APPLICATION NOTE

Using the existing CC9P-9360 modules with the new JSK DevBoard

---

March 2007

INTRODUCTION

With the new Embedded Linux (EL) v4.0 and v4.1, Digi started releasing CC9P modules with a new version of the U-Boot bootloader.

One of the major differences is the usage of the internal COM ports vs. the external COM ports.
The previous devboard, A9M9750DEV, was designed to use the COM ports via an additional chip, instead of the internal ones, part of the NS9360 and NS9750 chip.

With the new DevKit, Digi has changed this philosophy because most likely our customers will use the internal COM port, not an external additional chip.
This change reflects also in the bootloader.

Because of this change, if the user plugs the modules with previous releases of bootloader or Linux into the new JSK, the output that normally shows on COMA won't be visible.

It is then necessary to upgrade the module with the bootloader, kernel and rootfs shipped with EL4.x

NOTE: the new U-Boot v1.1.4 is using a total new set of commands; please refer to the specific documentation: "U-Boot Reference Manual", part of the distribution of EL4.x

INSTRUCTIONS

The U-Boot and Linux images to be downloaded into the module can be found under the directory "images" in the root directory of your EL installation.
i.e. /usr/loca/DigiEL-4.0/images

Specifically the images are:

U-Boot:	u-boot-cc9p9360js.bin
Kernel:	uImage-cc9p9360js
Rootfs:	rootfs-cc9p9360js-16.jffs2

There are 2 ways to update the bootloader on the module:

A) Using the JTAG Booster together with the adapter
B) using the A9M9750DEV Board

UPDATE USING THE JTAG BOOSTER

This method can be used only if the user has the following tools:

A) JTAG Booster (P/N FS285) - make sure is the version for 3.3V because the 5V version would damage the module
See Figure 1.

B) JT_ARM_0 Adapter for JSK (P/N FS360) - this allows the JTAG booster to be connected to the new JSK board
See Figure 2.

You will need a parallel port available in the PC to connect the JTAG Booster.

1) Plug the module into the new JSK DevBoard
2) Update the bootloader by following the instructions as described on the document: "JTAG_NS9xxxb.pdf", part of the installation CD
3) Boot, and stop the autoboot, so that only the U-Boot is running, not the application. The output should be something similar to Figure 4.
   Do not worry about the error messages, they appear because after the new U-Boot has installed the NVRAM content is no longer valid
4) Run the FLASH partition command: "flpart"
5) Verify there is no FLASH partition defined
6) Select Reset: "r"
7) Select type Linux: "l"
8) Now the correct partitions have been created
9) Select Quit: "q"
10) When asked if you want to Save, respond Yes: "y"
11) Update the kernel: "update linux tftp".
    The output should be something similar to Figure 5.
12) Update the rootfs: "update rootfs tftp".
    The output should be something similar to Figure 6.
13) Reboot via command: "reset" (or power cycle)
14) Now, after the boot, the module should have the correct images loaded and Linux can be started by typing: "dboot linux flash"

UPDATE USING THE A9M9750DEV BOARD

This method can be used if the user has a working kit with LxNETES 3.2 and the A9M9750DEV DevBoard.

NOTE: the described method below shows how to update the images via TFTP. If another method is chosen, the specific commands have to be used instead (i.e. update_uboot_usb vs. update_uboot_tftp).

1) Plug the module into the A9M9750DEV board
2) Boot, and stop the autoboot, so that only the U-Boot is running, not the application. The output should be something similar to Figure 3.
3) make sure you have the TFTP server up and running and all the environment variables set for your network
4) Run the command: “run update_uboot_tftp”
5) Make sure the update worked successfully
6) Reboot via command: “reset” (or power cycle)
7) Now the module is rebooted but you won’t see any output at all from the COM port
8) Power off the board
9) Plug the module into the new JSK DevBoard
10) Boot, and stop the autoboot, so that only the U-Boot is running, not the application. The output should be something similar to Figure 4.
    Do not worry about the error messages, they appear because after the new U-Boot has installed the NVRAM content is no longer valid
11) Run the FLASH partition command: “flpart”
12) Verify there is no FLASH partition defined
13) Select Reset: “r”
14) Select type Linux: “l”
15) Now the correct partitions have been created
16) Select Quit: “q”
17) When asked if you want to Save, respond Yes: “y”
18) Update the kernel: “update linux tftp”.
    The output should be something similar to Figure 5.
19) Update the rootfs: “update rootfs tftp”.
    The output should be something similar to Figure 6.
20) Reboot via command: “reset” (or power cycle)
21) Now, after the boot, the module should have the correct images loaded and Linux can be started by typing: “dboot linux flash”

FIGURE 1. FS285

FIGURE 2. FS360

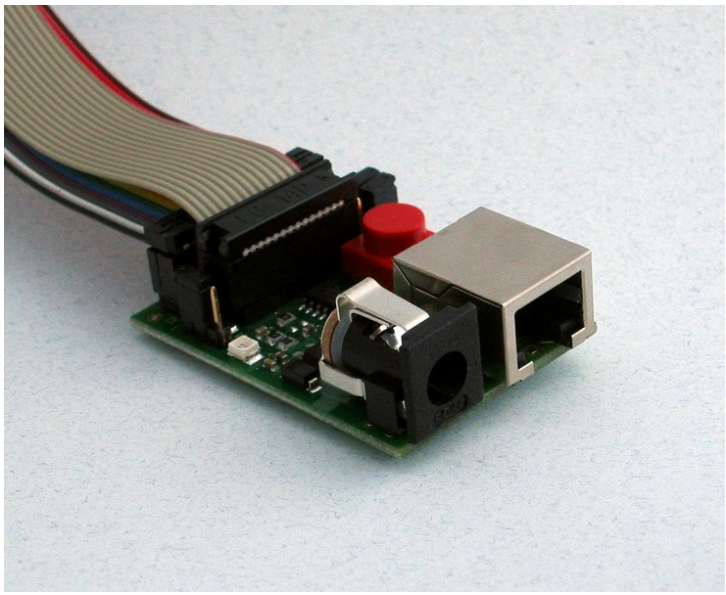

FIGURE 3.

```
U-Boot 1.1.3 (Jan 12 2006 - 14:06:18) FS.2
for FS Forth-Systeme CC9P9360 module on A9M9750DEV_1 board
CPLD Version: 2.1
FPGA Version: 2.1

U-Boot code: 00080000 -> 000A8714  BSS: -> 000EB200
RAM Configuration:
Bank #0: 00000000 64 MB
NAND:  64 MB
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  0
```

FIGURE 4.

```
U-Boot 1.1.4 (Mar 19 2007 - 16:34:50) DEL_4_1_BRC1
for Digi CC9P9360 on Development Board

DRAM:  64 MB
NAND:  64 MiB
NVRAM: Original Critical is BAD
NVRAM: Mirror Critical is BAD
NVRAM: Original Flash is BAD
NVRAM: Mirror Flash is BAD
NVRAM: No NVRAM contents usable
NVRAM: Resetting Workcopy
*** Warning - bad CRC or NAND, using default environment

In:    serial
Out:   serial
Err:   serial
CPU:   NS9360 @176.947200MHz
SPI ID:not a available or debug download
CC9P9360 #
```

FIGURE 5.

```
CC9P9360 # update linux tftp
TFTP  from  server  192.168.10.102;  our  IP  address  is
192.168.10.112
Filename 'uImage-cc9p9360js'.
Load address: 0x200000
Loading:
#################################################################
#################################################################
```

```
#################################################################
#################################################################
done
Bytes transferred = 1238472 (12e5c8 hex)
Calculated checksum = 0xe4d8a5e3
Erasing:   complete
Writing:   complete
Update successful
```

FIGURE 6.

```
CC9P9360 # update rootfs tftp
TFTP from server 192.168.10.102; our IP address is
192.168.10.112
Filename 'rootfs-cc9p9360js-16.jffs2'.
Load address: 0x200000
Loading:
#################################################################
#################################################################
#################################################################
#################################################################
#################################################################
#################################################################
#################################################################
#################################################################
#################################################################
#######################
done
Bytes transferred = 2845548 (2b6b6c hex)
Calculated checksum = 0x303c7d15
Erasing:   complete
Padding last sector
Writing:   complete
Writing cleanmarkers
Update successful
CC9P9360 #
```

[end of document]