select a drive, and may need to have a floppy diskette in the drive. See **Testing FDC** below.

V

Display the ROM version. This just prints the ROM signon message again, which contains the version.

Testing The FDC

The floppy disk system may be tested in the following way. The FLPY test command will output a command to the WD1793 and then read the status register for a period of time, tracking any change to the status bits. On each change, the new status register value and iteration count are stored. At the end

of the test, the stored changes are printed. The test will end when the iteration counter rolls over (FFFF to 0000) or when the number of changes exceeds the limit (default 256), therefor this command should never hang. Optional parameters may be specified to give the WD1793 command to issue (default D0 – Force Interrupt), and the change count limit. Both are entered in hexadecimal. If the count limit is to be entered, the command must also be entered.

The user must select a drive, and typically turn on the motors. Depending on the desired test (command), it may also be necessary to have a floppy disk in the driver, and that disk may need to be formatted. This is entirely dependent on the desired test. The user is also responsible for turning off motors and deselecting the drive. Note that the Kaypro 10 uses the drive B select signal as the harddrive controller RESET.

The floppy control bits vary with the Kaypro model, and possibly even drive model (quad-density drives especially). Also note that the */83 models use a different system port (Z80-PIO) than the others. See also “Using SysPort on */83 Models” below. Here are the relevant bits (expressed as hexadecimal mask) for the major families:

*/83 (PIO 1CH)	Drive A select	01	0=select
	Drive B select	02	0=select
	Side select	04	0=side-1
	DD enable	20	0=DD
	Motor control	40	0=ON
Kaypro 10 (port 14H)	Drive A select	01	0=select
	WD1002 reset	02	1=reset
	Side select	04	0=side-1
	Motor control	10	1=ON
	DD enable	20	0=DD
*/84 (port 14H)	Drive A select	01	0=select
	Drive B select	02	0=select
	Side select	04	0=side-1
	Motor control	10	1=ON
	DD enable	20	0=DD

Note that the motor control bit is the major difference, with not only the bit location but also the polarity of the bit needed for “Motor ON”.

The general practice with SysPort is to read the current value, modify it, then write the new value. In this environment, there should be no hidden changes to the port and so it is reasonable to just know what the right value is without reading the port.

On a 4MHz Z80 Kaypro, each iteration count represents about 12 microseconds. On a 2.5MHz Z80 Kaypro, one iteration is about 19.2 microseconds.

The full syntax of the floppy test command is:

T FLPY [*cmd* [*limit*]]

Where *cmd* defaults to D0 (Force Interrupt) and *limit* defaults to 00 which means 256. See documentation for the WD1793 for available commands, their dependencies, and their side effects. Remember that STEP, SEEK, and RESTORE commands will require valid step rate bits for the drives being used. Also remember that commands which result in DRQ will not work correctly here.

T FDRD [*sector*]

Where *sector* defaults to 0 (Kaypro formatting numbers sectors 0-9 on side 0, and 10-19 on side 1). Note that *sector* must be entered in hexadecimal. This command prints the contents of the status register when finished.

Examples

In the examples, characters typed by the user are underlined. *(CR)* means pressing the RETURN key.

Getting Help

This shows the signon message and how to display the help menu.

```
Kaypro Monitor v2.1

: ?(CR)
D <start> <end> - display memory in HEX
S <start> - set/view memory
    (CR) = skip fwd, '-' = skip bkwd, '.' = done
G <start> - go to address
F <start> <end> <data> - fill memory
M <start> <end> <dest> - Move data
I <port> [num] - Input from port
O <port> <value> [...] - Output to port
N <hw> - iNitialize hardware (KB83, KB84, CRTC)
T <hw> - Test hardware
  (KBD, CRTC, VRT, CRTR, FDRD, FLPY)
V - Show ROM version
^C aborts command entry
:
```

Executing Code and Breakpoints

Set a RST 7 instruction into memory and jump to it.

```
: S8000(CR)
8000 00 FF
8001 00 .
: D8000 800F(CR)
8000 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
: G8000(CR)
GO 8000 ?Y
*** RST 8001

:
```

*Using SysPort on */83 Models*

Initializes the PIO channel A for use as the sysport, then alternately selects drive B and drive A.

```
: O 1C 81(CR)       (make certain BANK stays on)

: O 1D CF 8(CR)     (init and activate PIO ch A)

: O 1C 82(CR)       (turn on drive B LED)

: O 1C 81(CR)       (turn on drive A LED)

: O 1C 80(CR)       (turn off both LEDs)
```

CRTC 6545 Testing

The CRT Controller chip, 6545 or sometimes 6845E, exists only on models */84 and 10. It powers-on and RESETs to a state where the video is disabled. The following commands may be used to initialize the core video functionality, and test for the “update” status (which is used to read/write the video RAM). The first command initializes the CRTC and enables video. After this command, the display should light up and be filled with random/garbage characters (whatever is in the video RAM after power-up). The second command runs a quick test to see if the video update status bit is working. In the case that it is not working, the command will hang. Pressing any key should abort and get back to the monitor prompt.

```
: N CRTC(CR)        (display should light up)
: T CRTC(CR)
Wait... Update Update Update
:
```

An additional test is to check that vertical retrace is happening, and at about 50Hz. With the CRT initialized:

```
: T VRT(CR)
80 001C
A0 064B
80 06AF
A0 0CDE
80 0D42
A0 1371
80 13D5
A0 1A03
80 1A68
A0 2096

:
```

The VRT bit mask is 20 (hex), and we see that we get a pulse about every 0693H iterations (1683 decimal). At 12 uS per iteration, this computes to approximately 20 mS which is 50Hz.

Keyboard Testing

The keyboard uses SIO channel B on the same Z80-SIO chip as this monitor port is using, so that part is likely to also work. However, the keyboard itself may be tested by initializing the SIO port and

running a test that will show the received codes for keys pressed on the keyboard. The keyboard initialization is slightly different for */83 models, so use the appropriate command syntax for the Kaypro model being tested. The Kaypro keyboard contains the “beep” functionality, which is activated by sending a Ctrl-D character out to the keyboard. This can be tested using the Output command as shown in the example.

```
: N KB84(CR)        (initialize keyboard port on */84 and 10 models)
: T KBD(CR)         (start keyboard test, press keys on keyboard)
Wait... 61 73 64 66 77 65 65 72 F3 F4 (CR)Abort
: O 05 04(CR)
                    (keyboard beeps)
:
```

Floppy Testing (Index hole)

Here is a simple operational test for the floppy subsystem. This examples first selects drive A and turns on the motor (also, the user is expected to insert a floppy disk). Then the test is run to track the index pulse coming from the spinning floppy diskette. This test is for running on models */84 and 10.

```
: I 14(CR)          (get current value of SysPort)
Input 14 = AF

: O 14 BE(CR)       (select A and motor on)

: T FLPY(CR)        (run test and dump changes)
04 0000
06 3901
04 39E6
06 7A17
04 7AFC
06 BB2C
04 BC11
06 FC41
04 FD26

: O 14 AF(CR)       (motor and select off)

:
```

In this example, you can see that the TRK00 bit is on and the INDEX bit is cycling (indicating that the drive is spinning the floppy and sending index pulses to the WD1793). Also, a crude calculation of the motor speed can be made. Subtracting the first two adjacent “06” statuses gives 4116H or 16662 iterations. At 12 uS per iteration this computes to 199.944 milliseconds, where the perfect result would be 200 mS to match the 5 RPM speed of the standard floppy motor.

More sophisticated tests may be done by setting up WD1793 registers and using a more complex FDC command. Note that performing actual disk I/O is not practical with this interface, but SEEK with verify commands will read sector headers to verify the track, and so do a modicum of I/O internally.

Floppy Testing (Stepping)

This test checks the ability of the drive to step in/out, and the FDC management of that. There is no need to have a floppy in the drive for this test, but the head protector should be removed and the drive

door left open. SEEK and RESTORE commands are set to 30mS per step, and each should span 5 tracks, so total time should be about 150mS. When issuing the SEEK and RESTORE commands, listen for the head stepping noise.

```
: I 14(CR)
Input 14 = EF      (current sysport)

: O 14 FE(CR)      (motor on, select A)

: O 10 03(CR)      (RESTORE - make sure drive is at track 0)

: O 13 05(CR)      (set destination track to 5)

: T FLPY 13(CR)    (send SEEK command)
45 0002            (24uS: command starts, BUSY on, TRK00 on)
41 00A4            (1.9mS: TRK00 goes off)
40 30D7            (150mS: command done)

: I 11(CR)
Input 11 = 05      (check current track = 5)

: T FLPY 03(CR)    (send RESTORE command)
41 0002            (24uS: command starts, BUSY on)
45 28C4            (125mS: TRK00 on)
44 30FD            (150mS: command done)

: O 14 EF(CR)      (motor off, drive select off)

:
```

Floppy Testing (Reading media)

For this test, a formatted floppy is required to be in the drive. The SysPort value is chosen for the density and side, as well as drive and motor-on. This examples reads the first directory sector (sector 0A on side 1) of a DD floppy:

```
: I 14(CR)
Input 14 = EF

: O 14 DA(CR)      (motor on, DDEN, side 1, drive A)

: T FDRD 0A(CR)    (read sector 10)
00                 (no errors)
: D 8000 81FF(CR)
8000 00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 3A   .PIP     COM...:
8010 02 03 04 05 00 00 00 00 00 00 00 00 00 00 00 00   ................
8020 00 45 44 49 54 20 20 20 20 43 4F 4D 00 00 00 80   .EDIT    COM....
8030 06 07 08 09 0A 0B 0C 0D 00 00 00 00 00 00 00 00   ................
8040 00 53 54 41 54 20 20 20 20 43 4F 4D 00 00 00 29   .STAT    COM...)
8050 0E 0F 10 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
8060 00 43 20 20 20 20 20 20 20 43 4F 4D 01 00 00 80   .C       COM....
8070 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20   ...............
8080 00 43 20 20 20 20 20 20 20 43 4F 4D 02 00 00 34   .C       COM...4
8090 21 22 23 24 00 00 00 00 00 00 00 00 00 00 00 00   !"#$............
80A0 00 52 4D 41 43 20 20 20 20 43 4F 4D 00 00 00 6A   .RMAC    COM...j
```

```
80B0 25 26 27 28 29 2A 2B 00 00 00 00 00 00 00 00 00   %&'()*+.........
80C0 00 4C 49 4E 4B 20 20 20 20 43 4F 4D 00 00 00 7B   .LINK    COM...{
80D0 2C 2D 2E 2F 30 31 32 33 00 00 00 00 00 00 00 00   ,-./0123........
80E0 00 4C 49 42 20 20 20 20 20 43 4F 4D 00 00 00 38   .LIB     COM...8
80F0 34 35 36 37 00 00 00 00 00 00 00 00 00 00 00 00   4567............
8100 00 2F 20 20 20 20 20 20 20 43 4F 4D 00 00 00 0A   ./       COM....
8110 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   8...............
8120 00 4D 41 4B 45 4C 49 42 20 53 55 42 00 00 00 04   .MAKELIB SUB....
8130 39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   9...............
8140 00 4A 55 4E 4B 20 20 20 20 53 55 42 00 00 00 01   .JUNK    SUB....
8150 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   :...............
8160 00 42 45 4C 4C 20 20 20 20 41 53 4D 00 00 00 01   .BELL    ASM....
8170 3B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ;...............
8180 00 43 4F 4D 50 4C 49 42 20 53 55 42 00 00 00 0D   .COMPLIB SUB....
8190 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ?...............
81A0 00 49 4E 44 45 58 20 20 20 52 45 4C 00 00 00 01   .INDEX   REL....
81B0 3C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   <...............
81C0 00 43 4F 4D 4D 41 4E 44 20 52 45 4C 00 00 00 09   .COMMAND REL....
81D0 45 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   E...............
81E0 00 45 58 45 43 20 20 20 20 52 45 4C 00 00 00 04   .EXEC    REL....
81F0 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   F...............
: O 14 EF(CR)      (motor, etc, off)

:
```

Memory Test ROM

The memory test binary image may be downloaded here:

http://sebhc.durgadas.com/kaypro/memtest.bin

This image does not depend on RAM to run (uses no stack or RAM variables) but it does use the serial port for progress (output) only. It does a memory test in the range 8000H-FFFFH and will alternate drive A and B LEDs (*/84 and 10 models) or cycle through ‘0’ to ‘9’ in the first character of the display (*/83 models). If a memory error is detected, it will flash the drive A LED or alternate ‘E’ and ‘R’ in the first character of the display (depending on the Kaypro model).

Serial port output begins with a signon message showing the test version. During the test, it will output the current “seed” value being run (00-99 BCD). If an error is encountered, it will print the error address, expected value, and actual value (and “Error”).