Errata 1

Manual: 30504

Manual Index: 01

Table of Contents:

1. *Reference: Page 5-11, Chapter 5.4.5.1 ..... 1 - 3*
2. *Reference: Page 5-12, Chapter 5.4.5.1 ..... 1 - 3*
3. *Reference: Page 5-16, Chapter 5.7 ..... 1 - 3*
4. *Reference: Page 5-16, Chapter 5.7 ..... 1 - 6*

This page has been intentionally left blank.

1. Reference: Page 5-11, Chapter 5.4.5.1

The referenced information is revised as follows.

The sentence:

“Figure 5-1 on the next page illustrates this concept.”

in the first paragraph of this chapter is deleted.

2. Reference: Page 5-12, Chapter 5.4.5.1

The referenced information is revised as follows.

The Figure 5-1 and the remaining text on this page are deleted.

3. Reference: Page 5-16, Chapter 5.7

The referenced information is revised as follows.

The following NetBootLoader commands are added:

FDT

FUNCTION:	Enable creation of a flattened device tree (FDT)
SYNTAX:	**`fdt [<switch>]`** where: fdt — command <switch> — parameter: numeric string 0, 1 when set to 0 (zero) disables creation of an FDT (image is booted from 0x100) when set to 1 (one) enables creation of an FDT (image is booted from 0x0) when not specified, displays current setting
DESCRIPTION:	This command is used to specify the creation of an FDT for an operating system if required.
USAGE:	Create an FDT COMMAND / RESPONSE: **`NetBtLd> fdt 1`** **`Enabling creation of flattened device tree.`** **`Boot image from address 0.`** **`NetBtLd>`**
	Display the current FDT setting COMMAND / RESPONSE: **`NetBtLd> fdt`** **`Creation of flattened device tree is enabled.`** **`Image is booted from address 0.`** **`NetBtLd>`**

FDT_SHOW

FUNCTION:	Display the flattened device tree (FDT) information
SYNTAX:	**`fdt_show [<num>]`** where: fdt_show command <num> parameter: string 0, 1, 2, 3 ID number of the image for which FDT information is to be displayed no other values than those above are supported when not specified, displays FDT information for image 0
DESCRIPTION:	This command is used to have the NetBootLoader create and display the contents of the FDT for the image specified. The information provided can be used for system analysis or diagnostic purposes. Once the operating system is booted, this information is no longer available.

RUN

FUNCTION:	(Loads and) starts a bootable image
SYNTAX:	**`run [<num>]`** where: run command <num> parameter: string 0, 1, 2, 3 ID number of the image to be booted no other values than those above are supported if no image is specified, the image in the data buffer is booted
DESCRIPTION:	This command is used to (load and) start a boot image.

SPEED

FUNCTION:	Enables or disables fast booting
SYNTAX:	**`speed [<switch>]`** where: speed command <switch> parameter: numeric string 0, 1 when set to 0 (zero) disables “fast” boot mode when set to 1 (one) enables “fast” boot mode when not specified, displays current setting
DESCRIPTION:	This command is used to specify to the NetBootLoader that either the complete booting process is to be performed or that a shortened boot process is to be performed.
USAGE:	Enable “fast” booting COMMAND / RESPONSE: **`NetBtLd> speed 1`**
	Display the current setting COMMAND / RESPONSE: **`NetBtLd> speed`** **`Speed mode : "fast" boot up process.`**

4. Reference: Page 5-16, Chapter 5.7

The referenced information is revised as follows.

The following NetBootLoader command descriptions are modified, only the new version of each description is presented:

CBL

FUNCTION:	Set or display the parameters of the bootline function
SYNTAX:	**cbl [<num> (-|<bootline>)]** where: cbl command <num> parameter: string 0, 1, 2, 3, c ID number of the image to be associated with the bootline or bootline which is common to all images no other values than those above are supported - option: delete bootline for image specified or common <bootline> parameter: value: string (max. of 256 characters) <x … x> defines the bootline to be used with the kernel indicated by <num> or the common bootline
DESCRIPTION:	When an image is programmed to FLASH, it is assigned an ID number (0, 1, 2, or 3). This number is used to identify which image is to be addressed by the command CBL. In addition, a bootline common to all images may also be defined using the “c” parameter. If the command CBL is invoked without parameters, it returns the contents of all bootlines. Invoking the command CBL with the <bootline> parameter overwrites any previous bootline for the image specified.
USAGE:	Display the bootline for image 2 COMMAND / RESPONSE: **cbl** <contents of all bootlines are displayed>

DHCP

FUNCTION:	Interface to a DHCP or BOOTP server; exchange network configuration parameters
SYNTAX:	**dhcp [<timeout>]** where: dhcp command <timeout> parameter: value: numerical string <[n …]n> time, in seconds; must be >= 5 seconds
DESCRIPTION:	This command is used to set the network parameters for operation of the Ethernet port via either a DHCP or BOOTP server. Initially the EB8541 does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done either manually from the TERM interface using the command NET, or, if a DHCP or BOOTP server is available, it can be done automatically by the DHCP command. Manually configured parameters are permanently stored. Parameters configured using the DHCP command are temporary and will be lost if the system is reset or cold started. Prior to using the DHCP command, the IP address must be set to 255.255.255.255 with the NET command.
USAGE:	Program NetBootLoader (normal operation) COMMAND / RESPONSE: **NetBtLd> dhcp** **Sending request... reply from BOOTP/DHCP server.** **Network initializated ok.** **Server address is 192.168.112.2, our IP address is 192.168.112.14.** **Filename :** **NetBtLd>**

INFO

FUNCTION:	Display system information
SYNTAX:	**info**
DESCRIPTION:	The command INFO is used to display an information summary for the running system. Displayed are the following: CPU type, the board type, the size of the installed RAM and FLASH, and the areas occupied by the NetBootLoader and the programmed images. This information is displayed in hexadecimal offsets. Images programmed using the “-o” option of the command LF are not shown.
USAGE:	Display system information COMMAND / RESPONSE: **info** **CPU : Motorola MPC8541** **Board : EB8541** **Ram : 4000000** **Flash :** **Name : AMD 29LV640** **Bank : 0** **Bytelane : 0** **BankPortsize : 16** **ChipPortsize : 16** **Offset : 0x0** **Size : 0x800000** **NetBootLoader used FLASH: 0x0 - 0x60000** **Sector usage map:** **0x0000: nnnnnn00 00000000** **0x0010: 00000000 00000000** **0x0020: 00000000 00000000** **0x0030: 00000000 00000000** **0x0040: 00000000 00000000** **0x0050: 00000000 0011....** **0x0060: ....** **0x0070: ....** Where: n = NetBootLoader; 0 = image 0; 1 = image 1; . = usage unknown

LF

FUNCTION:	Load contents of the data buffer or area of visible memory to Flash
SYNTAX:	**`lf [<num>] [[-r]|[[-o[=]<offset>[ -k]]`** **`… [-m[=]<adr> -l[=]<len>]]]`** where: lf command <num> parameter: numeric string 0, 1, 2, 3 ID number assigned to this image -r option: delete image specified, or, if none specified, delete image 0 -o option: offset <offset>parameter: hexadecimal <[x …]x> program to FLASH offset of ... -k option: keep retain surrounding contents -m option: memory (address) <adr> parameter: hexadecimal <[x …]x> absolute address of image to be programmed -l option: length <len> parameter: hexadecimal <[x …]x> length of image to be programmed
DESCRIPTION:	If <num> is not specified, 0 is assumed. Without options, the FLASH is programmed using the contents of the data buffer. If no image is available in the data buffer, programming is terminated. If no offset option (“-o”) is specified the image is added along with the CRC and length information. Normally, the local data buffer holds the image to be programmed. However, if the “-m” and “-l” options are specified, the image is programmed from the absolute address specified.

LF

DESCRIPTION:	If the “-o” option is specified, the contents are programmed exactly at this offset in FLASH. No length and no CRC information is added. In addition, no image number is assigned (even if specified), and the image cannot be loaded and started with the RUN command. The “-k” option can be specified to prevent deletion of the surrounding FLASH contents. FLASH can only be erased sector-wise. If an image is programmed to a certain offset with the “-o” option, at least this sector (and maybe one or more of the following sectors depending on the size of the image) will be erased. The “-k” option can be used to retain the surrounding data, however, this slows down the operation significantly. To achieve fast programming of parameter images without destroying other FLASH contents, the data should be placed at a sector boundary and the sector(s) must not contain any other data or executable images. If organized this way, use of the “-k” option can be avoided. Note:The “lf” command cannot be used to program the NetBootLoader.
USAGE:	Program FLASH from data buffer and add CRC and image length (Image ID = 0 is assumed) COMMAND / RESPONSE (none): `lf`
	Program FLASH from visible address at 0x87000000 for length of 0x123456 COMMAND / RESPONSE (none): `lf -m=87000000 -l=123456`
	Program FLASH from data buffer to offset 0xF4240 and retain adjacent FLASH contents COMMAND / RESPONSE (none): `lf -o=f4240 -k`
	Delete image 1 from FLASH contents COMMAND / RESPONSE (none): `lf 1 -r`

NET

FUNCTION:	Set or display the parameters for the Ethernet interface
SYNTAX:	**net [<ip-addr>] [-netmask <netmask>]** **… [-gw <gateway>] [-num <num_net>] [-f]** where: net — command <ip-addr> — parameter: numerical string IP address of CPU board: nnn.nnn.nnn.nnn -netmask — option: netmask <netmask> — parameter: numerical string netmask of CPU board: nnn.nnn.nnn.nnn -gw — option: gateway <gateway> — parameter: numerical string gateway address for network: nnn.nnn.nnn.nnn -num — option: number <num_net> — parameter: numerical string 0, 1 ... logical identifier of Ethernet port addressed by this command -f — option: force CRC update
DESCRIPTION:	To set or display the parameters of the Ethernet interface the command “net” is used. Initially the CPU board does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done from the TERM interface using the command “net ... -f”. Using the “-f” option forces a CRC to be performed and stored along with the other configuration parameters in the serial EEPROM. Once the initialization of the Ethernet interface is done, the CPU board must be restarted for the parameters to take effect. Later changes to the parameters do not require the use of the “-f” option to force a CRC. This is done automatically. Only in the event that the Ethernet interface does not properly initialize, may it be necessary to re-enter the parameters using the “-f” option. If [-num <num_net>] is not specified, -num 0 is assumed.

SCRIPT

FUNCTION:	Provides very basic scripting capability
SYNTAX:	**`script [<newscript>|-]`** where: script — command <newscript> — parameter: string <[x …]x> "<newscript>" may only include simple commands; flow control constructs are not permitted; commands must be separated by semi-colons - — option: delete script contents
DESCRIPTION:	With the SCRIPT command, it is possible to control the boot process. During booting, if a valid script is available, the NetBootLoader will process it once the boot wait time is expired. If this command is issued without any parameters, the script contents are displayed.
USAGE:	Download a boot image from a TFTP server and run the boot image. COMMAND / RESPONSE (none): **`script dhcp; tftp; run`** Upon the next reset or cold start, after the boot wait time has expired the commands DHCP, TFTP, and RUN will be executed in that order. The above command sequence configures an Ethernet port, downloads the specified bootable image from an TFTP server, and then starts this image.

SF

FUNCTION:	Store FLASH contents or area of visible memory to data buffer
SYNTAX:	**sf (-o[=]<offset> -l[=]<length>)** | **…(-m[=]<add> -l[=]<length>)** | **…(<num>[-o[=]<offset>][-l[=]<length>])** where: sf command -o option: offset <offset> parameter: value: hexadecimal relative offset to start of FLASH contents or image to be stored to the data buffer -l option: length <length> parameter: value: hexadecimal length of FLASH contents or area of visible memory to be stored to the data buffer -m option: memory (address) <adr> parameter: hexadecimal <[x …]x> absolute address of image to be programmed <num> parameter: numerical string 0, 1, 2, 3 number of the image to be stored
DESCRIPTION:	With the command “sf” a selected portion of the FLASH contents or visible memory may be copied to the local data buffer, e.g. for a subsequent upload to the ftp server with the “put” command.
USAGE:	Store 64 kB of FLASH contents to the data buffer beginning at an offset of 1 MB COMMAND / RESPONSE (none): **sf -o=100000 -l=10000**
	Store FLASH image 2 to the data buffer COMMAND / RESPONSE (none): **sf 2**

SQ

FUNCTION:	Set or display the boot sequence
SYNTAX:	**sq [<num1><num2><num3><num4>]** where: sq command <num1> parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted <num2> parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted <num3> parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted <num4> parameter: numeric string: “0, 1, 2, 3” ID number of image to be booted
DESCRIPTION:	Up to four bootable images may be programmed into FLASH. The boot sequence defines to the NetBootLoader the order in which images are to be accessed when booting. The NetBootLoader starts with “<num1>” and continues until a valid image is found. In the case that a valid image is not found, the NetBootLoader stops searching and waits for operator intervention. All four number parameters must be defined even if there is not an image in the FLASH with that ID number. Any given ID number may only be used once: e.g. a sequence of 0120 is not permitted. The default sequence is 0123 if the boot sequence has not been programmed.
USAGE:	Display the current boot sequence setting. COMMAND / RESPONSE: **sq <cr>** **Bootsequence: 0 - 3 - 1 - 2**
	Set the boot sequence to 3201. COMMAND / RESPONSE(none): **sq 3201**

This page has been intentionally left blank.