**Document No: CSI426KayproII-MAN1.01**
**Date:** *December 4, 1998*

# **Kaypro II Emulator** Instruction Manual

- **Brandon Buist**
- **Joe Diethorn**
- **Jim Gilmer**
- **Shannon Steinmetz**
- **Hung To**

# Table of Contents

## Introduction

This manual describes the operation of the Kaypro II emulator. The emulator is a Java based emulation of a Kaypro II computer system. Java is a language that allows remote programs to be executed on a client computer via the World Wide Web. The Kaypro II emulator resides on a remote host machine. The user may run the emulation on a compatible browser from their own computer system.

**This document assumes the user is familiar with the Kaypro II and the CP/M operating system.*

**The implementation features a number of convenient and useful features, including:**

- Kaypro II emulation at a hardware level
- Printer port simulation
- Hardware style breakpoints
- Opcode level debugging
- CPU register display
- Memory dump utilities
- Dual virtual diskette drives with CP/M pre-loaded
- Real and fast mode video options

In addition to the above, the system is coded in such a way that it is expandable. This expandability requires changes to the source code. The modular nature of the implementation allows easy modification. Possibilities include:

- Easily upgraded system components
- Expanded functionality within the base component group
- Pedagogical activities

### *Hardware Level Emulation*

Figure 1 depicts the emulation model for the Kaypro II. A number of options were available. The emulator could have been written at the User Function level, the CBIOS level, The Resident ROM level, or the Hardware level. Each has specific advantages and disadvantages.

For example, if the system were emulated at the CBIOS level, and a user program were to write directly to a port, the emulation could fail.  This is because the specific port function would not be supported.  Typically, however, CP/M programs were well-behaved.  That is, most utilized CBIOS vectors for access to low-level functions.  Most programs did not make these “end-runs” to the hardware.  The Kaypro II emulator represented here affords the user an extra measure of protection for the occasional renegade program.

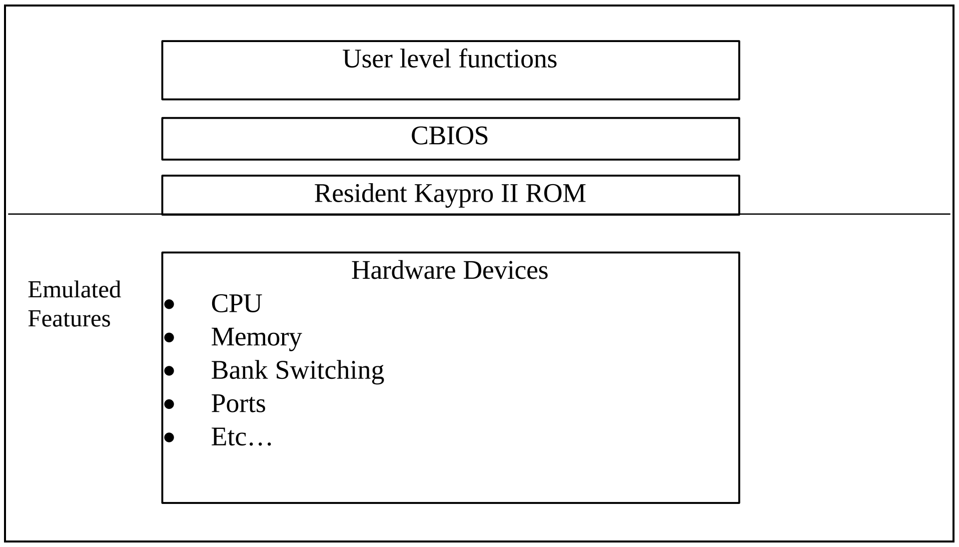


**Figure 1, Emulation Levels**

### *Printer Port Simulation*

The Kaypro II emulator includes printer port simulation.  This allows the user to view output to the printer port.  The printer output is simple text.  This assumes that the printer port is being used as such.  No provision is made for external devices connected to the printer port.

The printer parallel/printer port is described in detail later.

### *Hardware Style Breakpoints, Opcode Level Debugging, CPU Register Display, Memory Dump Utilities*

The Kaypro II emulation includes a basic set of debugging options.  These options allow the user to single step code, view memory and set break points.

***What is a "hardware style breakpoint?"***

Breakpoints can be set on memory location, or opcode.   The breakpoints are "hardware style" because the emulation takes place at the hardware level.  The emulator itself monitors memory and opcodes.  When a specific opcode or memory location is effected, the emulator takes care of the breakpoint logic.

In contrast, some systems implement breakpoints at a software level.  One method is to replace existing opcodes with illegal or alternative opcodes.

<u>**For example**</u>

If a breakpoint were to be placed at location 0x100…

   The opcode at memory location 0x100 may be changed to a subroutine jump to emulation code
   This code would handle the breakpoint
   Once handled, the previous opcode would be executed, and execution continued via a return

The problem with this method is that it introduces a number of complexities.  The most obvious is the diddling with the target code.  This can effect timing, or introduce unforeseen side effects.

## *Dual Virtual Diskette Drives With CP/M Pre-Loaded*

The emulator includes two virtual floppy disk drives.  The virtual diskettes have CP/M preloaded, and are accurate to the track and sector level.  Since the drives are emulated at a hardware level, even direct manipulation of the drive chip is possible.  Seek times are not accurate, but could be implemented.

## *Real And Fast Mode Video Options*

The Kaypro II emulation allows the user to select real or fast mode video display.  Real mode duplicates the actual Kaypro II character ROM.  Because this is graphics intensive, a fast mode is also supplied.  Fast mode uses standard system fonts.  The result is faster screen refresh.   It should be noted that the screen process runs as a separate task.  This reduces the impact of the graphics display on the executing code.  The greatest effect is the refresh of the Kaypro screen.

<u>**For example**</u>

If Real mode is selected on a system with a relatively slow processor
The code may execute quickly, but the screen may take a while to update.

## System Requirements

1. The emulator is viewed via the World Wide Web.  This means that the user system must have internet access.
2. The browser must be Java enabled.  It must support Java JDK version 1.1.6 or greater.
3. We recommend Internet Explorer 4.0[1] or greater when running on a Windows platform.
4. The CPU should be a Pentium 90 or greater.
5. Screen resolution should be 640 x 480 or greater. 1024 x 768 is recommended.
6. Screen color depth should be at least 256 colors.
7. Hard drive should be sufficient to handle temporary files  5 megabytes of free space is recommended.
8. 56KB modem, or T1 connection to internet

[1] The Java virtual machine within the browser is most important.  Known compatible virtual machines include Microsoft's VM for Java (tm), 4.0 Release 4.79.0.2424.  Microsoft's 5.0 virtual machine is known to have problems.

## Setting The System Up Locally

Typically the emulation is executed via the web (see next section). However, a local system may be set up as well. The following HTML code will allow the Kaypro II emulator to operate locally.

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<TITLE>
HTML Test Page
</TITLE>
</HEAD>
<BODY>

Kaypro II Emulator, Sample Applet HTML<BR>

<APPLET
  CODEBASE = "."
  ARCHIVE  = "kaypro.jar"
  CODE     = "kaypro.kaypro.class"
  NAME     = "Kaypro II emulation, proof of concept"
  WIDTH    = 656
  HEIGHT   = 600
  HSPACE   = 0
  VSPACE   = 0
  ALIGN    = middle
>
<param name="mode" value="1">
<param name="URL" value="http://clem.mscd.edu/~diethorn/logo/">
<param name="Dir" value="c:\kaypro\">
</APPLET>
</BODY>
</HTML>
```

The Kaypro applet has a number of parameters associated with it. They are…

| **Parameter** | **Values** | **Description** |
|---|---|---|
| Mode | 0= use URL parameter to load from URL<br><br>1=use Dir parameter to load from local drive | The emulator my be operated from the internet, or a local hard drive. The mode parameter specifies which |
| URL | A fully qualified URL (including end slash) | The URL address containing the Kaypro image files. Ignored if mode is set to 1. |
| Dir | A fully qualified directory (including end back-slash) | The local directory containing the Kaypro image files. Ignored if mode is set to 0. |

The following files must be located in the Kaypro directory.  The Kaypro directory should be located right off of the root.

The \kaypro directory on a local installation must include the following files:

Kp1.gif, kp2.gif, kp3.gif, kp4.gif, kp5.gif, kp6.gif, kp7.gif, kp8.gif, kaylogo.gif, Kaypro.jar, Kaypro.html

These files supply the graphics image files (.gif), Java code (.jar) and HTML files needed to operate the Kaypro emulator.  The HTML example shows a setup for a locally operated applet.

If the code is executed off of the Internet, the local machine needn't have any files resident.  Only the host server needs code and images loaded.

## Starting The Emulator

The emulation may be viewed via the Internet, and the World Wide Web.

Once the Internet connection is established, type the following http address into the browsers address box[2].

http://www.yoy.org/kaypro/html/kaypro.htm

The Kaypro II main screen appears. Although the web page should appear immediately[3], the actual Java program must load as well. This may take a few minutes. Typically, web browsers have an animation in one corner of the browser. When this animation stops, the page should be loaded[4].

*Load speed is dependent on the connection. A 28.8K Modem should take approximately 2 minutes to load the software.

Once connection is established, and the Kaypro II screen appears, the user is free to begin typing into the emulation. The user may treat the emulator just as they would the actual Kaypro II computer system.

If no screen appears, or the browser indicates an error, make sure that the correct address has been typed into the proper area of the browser. Make sure that you are actually connected to the Internet. One way to test the connection is to go to a known web page. If this fails, the chances are that you have not established a connection. Keep in mind that the Internet is a volatile place.

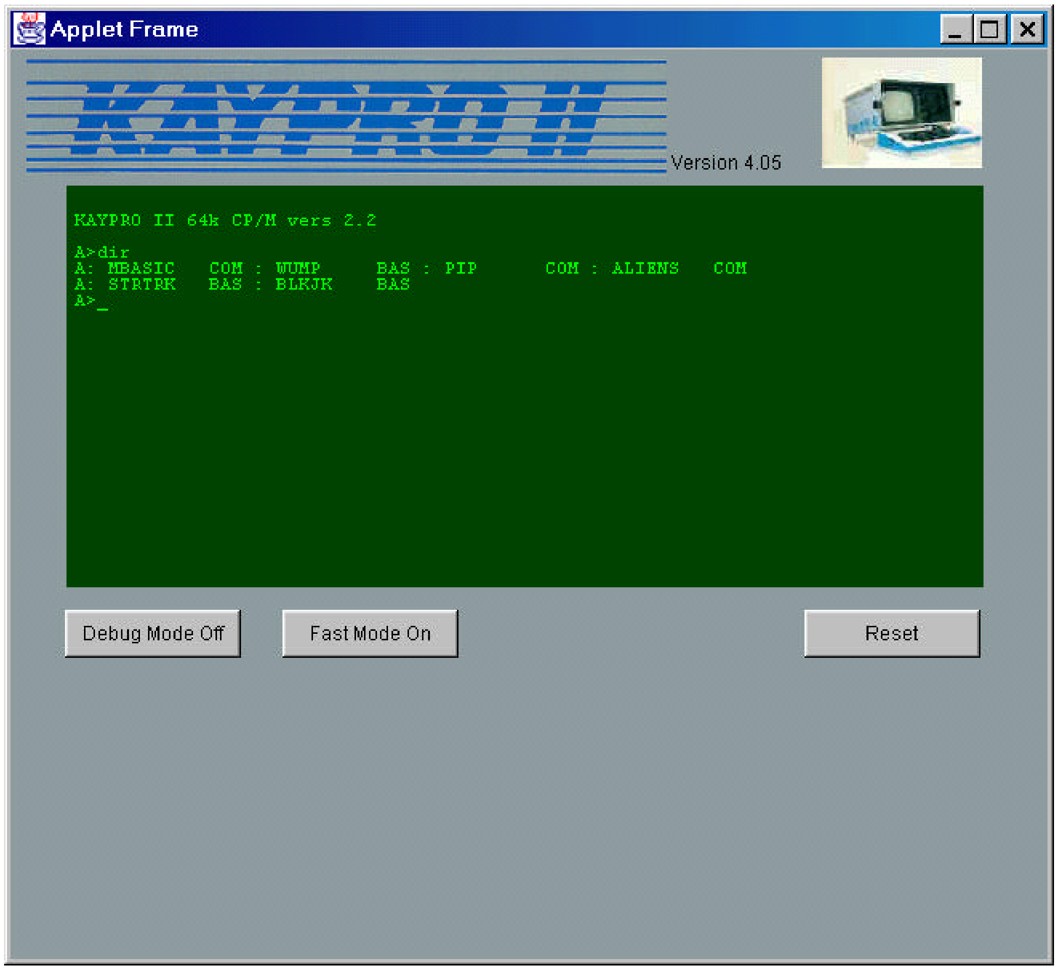
It is possible that the connection you seek is down. In some cases, it may simply require trying back later.

Make sure that your browser is Java enabled. If it is not, you will have to acquire one that is. See System Requirements section for Java requirements.

[2] For more information on how to point the browser to a specific http page, please see your browser manual.
[3] Depending on connection speed.
[4] Not all browsers work this way. In fact, some animations stop while the Java code is being loaded. Before giving up, wait a few minutes even if the animation has stopped.

## The Main User Screen

Once the emulator is loaded, the following screen should appear.  This is the screen, as it appears in its initial form.  Selecting some options may change the appearance.  This is described below.



```
KAYPRO II 64k CP/M vers 2.2

A>dir
A: MBASIC   COM : WUMP     BAS : PIP      COM : ALIENS   COM
A: STRTRK   BAS : BLKJK    BAS
A>_
```

**Figure 2, Initial Screen**

The large dark area of the user screen represents the Kaypro II Screen.  This area should appear in dark green.  The color was chosen to represent a green phosphor screen.  Actual color may vary on some systems.

A spinning rotation of a Kaypro II computer appears in the upper right corner.  These images may take a few seconds to load.

The Initial Screen is where you will type commands into the Kaypro II emulator and receives output from the emulated Kaypro II.  The screen contains a number of buttons at the bottom.  They are:

- Debug Mode Off
- Real Mode
- Reset

Some of the buttons change text when pressed.  The specific operation of each is described below.

### *Debug Mode Off Button*

This button toggles the debug mode on or off when clicked by the user.  Debug mode allows the user to step through code, view memory and set breakpoints. When the debug mode is turned on, another set of buttons is displayed.  These buttons include Run Mode, Step, Set Breakpoint, Generate Interrupt, View Options and Memory Dump (see Figure 3).  This button's default setting is debug mode off.

The button text is "Debug Mode Off" when the debug mode is turned off, or the button text is "Debug Mode On" when the debug mode is turned on.

### *Real Mode Button*

This button toggles between the fast mode and real mode. The fast mode will set the emulation into a mode where all text output from the emulator is displayed in a text box. This method of display is quicker, but is not a completely accurate depiction of the Kaypro II screen.

The real mode will set the emulator into a mode where the Kaypro II Screen will generate the screen just like the Kaypro II.  Real mode uses graphics to duplicate the actual Kaypro II character ROM.  Because it is graphical, it real mode display is slower.

The button text is "Fast Mode" when fast mode is turned on. The button text is "Real Mode" when the real mode is turned on.

### *Reset Button*

This button resets the Kaypro II (activates CPU reset line).

This button has the same function as the reset button on back of the actual Kaypro II computer.

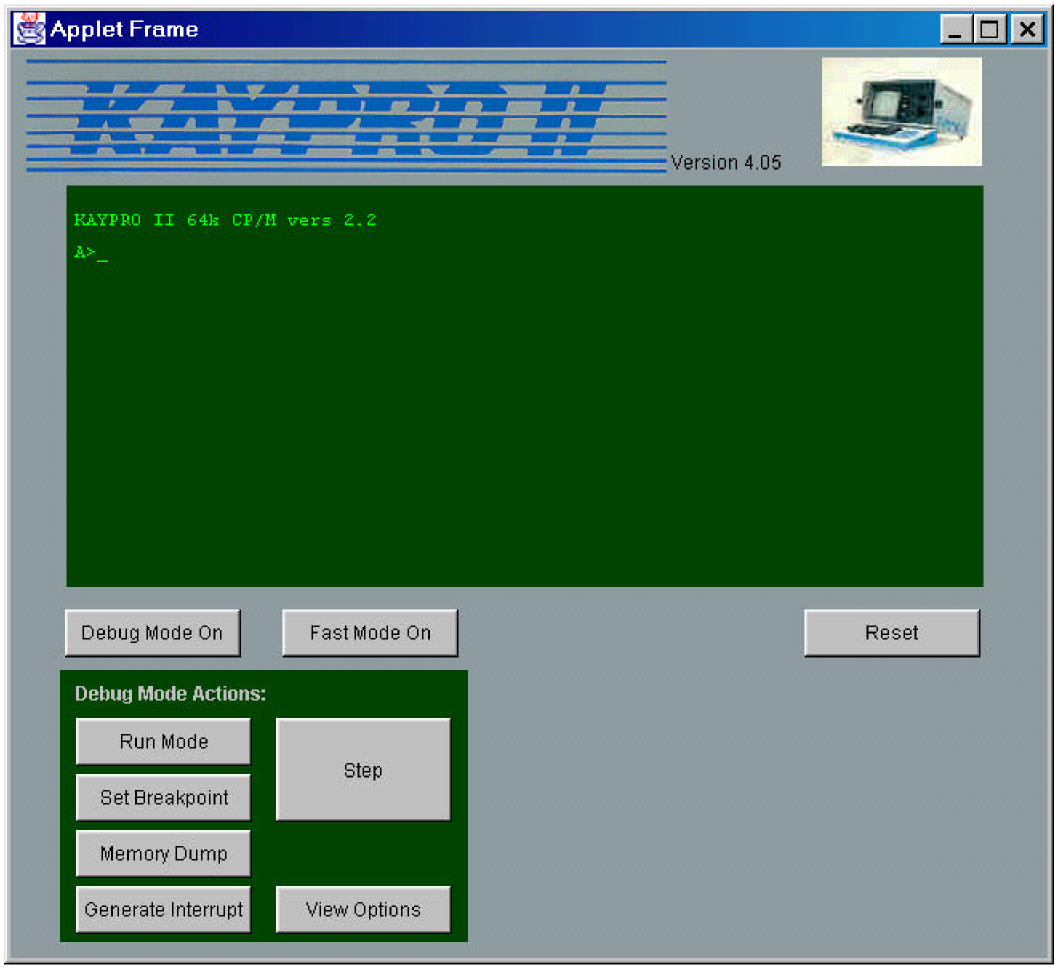
## Debugging

The emulator features a debug mode of operation. Figure 3 shows the user screen when the debug mode is activated.  This mode is activated when you hit the Debug button (see above).  When debug mode is entered, a debug window also appears (Figure 4).  The debug window allow you to display opcodes, list memory and see status messages.  The debug window disappears when you exit the debug mode. The debug mode allows you to perform a variety of debugging activities.

When debug mode is activated, additional buttons appear at the bottom of the screen. They are:

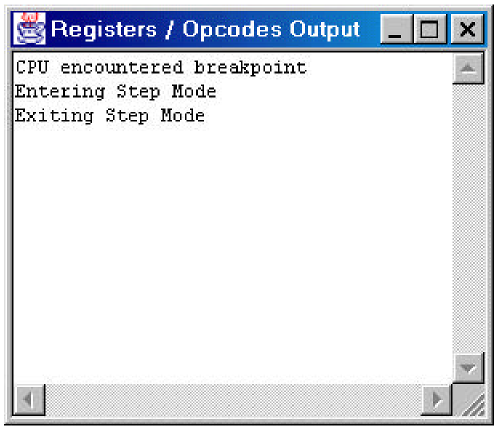
- Run Mode
- Step
- Set Breakpoint
- Generate Interrupt
- Memory Dump
- View options

Some of the additional buttons change text when pressed.  The specific operation of each is described below.



```
KAYPRO II 64k CP/M vers 2.2
A>_
```

**Figure 3, Debug Mode**



```
CPU encountered breakpoint
Entering Step Mode
Exiting Step Mode
```

**Figure 4, Registers/Opcode Window**

### *Run Mode*

Run mode allows you to run the emulation at full speed, or single-step the emulator. This button sets the emulator to run in either debug, or step mode. When in Run Mode, the emulator will automatically step through the machine language instructions. In Step mode, the processor will halt after each instruction is executed. To step to the next instruction, you must press the step button (described below).

You can list the contents of instructions and the executed opcode as well. This is described below.

The button text is “Step Mode” when the step mode is turned on (run mode turned off), or the button text is “Run Mode” when the run mode is turned on (step mode turned off). This button’s default is step mode turned on (run mode turned off).

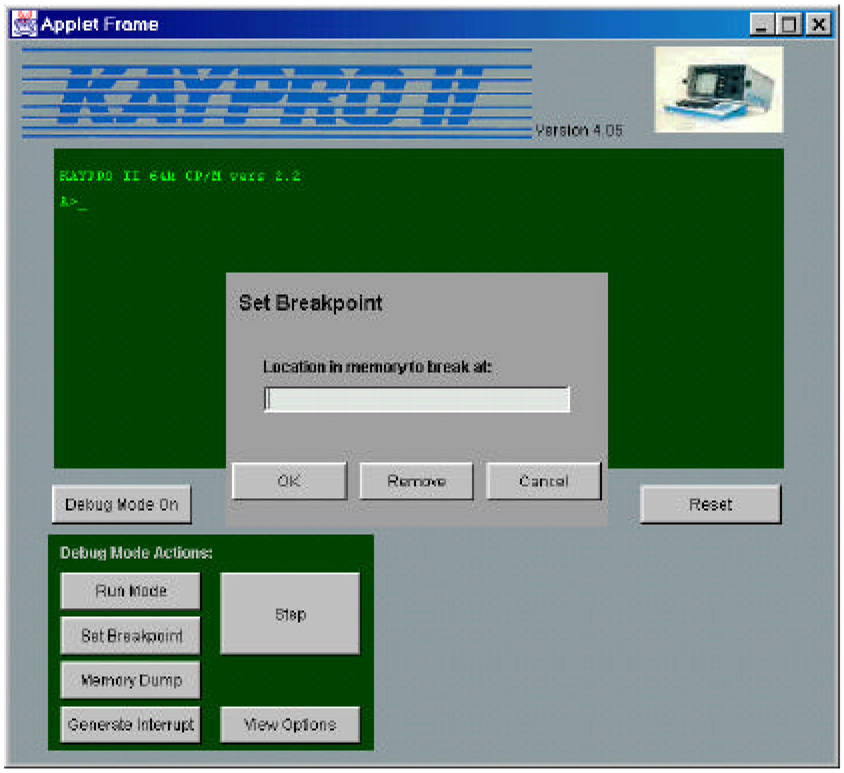
### *Step*

This button is enabled when step mode is turned on. This button single steps through the instructions in memory each time the user clicks the button. Only one instruction is executed at a time.

If you wish to display the instruction opcode, or the register contents after each instruction is executed, you must select the view options, as described below.

### *Breakpoint*

The user may set a breakpoint. A breakpoint specifies where a program will halt execution. You may set a single breakpoint. A breakpoint is set on a particular memory address. When the instruction at the specified memory address is executed, the emulator halts the CPU, and displays the current opcode and register contents (is specified - see View Options below). The emulator is then automatically put into step mode.



```
KAYPRO II 64k CP/M vers 2.2
A>_
```

Set Breakpoint

Location in memory to break at:

**Figure 5, Breakpoints**

Breakpoints may be entered in hex or decimal. Hex numbers are entered with a 0x prefix.

Examples would be:

    0x100
    0x00
    0x1f

Decimal numbers are entered as follows:

    100
    77
    23

Invalid numbers will display an error.

Breakpoints may be set or reset.  To set a breakpoint select OK from the breakpoint dialog.  To reset (disable) a breakpoint select Remove from the breakpoint dialog.

Typically a breakpoint is set and you enter run mode.  The emulation executes at full speed until the breakpoint is encountered.  When this occurs, the system is automatically set into step mode.  Care should be taken here.  It is easy to forget that a breakpoint is set. When the system halts it can be misinterpreted as a program "hanging."  If you feel the system has hung, check to see if a breakpoint has been encountered (system is in step mode).

If you turn debug mode off, the system automatically disables an active breakpoint.

***Difference between HALT command, and a breakpoint***

Please note that there is a difference between a breakpoint halting the emulator, and a HALT command halting the processor.  The Z-80 HALT command is recognized by the CPU, while a breakpoint is not.  The CPU executes the HALT command.  It consumes cycles.  The emulator itself processes a breakpoint.  Unlike the HALT command, it is not processed by the Z-80, and hence is not an instruction.  The emulator itself watches for breakpoints.  When one is reached, the emulation is stopped.

***Breakpoints, further defined***

Also, please note that breakpoints halt emulation only when an instruction is executed at the specified memory location.  No breakpoint is executed if a simple memory read or write (memory access) is executed.  For example, if a breakpoint is set for memory location 0x100, and an instruction writes a 0x77 to location 0x100, operation continues uninterrupted.

When the breakpoint button is pressed a dialog box allows the user to enter the location in memory that the user wants to stop the emulation.  After the location is entered the emulation runs until the location in memory is reached.

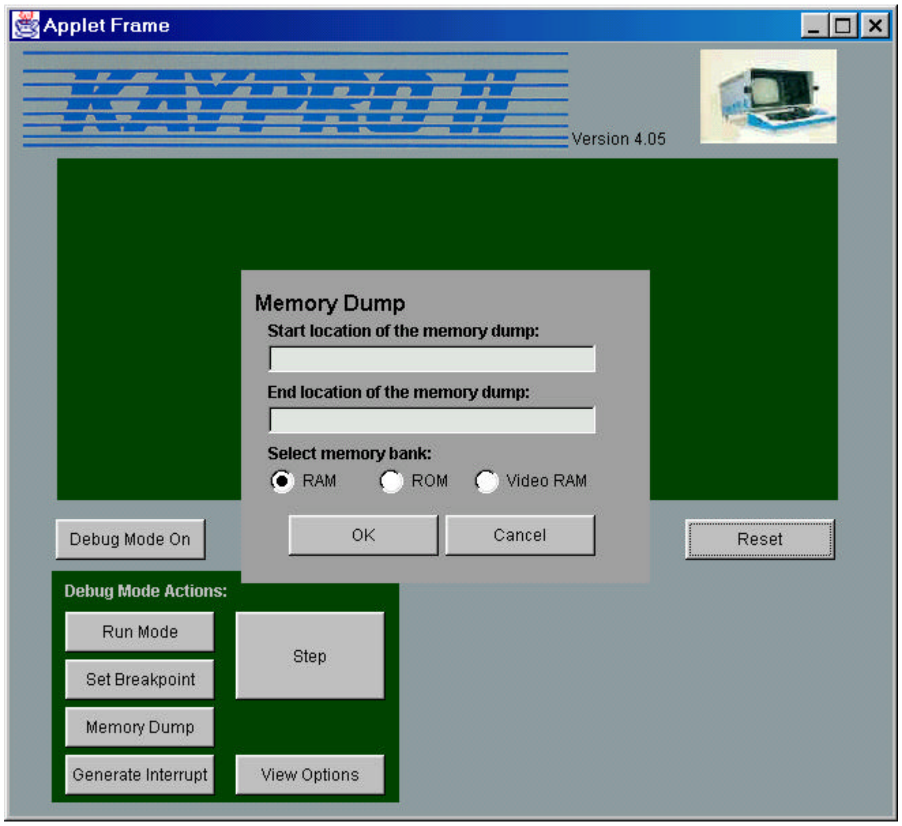
## *Generate Interrupt*

This button generates a NMI interrupt.  An NMI, or Non-Maskable Interrupt, can not be disabled by the CPU.  NMI's in the Kaypro II are typically used for disk access.  The NMI line is typically activated when a byte of data is to be transferred between an I/O port, and the disk controller chip.

The Generate Interrupt button allows you to simulate these actions.  With nonstandard software, the NMI may be used to test program function as well.

## ***Memory Dump***

Memory values can be displayed to a text window.  The memory dump feature allows you to display a specified range of memory.



Memory Dump

Start location of the memory dump:

End location of the memory dump:

Select memory bank:

[✓ RAM] ROM Video RAM

OK Cancel

**Figure 6, Memory Dump**

The Kaypro II emulator implements a bank switching methodology.  Bank one of memory contains 64K of straight linear memory.  Bank two overlays the lower portion of this memory with the system ROM, and video memory.

The memory dump button brings up a dialog box that asks the user the start location and end location of the memory dump.  The user may specify the memory they wish to dump as well.  The user can select from RAM, ROM or Video memory.

Figure 7 shows the location of various memories in the Kaypro II.

| Type | Low | High | Bank |
|---|---|---|---|
| RAM | 0x0000 | 0xFFFF | 0 |
| ROM | 0x0000 | 0x2FFF | 1 |
| Video | 0x3000 | 0x3FFF | 1 |
| RAM | 0x4000 | 0xFFFF | 1 |

**Figure 7, Memory Table**

<u>For example:</u>
If you would like to display the first 5 bytes of video memory, you would select a start value of 0x3000, an end value of 0x3005, and set the bank to video.

The emulator then dumps the desired memory into the registers/opcode window.

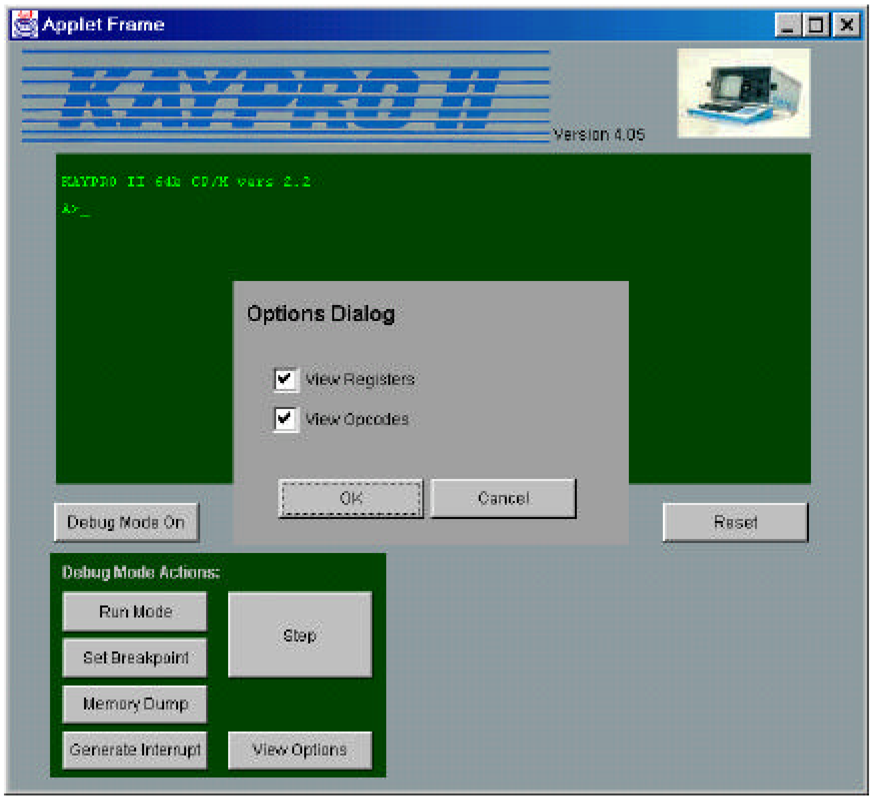
## *View Options*

The view options button allows you to adjust the debug display options. When the emulator enters step mode, it can display the CPU registers, and/or the opcode. If neither option is selected, no display will be generated when you hit the step button.

Selecting View Registers will display the Z-80 register contents on each single step operation.

Selecting View opcode display the source opcode for each single step operation.

This button brings up a dialog box (Figure 8) that allows the user to set a number of options for the debug mode (explained below).



```
KAYPRO II 64k CP/M vers 2.2
A>_
```

Options Dialog

- [✓] View Registers
- [✓] View Opcodes

**Figure 8, Debug Options**

### View Registers

When the user check marks this option the debug output window (Registers/Opcode window) displaying the current value of all registers whenever the processor breaks, or is single stepped.

### View Opcodes

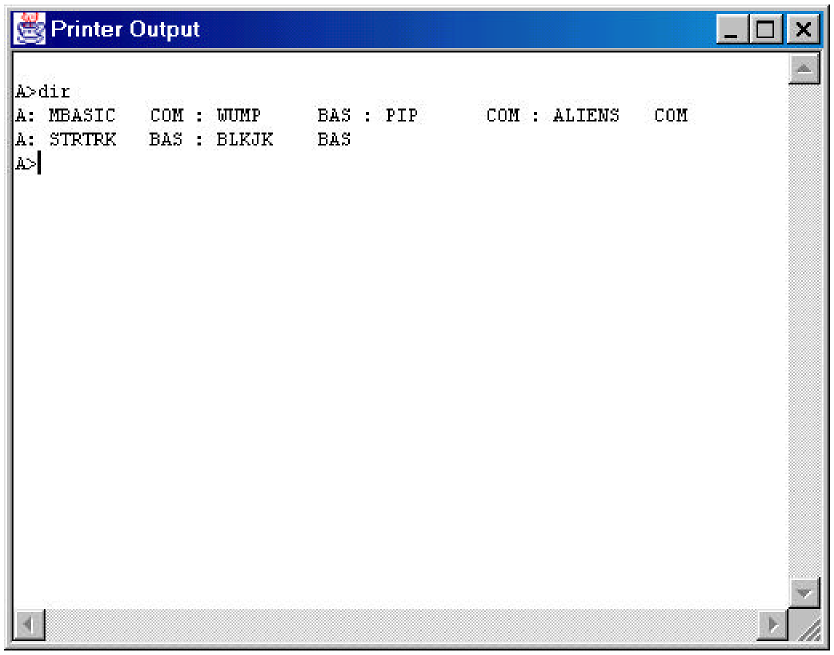
When the user check marks this option the debug output window (same window as registers) displays the current opcode being processed whenever the processor breaks, or is single stepped.

## Printer Screen

The Printer screen, as seen in Figure 9, is connected to the parallel input and output device for the Kaypro II system. The printer screen appears when the user starts the emulation, and automatically goes away when the emulation is stopped. The Kaypro II toggles output to the printer by typing <Ctrl> p. As mentioned above, it is assumed that a printer is connected to this port. Other peripherals are not supported.

All commands sent to the printer are displayed on the parallel port printer screen.

The emulation of the printer screen acts just like an ordinary printer. Clicking the printer screen scroll buttons located on the right side will scroll the display. The top button scrolls the screen downward and the bottom button scrolls the screen upward.



```
A>dir
A: MBASIC   COM : WUMP     BAS : PIP      COM : ALIENS   COM
A: STRTRK   BAS : BLKJK    BAS
A>
```

**Figure 9, Parallel/Printer Screen**