MotorolaProcessorRTEMS How-to Flash an mvme6100

How to Flash a Motorola MVME6100 Processors to use RTEMS

Now we'll get the software that the SBC will need to boot, and write it to flash memory. In the example output
that follows, the text that you need to enter is **highlighted and in bold**. Please note that the mvme5500 does requires a few different commands from the mvme6100. To flash an mvme5500 click here.

- Login to your computer, and run a terminal emulation program. If your are logged into a taylored linux node, then you will need special access to the linux box serial port. Users with sudo privledge on this particular node will have such access as will users that have ownership of the colsole port on this node. To obtain the latter type of access you can send a request to unix-admin@slac.stanford.edu asking to make taylor give the person logged in at the console ownership of the serial port via console.perms(5).

`The terminal emulation software allows you to see the output from the SBC, and to type commands to the firmware monitor.`

- - On Linux, you can use minicom. On MacOS X use zterm. On Windows, TeraTerm is popular. You might need to have administrative privilege to use these programs on certain platforms.
  - Make sure the settings for the connection are correct; 8 data bits, 1 stop bit, no parity, 9600 baud. Also make sure that you're accessing the correct serial COM port on the computer.
- Power up the crate or enclosure holding the SBC. After a few seconds, some text will appear in the terminal emulation window. You'll be prompted to enter a command as shown below.

```
Copyright Motorola Inc. 1999-2007, All Rights Reserved
MOTLoad RTOS Version 2.0, PAL Version 2.1 RM02
Wed Feb 7 11:32:02 MST 2007

MPU-Type =MPC74x7
MPU-Int Clock Speed =1266MHz
MPU-Ext Clock Speed =133MHz
MPU-Int Cache(L2) Enabled, 512KB, L2CR =C0000000
MPU-Ext Cache(L3) Enabled, 2MB, 211MHz, L3CR =DC026000

PCI bus instance 0 =64 bit, 133 Mhz, PCI-X
PCI bus instance 1 =64 bit, PCI

Reset/Boot Vector =Flash1

Local Memory Found =20000000 (&536870912)
User Download Buffer =006B7000:008B6FFF

MVME6100>
```

1. First, set the time of day using the set command. The time is of the format **MMDDYYHHMMSS**. So, **November 3, 2005** at **2:27** PM would be set like this:

```
MVME5100>set -t110305142700
MVME6100>
MVM6100>time
THU NOV 3 14:27:01 2005
```

Next, download the NetBoot program into the IOC. Use the network address that you obtained during "Inventory".
It needs to be entered with the *tftpGet* command, after the "-c" (for client) on the command line. Note that the backslash () is put at the end of the line as a continuation marker. You can simply type the remainder of the command without the backslash.

In order to use the tftp server your ethernet wall connection MUST be on the lclsdev subnet (ie 134.79.219.xxx) or the lclsdmz subnet (ie 134.79.151.xxx). Note that the "xxx" needs to be replaced with the correct value from your own address. If your ethernet wall connection is NOT on the lclsdev or lclsdmz subnet then you can load the flash in your cpu from another configured cpu (of the same model). This is done by copying the flash in the configured cpu over the VME backplane to the cpu you intend to flash. Instructions on how to perform this operation can be found by clicking here.

	Subnet	**TFTP Server**	**Gateway**	RTEMS Image
Development	lclsdmz	lcls-dev1 134.79.219.136	132.79.219.1	/rtems/4.10.2/powerpc-rtems/beatnik/img/netboot.flashimg.bin
LCLS	lclsioc	lcls-srv04 172.27.8.28	172.27.8.1	/rtems/rtems-4.10.2/target/ssrlApps/powerpc-rtems/beatnik/img/netboot.flashimg.bin
FACET	facetca	lcls-srv04 172.27.8.28	172.27.72.1	/rtems/rtems-4.10.2/target/ssrlApps/powerpc-rtems/beatnik/img/netboot.flashimg.bin

The "-s" parameter allows us to specify the server address. For the lclsdev subnet, it is 134.79.219.12, the address of the lcls-dev2 server. For the lclsdmz subnet, it is 134.79.155.15, the address of the mccdev VMS server. The "-m" parameter indicates a netmask to use to access the server.
Finally, the "-f" parameter indicates the name of the file (under /tftpboot on lcls-dev, elsewhere on mccdev) containing the sequence of values to be loaded into memory.

```
MVME6100>tftpGet -c<cpu_ip> -s<tftp server_ip> -g<gateway_ip> -m255.255.252.0 -f<rtems_flash_image>

Network Loading from: /dev/enet0
Loading File: mv6100/netboot-mve.flashimg.bin
Load Address: 005C3000

Client IP Address = <cpu_ip>
Server IP Address = <server_ip>
Gateway IP Address = <gateway_ip>
Subnet IP Address Mask = 255.255.252.0

Network File Load in Progress...

Bytes Received =&614736, Bytes Loaded =&614736
Bytes/Second =&204912, Elapsed Time =3 Second(s)
MVME6100>
```

For the mvme6100 processor, the command is identical. The same version of the NetBoot program is downloaded, although the output on the screen will be slightly different.
At this point, we'll disable the network activity on all adapters, to ensure nothing interrupts us while we copy the image to flash memory. This is not essential, because it is not likely that any network activity would have an effect on the system at this point.

Example:

```
MVME6100> tftpGet -c134.79.219.146 -s134.79.219.136 -g132.79.219.1 -m255.255.252.0 -f/rtems/4.10.2/powerpc-rtems/beatnik/img/netboot.flashimg.bin
Network Loading from: /dev/enet0
Loading File: /rtems/4.10.2/powerpc-rtems/beatnik/img/netboot.flashimg.bin
Load Address: 006B7000
Download Buffer Size = 00200000

Client IP Address      = 134.79.219.146
Server IP Address      = 134.79.219.136
Gateway IP Address     = 132.79.219.1
Subnet IP Address Mask = 255.255.252.0

Network File Load in Progress...

Bytes Received =&824776, Bytes Loaded =&824776
Bytes/Second   =&824776, Elapsed Time =1 Second(s)
MVME6100>
```

```
MVME6100> netShut
/dev/enet0 Disabled
/dev/enet1 Disabled
MVME6100>
```

Now we can actually copy the memory image that we downloaded in step 4 above from RAM into Flash. That is, we're copying it from temporary memory into a more permanent memory that will last after power failures or restarts. If you notice in the *tftpGet* output, it tells us the load address, in this example for the mvme5500, it was 005C3000. The load address is the location to which the file was copied.
We'll copy this file from that memory location into the Flash memory, so it won't be lost. In this example, the flashProgram command is used; the "-s" parameter indicates the source of the data to be copied, in our example 0x005C3000. The "-v" parameter tells the software to be verbose about what it's doing. It prints a few lines, then asks you to confirm.

```
MVME5500> flashProgram -s5C3000 -v
Source Starting/Ending Addresses =005C3000/006C2FFF
Destination Starting/Ending Addresses =F2000000/F20FFFFF
Number of Effective Bytes =00100000 (&1048576)

Program Flash Memory (Y/N)? Y
Virtual-Device-Number =00
Manufacturer-Identifier =89
Device-Identifier =18
Virtual-Device-Number =01
Manufacturer-Identifier =89
Device-Identifier =18
Address-Mask =FE000000
Flash Memory Programming Complete
MVME5500>
```

For the mvme6100, the load address is 006B7000. Accordingly, for that processor, the command would be:

```
MVME6100> flashProgram -s6B7000 -v
Source Starting/Ending Addresses =006B7000/007B6FFF
Destination Starting/Ending Addresses =F4000000/F40FFFFF
Number of Effective Bytes =00100000 (&1048576)

Program Flash Memory (Y/N)? Y
Virtual-Device-Number =00
...
```

Test the Boot Process

You've "flashed" the processor. Now, you can emulate the boot process to test that it will work.

- First, we're going to confirm that the flash memory was written correctly, and test the boot procedure at the same time. The bmw command performs a Block Move Word request, and it ensures that the data being copied is word aligned. The idea is to do exactly what the normal boot procedure will do with each subsequent reboot of the processor; it will copy the contents of the persistent flash memory into main memory using the bmw command.
- We could easily just tell the processor to boot from the downloaded image that is still in memory, at address 5C3000 (for the mvme5500). But using bmw to copy the memory from flash will ensure that the flash memory contains the right stuff.
- If we look back to the output from the flashProgram command in a previous step, we see that the "Destination Starting/Ending Addresses" was printed out. This is the address in Flash memory that has the copy we need. This is handy, since the **bmw** command needs the start and end addresses (rather than the number of bytes, as one might expect.)
    - The "-a" and "-b" parameters indicate the start and end addresses of where we want to copy from (the source of the copy.) The "-c" parameter indicates the location to which the data will be copied. In this example, we'll copy the data to location 4000000 in main memory.
    - First, type the following:

```
MVME6100> bmw -af4000000 -bf40fffff -c4000000
MVME6100>
```

-
    - After copying the data, we can use the go command to start executing the instructions represented by that data. Shortly after the program begins, it will attempt to load the rest of the RTEMS software into memory from the network.
    - Instead of allowing that to happen, we can press any character on the keyboard to interrupt the process. You only have a couple of seconds to interrupt the process, but at that point, you will have the attention of the netBoot software, and can change its configuration parameters. Keep in mind that the prompt only stays there for a few seconds, so pay close attention and press any key when you see that prompt.

```
MVME6100> go -a4000000
Exception handling initialization done

---

Welcome to rtems-4.7(PowerPC/PowerPC 7455/mvme5500) on MVME5500-0163
Build Date: 20050506PDT18:52:37

---

Now BSP_mem_size = 0x1FE00000
BSP_Configuration.work_space_size = 15000
Registering /dev/console as minor 0 (==/dev/ttyS0)

RTEMS bootloader by Till Straumann <strauman@slac.stanford.edu>
$Id: netboot.c,v 1.21 2005/04/26 02:05:03 till Exp $
CVS tag $Name: $
Your CPU Temperature calibration changed or was not initialized...
To calibrate the CPU TAU (thermal assist unit), you must observer the following steps:
1. Let your board stabilize to ambient temperature (POWERED OFF)
2. Measure the ambient temperature Tamb (deg. C)
3. Power-up your board and read the Temperature printed by SMON/FDIAG (Tsmon)
NOTE: use ONLY the info printed IMMEDIATELY after powerup
4. Set the calibration offset to Tamb - Tsmon

Type any character to abort netboot: <press any keyboard character here>2

Press 's' for showing the current NVRAM configuration
Press 'c' for changing your NVRAM configuration
Press 'b' for manually entering filename/cmdline parameters only
Press '@' for continuing the netboot (BOOTP flag from NVRAM)
Press 'd' for continuing the netboot; enforce using BOOTP
Press 'p' for continuing the netboot; enforce using BOOTP
but use file and cmdline from NVRAM
Press 'm' for continuing the netboot; enforce using NVRAM config
Press 'R' to reboot now (you can always hit <Ctrl>-x to reboot)
Press any other key for this message
```

- At this point, you've confirmed the flash memory contains the right data. The netBoot software will recognize several commands from your keyboard, and it's listed them after you've interrupted its boot process.
- Normally when it runs, netBoot attempts to download the software that you ultimately want to use. To do so, it keeps several parameters in non-volatile memory (NVRAM.)
- Now you're going to set those various parameters and record them in NVRAM. Press "c" to change the parameters that netBoot will use to download software. Keep in mind that your own processors IP address must be entered under "My IP", just as it's name.
- Typically, only the fields marked "My IP", "My name" and "Command Line Parameters" will have values unique to your processor. All other fields should be entered as you see here. If you have questions, please ask a Controls Software Engineer from theEPICS Team.
  - The field "Command Line Parameters" is set to:
    - (booting from NFS): **INIT=/boot/epics/iocCommon/user**, followed by a path to your boot-up script. If you're using the AFS filesystem, which is common at SLAC, then your path will typically be something like **/u/*<two-chars>*/*<login user-id>*/path-to-example**. If you are booting out of the shared lcls group space, the boot-up script is **/boot/g/lcls/epics /iocCommon/"<your-node-name>"/st.cmd**. See the example below.

```
Changing NVRAM configuration
Use '<Ctrl>-k' to go up to previous field
Use '<Ctrl>-r' to restore this field
Use '<Ctrl>-g' to quit+write NVRAM
Use '<Ctrl>-c' to quit+cancel (all values are restored)
Use '<Ctrl>-x' to reboot
Boot file (e.g., '/TFTP/1.2.3.4/path', '~rshuser/path' or 'nfshost:/dir:path'):
>magenta:afsnfs2:/afs/slac:/package/4.7.1/target/ssrlApps/powerpc-rtems/beatnik/bin/rtems.exe~~
Command line parameters:
>INIT=/boot/g/lcls/epics/iocCommon/lclsdev-33/st.cmd
Server IP: ><server_ip>
Gateway IP: ><gateway_ip>
My IP: ><cpu_ip>
My netmask: >255.255.252.0
My name: ><cpu_nodename>
My domain: >slac.stanford.edu
Loghost IP: >
DNS server 1: >134.79.18.40
DNS server 2: >134.79.18.41
DNS server 3: >
NTP server 1: >134.79.18.40
NTP server 2: >134.79.18.41
NTP server 3: >134.79.18.34
Use BOOTP: Yes, No or Partial (-> file and
command line from NVRAM) Y-N-P>N
Autoboot Delay: 0_.30secs (0==forever) >5
CPU Temp. Calibration - (LEAVE IF UNSURE) >

NVRAM configuration updated
```

1. Now you can continue with the boot process. When you're done setting the values in the previous step, you'll be given the list of netBoot commands again. Type "m" to continue with the boot process to make sure everything is correct.
2. You'll see a very long set of messages come from the processor console, ending in a prompt that says "Cexp>". RTEMS has successfully started running on your processor.
3. To get back to MOTLoad, type:

```
Cexp>rtemsReboot()
```

Saving an Automatic Boot Script

Those few steps that you took before changing the NVRAM are all that are needed to reboot the computer again. Now, we can save those steps so the processor can perform them automatically each time it restarts.

The MOTLoad firmware in Motorola processors makes use of environment variables to do various things. These variables retain their values between restarts, and when the power is off. They are stored in a reserved area of NVRAM.

In particular, the environment variable named mot-script-boot contains a string of characters that are interpreted as commands. These commands are executed each time the processor starts.

The gevEdit command is used to create or change an environment variable. These are different for each type of processor, because of the Flash memory address. Type the following:

```
MVME6100> gevEdit mot-script-boot
(Blank line terminates input.)
netShut
bmw -af4000000 -bf40fffff -c4000000
go -a4000000

Update Global Environment Area of NVRAM (Y/N)? Y
MVME6100>
```

Confirm that the mot-script-boot variable is set correctly. You can type gevShow, and the contents of the environment variables will be printed. Reboot the processor:

```
MVME6100> reset
```

Reboot the Processor

You can power cycle the crate or if you have a "Cexp>" prompt, you can type rtemsReboot() and the processor will restart. It will boot up automatically, loading the rtems.exe file specified in the NVRAM. Change that filename to boot another file as your IOC development proceeds.