DeviceMaster Bootloader Users Guide

Grant Edwards
Comtrol Corporation

June 21, 2001

1 Introduction

The DeviceMaster boards run a bootloader based on RedHat's "RedBoot" program. In addition to the RPSH-Si compatible TCP and MAC mode network interfaces RedBoot provides a boot console interface that can be used to perform various functions:

- View/modify board configuration:
  - Model number
  - Board revision
  - MAC address
  - IP configuration (address/netmask/gateway)
  - Password
  - Autoload timeout
  - Telnet enable/disable
  - HTTP authentication method
- Load file via serial port ([xyz]modem) or Ethernet (TFTP).
- Load file from flash.
- Save file to flash.
- Test serial port.

2 Important Differences From Older Products

There are several differences between the DeviceMaster and the RPSH-Si 2/4/8 port products:

1. **RPSH-Si** bootloader sends out a DHCP request and later passes the DHCP reply information to the downloaded application.

   DeviceMaster bootloader sends out a DHCP request to obtain basic IP address information. When the application program starts it sends out another DHCP request request further configuration information (netmask, name-server, etc.).

2. **RPSH-Si** bootloader will continue to send DHCP requests once per second if it gets no response.

   DeviceMaster bootloader will give up after a two attempts and disables IP networking.

3. **RPSH-Si 2-Port** does not implement DHCP lease renewal.

   DeviceMaster and RPSH-Si 4/8 port implements DHCP lease renewal.

4. **RPSH-Si** will always send DHCP requests unless a static IP address is configured. [1]

   DeviceMaster can be configured to disable IP networking (no BOOTP/DHCP requests will be sent).

5. **RPSH-Si** ignores the server and filename fields in the BOOTP/DHCP response.

   DeviceMaster will attempt to load via TFTP the specified file from the specified server if a filename and server address is present in the BOOTP/DHCP response. If the file is loaded successfully, it will then be executed.

3 Board Configuration Commands

These functions are performed via a command-line interface that is accessible via external serial port 0, the 4-pin debug header (57.6K, 8, none), or by telnet. When connected to the debug header or telnet you should see a `RedBoot>` prompt[2]

The external console port will be disabled on power-up. In order to start console services on that port, the string "#!DM" must be the first thing received on that port after power-up. When that string has been seen, the port will be enabled and a prompt will be displayed.

A password is required for telnet access, but no password is required for serial port access.

```
Comtrol DeviceMaster Boot Version 0.01
RedBoot(tm) debug environment - built 17:28:09, Mar  7 2001
Platform: Comtrol DeviceMaster (ARM 7TDMI)
Portions Copyright (C) 2000, Red Hat, Inc.
Portions Copyright (C) 2001, Comtrol Corp.
RAM: 0-7C0000
Id=0089,8897
FLASH: 0x05030000 - 0x05400000, 61 blocks of 0x00010000 bytes
each.
ks32C5000 eth: 00:C0:4E:0B:FF:FA  Hardware CRC
IP: 192.168.1.23, Default server: 0.0.0.0
RedBoot>
```

To see a list of available commands, type "help" followed by a carriage-return:

[1] We have had complaints about DHCP requests from customers using MAC addressing mode.

[2] The start-up messages are displayed only on the 4-pin diagnostic head and will not be visible via telnet unless the "version" command is entered.

```
RedBoot> help
       auth [noaccess,none,basic,md5,invalid]
           Set/show web authentication
   boardrev [rev-number]
           Show/set Board revision
      cache [ON | OFF]
           Manage machine caches
    disable
           Disable autoload of default app
       dump -b <location> [-l <length>]
           Display (hex dump) a range of memory
        fis {cmds}
           Manage FLASH images
         go [-w <timeout>] [entry]
           Execute code at a location
       help
           Display list of bootloader commands
         ip [addr mask gateway]
           Show/set IP address config
       load [-r] [-v] [-h <host>] [-m {TFTP | xyzMODEM | direct}] [-
b <base_address>] <file_name>
           Load a file
       loop  232|422|int port-number
           Run loopback test on port
        mac [XX XX XX XX XX XX XX]
           Show/set Ethernet MAC address
      model [model-number]
           Show/set Model number
   password [password]
           Set/Delete password
      reset
           Reset the system
     telnet [disable|enable]
           Set/Show telnet server enable
      terse
           Terse command response mode
       t485  port-number1 port-number2
           Run port-to-port 485 test
    timeout [seconds]
           Show/set bootloader timeout
    version
           Display RedBoot version information
```

The board configuration commands are ip, mac, model, boardrev, password, auth, timeout,

telnet. Typing the command will display the current value of that configuration item.[3] The command with a parameter will set the configuration item value.

3.1 Model Number

The model number is used by applications to determine what hardware features are available on the board. If the model number is set incorrectly, applications may not function correctly.

```
RedBoot> model
Model 5002120
RedBoot> model 5002113
Model 5002113
RedBoot>
```

3.2 IP Configuration

Setting the IP address to 0.0.0.0 will disable IP networking. Setting the IP address to 255.255.255.255 will cause the bootloader to use BOOTP to request an IP address.

```
RedBoot> ip
IP Config: IpAddr=192.168.1.23 IpMask=255.255.255.0 IpGate=192.168.1.1
RedBoot> ip 10.23.4.12 255.255.0.0 10.23.1.1
IP Config: IpAddr=10.23.4.12 IpMask=255.255.0.0 IpGate=10.23.1.1
RedBoot>
```

If only one parameter is provided to the IP command, the IP address value will be changed and the mask and gateway will be unaffected. Changes to IP configuration will take effect after reset.

3.3 MAC address

Sets or displays the MAC (Ethernet) address. Changes will take effect after reset.

```
RedBoot> mac
MAC: 00 C0 4E 0B FF FA
RedBoot> mac 00 c0 4e 0c 34 ea
MAC: 00 C0 4E 0C 34 EA
RedBoot>
```

3.4 Board Revision

Sets or displays the board revision.

```
RedBoot> boardrev
BoardRev 7
RedBoot> boardrev 3
BoardRev 3
RedBoot>
```

[3] For security reasons, the password command will not display the current password. If the password command is used with no parameters, the password will be set to the empty string.

3.5 Telnet Enable

Allows the user to enable or disable telnet access to the DeviceMaster. Accepted values are “enable” and “disable”:

```
RedBoot> telnet
Telnet enable
RedBoot> telnet disable
Telnet disable
RedBoot>
```

3.6 HTTP Authentication

Controls the type of authentication required for HTTP access. The types of authentication are:

noaccess – No HTTP access will be allowed – access forbidden error will be returned.

none – No authentication will be required.

basic – Plaintext password authentication required.

md5 – MD5 encrypted password authentication required.

invalid – No HTTP access will be allowed – invalid URL error will be returned.

```
RedBoot> auth
Auth: none
RedBoot> auth basic
Auth: basic
RedBoot>
```

3.7 Password

Changes the DeviceMaster password used to authenticate telnet and HTTP access. Unlink other commands, the password command will not display the current value of the password. The password command *always* sets the value of the password. The password length is limited to 15 characters.

```
RedBoot> password
Password ''
RedBoot> password FooBar1
Password 'FooBar1'
RedBoot>
```

3.8 Autoload Timeout

Sets the number of seconds after network initialization that the bootloader will wait before starting the default application program from flash ROM. A timeout value of 0 will disable auto-loading of the default application. The maximum value is 255 seconds.

```
RedBoot> timeout
Timeout 9 seconds
RedBoot> timeout 5
Timeout 5 seconds
RedBoot>
```

4 Terse Mode

In order to provide a console interface more amenable to programmatic control, the bootloader may be put into “terse” mode. The significant features of terse mode are:

- No prompt is issued.
- Command strings are not echoed.
- All responses consist of a single line. If the command was successfull, the reponse string begins with “+ ”. If the command failed, the response string begins iwth “- ”.

It is possible that bugs in some commands will still return diagnostic information. It is recommend that programs ignore lines that do not begin with “+ ” or “- ”.

5 Loading a File

It is possible to load a program or data file into RAM from three sources: serial port (the 4-pin diagnostic header or external port 0), Ethernet (from a TFTP server or via TCP), or from flash ROM on the DeviceMaster board. Loading from flash ROM is described in section 6.4. Loading from serial port or Ethernet is done using the “load” command. Files loaded with the “load” command default to Motorola S-Record format (which must end with a single “Entry-Point” record).

It is possible to load a binary file with the “load” command by using the -r and -b options.

5.1 Loading via Ethernet

5.1.1 TFTP

Loading a file from a TFTP server is done by using the “-h” option to load:

```
RedBoot> load -v -h 192.168.1.2 socket.srec
Entry point: 0x00000384, address range: 0x00000000-0x000718a8

RedBoot>
```

The “-v” option will cause a spinning status indicator to be displayed as the file is loaded.

5.1.2 TCP

Loading a file via TCP can be done by connecting to the telnet server (TCP port 23), logging in, and issuing a “load -m d” command. The telnet server will then expect to read an S-record file as input. Each line read will be acknowledged with a single linefeed character. When the last line in the S-record file (which must be an Entry-Point record) has been processed, a load summary will be printed.

5.2 Loading via serial port

Loading an S-record file via a serial port is done with a load command specify either xmodem or ymodem protocol:

```
RedBoot> load -m ymodem
```

Start ymodem transfer program.

```
Entry point: 0x00000384, address range: 0x00000000-0x000718a8

RedBoot>
```

It is also possible to load a binary file via x-modem. Binary download will be approximately 2-3 times faster except for sparse files. They entry point for a binary file will be the base address specified in the load command. For DeviceMaster eCos executables, the base address should be normally 0:

```
RedBoot> load -b 0 -r -m x
```

start x-modem download

```
CRC mode, 4085(SOH)/0(STX)/0(CAN) packets, 2 retries
RedBoot>
```

6 Flash Image System

RedBoot implements a rudimentary file system that allows program and data files to be stored in flash. The are various “fis” commands that can be used to manipulate this file-system. Typing “fis” followed by a carriage-return will display a list of sub-commands:

```
RedBoot> fis
*** invalid ‘fis’ command: too few arguments
Usage:
  fis create -b <mem_base> -l <image_length> [-s <data_length>]
                  [-f <flash_addr>] [-e <entry_point>] [-r <ram_addr>] [-
n] <name>
  fis delete name
  fis erase -f <flash_addr> -l <length>
  fis free
  fis init [-f]
  fis list [-c]
  fis load [-b <memory_load_address>] [-c] name
```

6.1 fis list

The “fis list” command displays a directly of the files currently stored in flash ROM:

```
RedBoot> fis list
Name              FLASH addr   Mem addr    Length      Entry point
FIS directory     0x053F0000   0x053F0000  0x00010000  0x00000000
default           0x05030000   0x00000000  0x00080000  0x00000384
RedBoot>
```

The “FIS directory” entry will always be there and is the file in which flash ROM bookkeeping information is stored. The “default” file is the program that will be run by RedBoot on startup. This will be described further in section 8. The list, delete, and create commands are the most frequently used:

6.2 fis delete

The “fis delete” command is used to delete a file from flash:

```
RedBoot> fis delete default
Delete image ‘default’ - are you sure (y/n)? y
... Erase from 0x05030000-0x050b0000: ........
... Erase from 0x053f0000-0x05400000: .
... Program from 0x007a0000-0x007b0000 at 0x053f0000: .

RedBoot> fis list
Name              FLASH addr   Mem addr    Length      Entry point
FIS directory     0x053F0000   0x053F0000  0x00010000  0x00000000
RedBoot>
```

6.3 fis create

The “fis create” command is used to store a region of RAM as a file in flash ROM. The basic form of the command is:

```
RedBoot> fis create -b 0 -l 0x10000 foobar
... Erase from 0x05030000-0x05040000: .
... Program from 0x00000000-0x00010000 at 0x05030000: .
... Erase from 0x053f0000-0x05400000: .
... Program from 0x007a0000-0x007b0000 at 0x053f0000: .

RedBoot> fis list
Name              FLASH addr   Mem addr    Length      Entry point
FIS directory     0x053F0000   0x053F0000  0x00010000  0x00000000
foobar            0x05030000   0x00000000  0x00010000  0x00000000
RedBoot>
```

It is also possible to specify a program entry point with the "-e" option and a specific location for the file in flash ROM with the "-f" option. If no address or length is specified, it will use the address and length of the S-Record file most recently loaded to RAM via serial port or Ethernet.

6.4 fis load

The "fis load" command loads a file from flash ROM into RAM:

```
RedBoot> fis load foobar
```

7 Executing a Program

The "go" command is used to execute a program. If no starting address is provided as a parameter to the "go" command, the entry address of the last file loaded into RAM will be used.

8 Default Application

The DeviceMaster version of RedBoot will wait for 5 seconds after startup for connections from a host. If no connection from a host is made, the bootloader will look for a file named "default" in the flash file-system. If a file named "default" exists, it will be loaded and executed. If a host initiates a download or if the the "dis" command is entered during the first 5 seconds, the "default" program will not be loaded.

Here is an example of the commands used to load a program from a TFTP server and save it in flash as the default application:

```
RedBoot> load -v -h 192.168.1.2 socket.srec
Entry point: 0x00000384, address range: 0x00000000-0x000718a8

RedBoot> fis create default
... Erase from 0x05030000-0x050b0000: ........
... Program from 0x00000000-0x000718a9 at 0x05030000: ........
... Erase from 0x053f0000-0x05400000: .
... Program from 0x007a0000-0x007b0000 at 0x053f0000: .
RedBoot>
```

If you wish to permanently disable the default application, use the "fis delete" command to delete it from the flash file-system (example shown in 6.2).

9 Updating

To update the bootloader: load and execute the "burn-redboot" program. Assuming the DeviceMaster board is connected to the Comtrol engineering network, the following command sequence will update the bootloader to the current engineering snapshot:

```
RedBoot> load -v -h 192.168.4.3 burn-redboot.srec
Entry point: 0x00000000, address range: 0x00000000-0x000228d8
/
RedBoot> go

Diag Startup
burn
Id=0089,8897
flash_erase_region(05010000,65536)
sector erase 05010000
sector erase 05020000
flash_program_buf(05010000,000028D8,65536)
ROM = 05000000
done -- resetting...
```

At this point the bootloader has been updated, and the baord should reset and run the new bootloader. The version number displayed by the Bootloader will be pretty much meaningless until the first official release. Until then, the build date is how you can tell if the bootloader has been updated.

10 Testing Serial Ports

There are two commands that run loopback (internal, RS-232, or RS-422) or port-to-port (RS-485) serial tests.

10.1 Loopback Tests

With a loop-back plug connected to a port, the “loop” command may be used to run a loop-back test in either internal loop-back mode, RS-232 mode or RS-422 mode. In RS-232 mode, modem control lines are also tested. In RS-422 mode and internal loop-back mode, only data paths are tested. If the test fails, a hexadecimal error code is displayed. The first digit of the error code indicates what portion of the test failed. Loop-back error codes are shown in Table 1.

Examples of loopback test commands are shown below.

RS-232 loopback test on port 2:

```
RedBoot> loop 232 2
Loopback pass
```

RS-232 loopback test on port 3:

```
RedBoot> loop 232 3
- Loopback failed RS-232: 10680
```

Internal loopback test on port 1:

```
RedBoot> loop int 1
Loopback pass
```

Table 1: Loopback Error Codes

Code	Description
`0x1ssss`	No data was present in the receive FIFO when there should have been. The chanel status register is displayed in the lower 4 digits (`ssss`).
`0x2ssss`	An error flag was present when receive data was read.
`0x3ttrr`	Receive data byte did not match transmit data byte. The transmit data byte is `tt` and the receive data byte is `rr`.
`0x40000`	Receive data was present after the expected last byte.
`0x5ssss`	CTS signal did not match expected value.
`0x6ssss`	RI signal did not match expected value.
`0x7ssss`	DSR signal did not match expected value.
`0x8ssss`	CD signal did not match expected value.

10.2 Port-to-Port RS-485 Test

Since RS-485 is a half-duplex physical layer, it is not possible to perform a loopback test. Instead an RS-485 cross-over cable must be connected between two different ports on the DeviceMaster.