

XEROX

PARC

14 June 1983

To: Microcode Developers
From: Edward Fiala
Subject: How to Debug Dolphin Microcode With Midas
Filed On: [Indigo]<D0Docs>DebugWithMidas.Bravo, .Press

This is a short description of how to debug Pilot, Cedar, or AMesa microcode with Midas. It handles any combination of microswitch/Star keyboards and CSL/LF monitors. Documentation for Midas can be found on [Indigo]<D0Docs>D0Midas.Press. You should read both the Midas document and the part of "Dolphin Booting and Maintenance Panel Codes" ([Indigo]<D0Docs>MPCodes.press) which describes booting.

Midas is a debugging program that runs under the Alto OS on a different machine from the one you are debugging. You can run it on an Alto or on the Alto partition of a Dolphin. In either case, you need a special cable to connect the printer interface of the Alto or Dolphin running Midas to the printer interface of the Dolphin being debugged.

If you don't already have a Midas disk, you will need to build one:

1. Spin up a clean disk on your Alto. Boot the NetExec and invoke NewOS. Use the long installation dialog and erase the disk. If you going to run Midas on a Dolphin, you don't have to do this--just make sure there are 2000 free pages on your Alto partition.
2. Obtain the following files from the place where you get Alto subsystems (e.g., [Maxc]):

Micro.run	(microcode assembler)
MicroD.run	(microcode loader)
Ftp.run	
Bravo.cm	and execute this command file
RunMesa.run	
Empress.run	
Find.run	
Waterlily.run	

3. Load D0MidasRun.dm and retrieve the other files below from [Indigo] with Ftp:

<D0Source>D0Lang.Mc	
<D0>D0MidasRun.dm	
<D0>Midas.programs	
<D0>MemErrors.midas	
<D0>MakeLoaderFile.bcd	(microcode boot file builder)

Because of file name conflicts, you can only have one of the following dumps of microcode sources loaded at-a-time (except that Initial can coexist with Pilot microcode if Pilot microcode is loaded last):

<D0Source>CedarUCode.dm	for Cedar microcode
<D0Source>PilotUCode.dm	for Pilot microcode
<D0Source>AMesaSources.dm	for AMesa microcode
<D0Source>InitialSources.dm	for Initial microcode

4. Call Ed Fiala or send message to Fiala.Pa if you have problems with these procedures.

There are several important limitations to the ways in which Midas can be used to debug a Dolphin. The first is that you must BEGIN with Midas. If instead you attempt to attach to a machine which is in an interesting state, then Midas will boot the machine while activating its Kernel microcode, and all RM registers and the TPC's for the tasks will be reset (possibly storage will survive booting, however).

The second limitation is that the various microcode systems that you run must all reserve space for the Midas Kernel and have an appropriate linkage between the fault handler and the Midas Kernel. Initial has an IMReserve for the Midas Kernel; if you assemble the Pilot microcode with WithMidas=1 in GlobalDefs.Mc, then Pilot also has an IMReserve for the Midas Kernel. SDD Pilot microcode at the present time is built with WithMidas=1, but the Cedar and Tor variants are not, and the normally released Alto emulator overwrites the Midas Kernel, so you will need to obtain or create special debugging versions of these to debug from Midas.

The third limitation is that you cannot activate Pilot directly from Midas by loading and running .Mb files. Instead, you must install the Pilot germ and microcode on the Dolphin SA4000, and then load the microcode by running the Initial microcode from Midas. The microcode installed on the SA4000 must be the SAME as that on your Midas debugging disk. In addition, you must have a Physical Boot Volume set.

Using Midas to Boot Pilot

Boot the Midas disk and type "Midas/i Pilot". The Pilot.Midas command file will first load Initial.Mb, insert a breakpoint at RamLoaded, and start Initial at SAPilotStart. If all goes properly, Initial will read the Pilot microcode and all of its overlays from your SA4000 into storage, and the breakpoint at RamLoaded will be hit after LoadRAM has loaded Pilot1 into the microstore; then Pilot1 symbols are loaded by RunProg and the RamLoaded breakpoint is reinserted in octal (at IMX 316); this is necessary because all previous symbols and breakpoint information (which pertained to Initial) have been flushed by Run-Prog; since the space occupied by the LoadRAM module in Initial has been IMReserved in Pilot, the RamLoaded symbol is no longer available.

The Pilot.Midas command file pauses at this point with a message saying that Pilot1 is loaded, and you may abort the command file or continue it. If you continue the command file, it will pause again at BootEmulators in Initialize.Mc after completing all device initialization but before loading the Pilot2 overlay. If you continue from there, it will pause a final time after loading the Pilot2 overlay; if you continue from there, your physical boot volume will be started.

Assuming that you abort the first time the Pilot.Midas command file pauses, then you have the following options:

- a) Kill the IMX 316 breakpoint and proceed, letting Pilot start normally. After it is running, you can get control with Midas control-C (or by selecting and left-buttoning "Abort") and establish a suitable debugging context as discussed below.
- b) Insert any breakpoints in Pilot1 that you wish and continue. This allows debugging initialization and other Pilot1 microcode.
- c) If you want to debug the terminal microcode, then the driver for the LF keyboard system is included in Pilot1, so you can insert breakpoints directly; for a CSL keyboard system (with either a CSL or LF monitor), you must proceed from the RamLoaded breakpoint--the next time you get to that breakpoint, the microcode for that terminal, obtained from the first or second overlay in PilotD0.Eb or CedarD0.Eb, will have been loaded into the microstore. See below for establishing a debugging context.

To debug Pilot2 microcode, you should continue the command file until it indicates that Pilot2 has been loaded; then abort it.

Displaying Pilot Microcode Symbols From Midas

When Pilot has been loaded and started as described above, you can setup a debugging context by doing Run-Prog on PILOT1SYMBOLS, PILOT2SYMBOLS, CSLSYMBOLS, or CSLFSYMBOLS. Run-Prog clears all previous symbols and breakpoints and loads symbols from one of the four Pilot overlays--it is impossible to have symbols from more than one overlay concurrently active. PILOT1SYMBOLS and PILOT2SYMBOLS also setup the middle column of the Midas display with registers generally of interest when debugging Pilot emulator code; CSLSYMBOLS and CSLFSYMBOLS setup the middle column with registers generally interesting when debugging one of the display drivers.

At any time you can change the contents of the Midas display by doing Read-Cmds on a variety of files in the menu with suggestive names. Among these are the following:

BBREGS	Displays BitBlit registers in middle column.
RDCREGS	Displays RDC registers in middle column.
GCDEBUG	Displays CedarGC registers.
MEMERRORS	Displays memory error registers in right column.
TX*	Various TextBlit registers.

Using Midas to Debug AMesa microcode

1. Obtain a special debugging version of the Alto emulator that contains an IMReserve for the Midas Kernel--normal Alto microcode overwrites the Kernel and cannot be used. [Indigo]<D0>NewAMesa.Mb with CSL keyboards or [Indigo]<D0>LFAMesa.Mb with LF keyboards are suitable; put one of these on your Midas disk.

2. Boot the Midas disk and type "NewAMesa;L"; this loads the Alto microcode. Start the Alto emulator with "EGO;G" "KGO;G" or "KGOP2;G" for ether boot, partition 1 boot, or

partition 2 boot, respectively. The Alto Executive will appear shortly on the screen.

Eprom Microcode

Sources and other files for the Rev-L EPROM microcode are on [Indigo]<D0Source>Proms>Rev-L>*. After you rebuild the .mb file for the EPROM microcode from the sources, you can debug it from Midas as follows:

1. Select Run-Prog on the Midas menu.
2. When the Run-Prog menu appears, select EPROM.

When SA4000Boot is loaded and the next menu appears, you can select "CONTINUE" to continue the boot sequence with the Initial microcode, obtained from the Initial microcode area of the SA4000.

Loading Initial

To load the Initial microcode, use the Initial command file:

1. Select Run-Prog on the Midas menu.
2. When the Run-Prog menu appears, select INITIAL.

When the command file is finished, the display will show some symbols of interest for the MemInit.Mc module. The various starting addresses are on pages 1 and 2 of Initial.Mc; they include the following:

SAAltoStart	boots and starts AMesa from the SA4000.
SAPilotStart	boots and starts Pilot from the SA4000.
EtherAltoStart	boots and starts AMesa from the 3 mb Ethernet.
EtherPilotStart	boots and starts Pilot from the 3 mb Ethernet.

Some interesting breakpoints in Initial are the following:

RamLoaded	(IMX 316) where LoadRAM finishes.
IMap	beginning of the map and storage test.
imRepeatStorageTest	after map test.; here you can change the contents of the SoftQThreshold register to 100b if you want to make the storage test put pages with correctable errors into service.
MemInitDone	end of storage test.
MicrocodeLoaded	after loading microcode into the VM block at 1400b.