

QNX BSP for phyCORE-i.MX35

by IBV - Echtzeit- und Embedded GmbH & Co. KG

Subject: Release Notes

1. Target System

- ◆ PHYTEC CPU Module phyCORE-i.MX35 (PCB# 1315.2):
 - ◆ Freescale i.MX357 applications processor
 - ◆ 32 MB NOR-Flash
 - ◆ 1024 MB NAND-Flash
 - ◆ 128 MB DDR2-RAM
- ◆ PHYTEC Mapper-Module PhyMAP-i.MX35 (PCB# 1318.2)
- ◆ PHYTEC Baseboard i.MX Carrier Board (PCB# 1280.4)
- ◆ Operating system QNX 6.4.1

2. Host Development System

- ◆ QNX Momentics 6.4.1
- ◆ Terminal emulation program (Qtalk, Momentics IDE Terminal, tip, HyperTerminal, etc.)
- ◆ RS-232 serial port or a USB-to-serial adapter, and a straight-through serial cable
- ◆ Ethernet link

3. System Layout

3.1. Memory Layout

Item	Address
OS image loaded at	0x80000000
NOR flash base address	0xA0000000

The interrupt vector table can be found in the buildfile located at `src/hardware/startup/boards/phycore-imx35/build`.

4. Getting Started

4.1. Starting Neutrino

- ◆ Step 1: Build the BSP

You can build a BSP OS image from the source code. For instructions about building a BSP OS image, please refer to the chapter "Working with a BSP" in the Building Embedded Systems manual.

Because of a limitation in the BSP import process you have to manually add the path to the "install/PLATFORM/sbin/" directory as a search path for "Other Files" to the QNX system builder project.

Copy or transfer the IFS image into your tftp server's directory.

- ◆ When compiling using the command line, the ifs image is in the "images/" directory.
- ◆ When compiling using the IDE, the IFS image is by default at "Workspace_root_dir/bsp-phycore-imx35/Images".

◆ Step 2: Connect your hardware

- ◆ Connect one end of the serial cable to the upper serial port (UART-DCE) on the carrier board.
- ◆ Connect the other end of the serial cable to the first available serial port of your host machine (e.g. COM1 on a Windows host).
- ◆ Connect an RJ-45 Ethernet cable to the X27 10/100 BaseT Ethernet RJ45 Connector on the carrier board.
- ◆ Connect the other end of the Ethernet cable to the Ethernet network where a TFTP server (which you'll use to transfer the boot image) exists.

On your host machine, start your favorite terminal program with these settings:

- ◆ Baud: 115200
- ◆ Bits: 8
- ◆ Stop bits: 1
- ◆ Parity: none

Then, apply power to the target. You should see output similar to the following:

```
U-Boot 2.0.0-rc8 (Jul 13 2009 - 16:30:26)

Board: PHYTEC phyCORE-i.MX35

mp11:      399000000 Hz
pp11:      300000000 Hz
arm:       399000000 Hz
gpt:       49875000 Hz
ahb:       99750000 Hz
ipg:       49875000 Hz
uart:      100000000 Hz
Malloc space: 0x87b00000 -> 0x87f00000 (size 4 MB)
Stack space : 0x87af8000 -> 0x87b00000 (size 32 kB)
running /env/bin/init...
```

```
Hit any key to stop autoboot: 1
```

Stop the boot process by pressing any key.

The U-Boot-v2 that comes with the board features a built-in editor called edit. Configure your environment by editing the /env/config file, adjusting IP addresses as necessary for your network. Changes can be saved using <ctrl>-<d>, or discarded using <ctrl>-<c>.

The first script to execute is /env/bin/init. At the end of that file, the boot script is called. Change the last line to call boot.qnx instead of boot.

The following content should be pasted to a new boot.qnx script:

```
#!/bin/sh
# Boot QNX image via TFTP
tftp bsp-phycore-imx35.ifs /dev/ram0; go 80000000
```

Changes to the environment configuration and script files have to be saved using the saveenv command.

With all changes in place, the board can be restarted using the reset command. The board should reboot, transferring the QNX OS image via TFTP from the host's file system. Eventually, booting should present output similar to the following:

```
Welcome to Neutrino 6.4.1 on the phyCORE-i.MX35x Board
Starting I2C driver...
Setting OS clock from on-board RTC...
Starting MX35 FEC Ethernet with TCP/IP stack (IP:192.168.X.Y) ...
Starting USB Host driver...
Starting Graphics driver...
Starting CAN driver...
#
```

At this point you're at the QNX command prompt where you can launch additional QNX programs.

4.2. Replacing U-Boot with QNX IPL

The QNX initial program loader (IPL) is a small and lightweight bootloader for the QNX operating system that consumes considerably less amount of flash memory.

The IPL for the phyCORE-i.MX35 board boots from NOR flash and assumes the following flash memory layout:

Address	Content
0xA0000000 - 0xA0008000	IPL
0xA0008000 - 0xA0020000	Usable area, e.g. configuration data, serial numbers, ...
0xA0020000 - 0xA0600000	QNX image (.ifs)
0xA0600000 - 0xA2000000	Usable area, e.g. FFS3 filesystem

The flash memory used on the phyCORE-i.MX35 board is a bottom configuration device where the first four blocks are only 32KB in size, while the remaining blocks are 128KB large. Because of a limitation of the QNX FFS3 library that's used by devf-generic to implement the flash file system and programming operations it isn't possible to erase the second, third and fourth block of the NOR flash (addresses 0xA0008000 to 0xA0020000) using QNX utilities. When erasing the first block using the QNX flashctl utility you'll be presented with a programming error, but the first sector will have been erased correctly.

Because of the above mentioned limitations this BSP leaves blocks two to four available for user specific purposes, and places the QNX image at the beginning of the first full-size block (offset 0x20000). The memory hole between the IPL and the QNX image makes a combined IPL and IFS image less useful, although the image can be used if no data is to be placed at addresses 0x8000 to 0x20000. The IPL will be padded to 128KB with the missing bytes filled with a 0xFF pattern in this case.

4.2.1. Erasing The Flash Memory

In order to erase the memory regions occupied by the IPL and the QNX image, the following commands can be used:

- ◆ Start the QNX flash driver
devf-generic -s0xA0000000,32M
- ◆ Unlock and erase 128KB starting at the beginning of the flash (for the IPL):
flashctl -p/dev/fs0 -o0K -l128K -vUe
- ◆ Unlock and erase 6MB (- 128KB) starting at 128KB (for the QNX image):
flashctl -p/dev/fs0 -o128K -l6016K -vUe

The first erase command is going to result in a protection error, because the flash driver will fail to unlock the second, third and fourth block. The second erase command therefore starts at offset 128K to erase the remaining full-sized blocks.

4.2.2. Writing IPL And IFS Binaries

Next, you'll need to have access to the IPL and IFS binaries. This can be achieved for example using a windows share and the fs-cifs client, or using a DOS formatted USB stick and the devb-umass driver. Consult the QNX documentation for details on how to use these utilities.

The IFS "ifs-phycore-imx35.bin" built from the default build file can be found in the "images/" directory of the BSP project. The IPL converted to a plain binary file can be found in the "images/" directory as well. When using the system builder in the QNX Momentics IDE an additional IFS "bsp-phycore-imx35.ifs" will be built at "Workspace_root_dir/bsp-phycore-imx35/Images".

- ◆ Copy the IPL to flash memory
dd if=/path/ipl-phycore-imx35.bin of=/dev/fs0 bs=1k
- ◆ Copy the QNX image to flash memory
dd if=/path/bsp-phycore-imx35.ifs of=/dev/fs0 bs=1k seek=128

Once the system is restarted, you'll be presented with the QNX IPL instead of the U-Boot-v2 bootloader that came preinstalled on the board:

```
QNX Neutrino IPL for phyCORE-i.MX35 board
```

```
Scanning flash at      0xA0020000
Found image at        0xA0020008
Jumping to startup at 0x800023E0
```

The IPL also offers a small command interface that can be used to download a QNX image using a serial connection and the QNX sendnto utility. In order to enter the command interface, the GPIO line GPIO2_23 has to be pulled low during boot. GPIO2_23 is available as signal x_EXP095 on the expansion connector of the phyCORE-i.MX35 carrier board.

4.3. Configuring Ethernet MAC Address

The MAC address of the i.MX35 built-in Ethernet controller can be configured using command line options or by reading it from flash memory.

In order to set the MAC address using command line options, the boot script needs to be modified. The boot script is either part of the build file (when building outside of the IDE via mkifs) or a separate file in case of a system builder project.

This BSP contains support for parsing the U-Boot v2 environment where the correct MAC address is already available. The startup code searches for the env file system at address 0xA0040000 (default for U-Boot v2 on the phyCORE-i.MX35) and parses the content of /config.

When using the QNX IPL a custom format for the MAC address and possibly other configuration information has to be implemented.

4.4. Driver Command Summary

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	startup-phycore-imx35	-	-	src/hardware/startup/boards/phycore-imx35
Serial	devc-sermx1 -e -F -c100000000 0x43F90000,45 0x43F94000,32	devc-sermx1	-	src/hardware/devc/sermx1
I2C	i2c-mx35 -l1 -u1 i2c-mx35 -l3 -u3	l2x-mc35	-	src/hardware/i2c/mx35
FEC Network	io-pkt-v4 -dimx35 mac=AABBCCDDEEFF -ptcpip	io-pkt-v4-hc ifconfig nicinfo ping ...	devn-mx35.so libsocket.so devnp-shim.so	src/hardware/devn/imx35
USB	io-usb -d ehci-mx31 ioport=0x53ff4100,irq=37 io-usb -d ehci-mx31 ioport=0x53ff4500,irq=35	io-usb usb	devu-ehci-mx31.so libusbdi.so class drivers	prebuilt binary only
NAND	fs-etfs-phycore_imx35 -m /fs/etfs	fs-etfs-phycore_imx35	-	src/hardware/etfs/nand2048/phycore-imx35
NOR	devf-generic -s0xA0000000,32M	devf-generic flashctl	-	src/hardware/flash/boards/generic
Graphics	io-display -dvid=0,did=0	io-display	devg-imx35.so libgf.so.1 libGLES_CM.so.1 libffb.so.2 libm.so.2 libOpenVG.so.1 libOpenVG-G12.so.1	src/hardware/devg/imx35
SD	devb-mmcsd-imx35 cam quiet	devb-mmcsd-imx35 mount umount	libcam.so cam-disk.so io-blk.so fs-dos.so fs-qnx4.so	src/hardware/devb/mmcsd
CAN	dev-can-mx35 -M can1	dev-can-mx35	-	src/hardware/can/mx35
RTC	i2c-mx35 -l1 -u1 rtc hw	rtc i2c-mx35 date	-	src/utils/r/rtc

Some of the drivers are commented out in the default buildfile. To use the drivers in the target hardware, you'll need to uncomment them in your buildfile, rebuild the image, and load the image into the board.

5. Known Issues for This BSP

- ◆ Card insertion and removal detection isn't implemented in the `devb-mmc-sd-imx35` driver. The SD card has to be inserted prior to starting the driver, and the card must not be removed while the driver is running.
- ◆ The serial driver `devc-sermx1` doesn't support hardware flow control.
- ◆ Because of a limitation of the QNX FFS3 library that's used by `devf-generic` to implement the flash file system and programming operations it isn't possible to erase the second, third and fourth block of the NOR flash (addresses `0xA0008000` to `0xA0020000`) using QNX utilities.
- ◆ When building an OS image via the QNX system builder you'll get an error about a missing file "imx35.conf". You have to manually add the path to the "install/PLATFORM/sbin/" directory as a search path for "Other Files" to the system builder project.

6. Sales / Technical Support

To get this BSP or to obtain technical support for the BSP, please contact:

IBV - Echtzeit- und Embedded GmbH & Co. KG

Keltenstr. 2

D-86343 Koenigsbrunn

Germany

Phone: +49 8231 9586-041

Fax: +49 8231 9586-049

Email: info@ibv-augsburg.de

Web: <http://www.ibv-augsburg.de/>

For more information about the PHYTEC i.MX35, please contact:

PHYTEC Messtechnik GmbH

Robert-Koch-Str. 39

D-55129 Mainz

Germany

Phone: +49 6131 9221-41

Fax: +49 6131 9221-33

Email: contact@phytec.de

Web: <http://www.pyhtec.de/>