Release Notes for the QNX Neutrino 6.4.1 BSP for Freescale i.MX53 EVK#

System requirements#

Target system

- QNX Neutrino RTOS 6.4.1

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

System Layout#

The tables below depict the memory layout for the image.

Memory layout

Item	Address
OS image loaded at	0x90100000

The interrupt vector table can be found in the buildfile located at **`src/hardware/startup/boards/mx53evk/build`**

Getting Started#

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.

- 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-freescale-mx53evk-src/images`**.

Step 2: Connect your hardware

1. Set up the i.mx53 EVK board in boot mode. Refer to the manual for the correct default jumper settings to use the CPU board.
2. Connect one end of the serial cable to the P5A1 (UART) serial port on the CPU board.
3. Connect the other end of the serial cable to the first available serial port of your host machine (e.g. ser1 on a Neutrino host).

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 2009.08 (Mar 29 2010 - 04:22:30)

CPU:   Freescale i.MX53 family 1.0V at 800 MHz
mx53 pll1: 800MHz
mx53 pll2: 600MHz
mx53 pll3: 216MHz
ipg clock     : 60000000Hz
ipg per clock : 50000000Hz
uart clock    : 21600000Hz
cspi clock    : 54000000Hz
ahb clock     : 120000000Hz
axi_a clock   : 300000000Hz
axi_b clock   : 200000000Hz
emi_slow clock: 200000000Hz
ddr clock     : 300000000Hz
esdhc1 clock  : 120000000Hz
esdhc2 clock  : 120000000Hz
esdhc3 clock  : 120000000Hz
esdhc4 clock  : 120000000Hz
Board: MX53 EVK 1.0 [POR]
Boot Device: SD
I2C:In:    serial
Out:   serial
Err:   serial
Net:   FEC0 [PRIME]
Hit any key to stop autoboot:
```

Step 3: Boot the IFS image

SD Card Initialization#

You will need a properly formatted SD card to boot the MX53. First, U-boot must be installed onto the card. To do this, you will need to download a Linux BSP image release from the Freescale website. As of this writing the most current file is L2.6.31_10.05.02_ER_images_MX5X.tar.gz. Once you have the BSP images extracted, change into the root directory of the BSP and execute the following command to get U-boot onto the SD card:

```
dd if=u-boot-mx53.bin of=/dev/X bs=512 && sync && sync
```

Replace the X with what your system assigns to the SD device. You can find this information with the dmesg command.

Note that the dd command will delete the partition table on the card. A filesystem is not required to boot, but if you would like one you scroll down to the later section and follow the instructions there.

Now, to get the raw OS image onto the SD card, navigate to your QNX bsp images directory, and execute the command

```
sudo dd if=ifs-mx53evk.raw of=/dev/X bs=512 seek=2048 && sync && sync
```

Again, replacing the 'X' with the proper value. Insert the SD card in the top MMCSD port on the target board and apply power to the board. After U-boot is started, you can change a few environment variables to enable automatic booting. At the U-boot prompt, execute the following commands:

```
set bootcmd_qnx 'mmc read 0 0x90100000 0x800 0x3800;go 0x90100000'
set bootcmd 'run bootcmd_qnx'
```

After these commands have been executed, use the **`print`** command to verify that they have been set properly and then the **`save`** command to save them to the memory card. Now, execute **`boot`** or reset the target board to attempt to boot into QNX.

If the image is successfully loaded U-Boot will display:

```
## Starting application at 0x90100000
```

You also should see QNX Neutrino boot, followed by the welcome message on your terminal screen:

```
CPU0: L1 Icache: 512x64
CPU0: L1 Dcache: 512x64 WB
CPU0: L2 Dcache: 4096x64 WB
CPU0: VFP 410330c2
CPU0: 412fc085: Cortex A8 rev 5 800MHz
fec_disable
init_hwinfo
mx51_init_i2c
mx51_aud_mux_init
mx51pdk_fec_init
mx51_usb_host_init
mc51_init_esdhcv2
mx51_cspi_mux_init
init_audio

System page at phys:70011000 user:fc404000 kern:fc404000
Starting next program at vfe044e74
cpu_startnext: cpu0 -> fe044e74
VFPv3: fpsid=410330c2
coproc_attach(10): replacing fe066c84 with fe066444
coproc_attach(11): replacing fe066c84 with fe066444
Welcome to QNX Neutrino 6.4.1 on the i.MX53 EVK (ARM Cortex-A8 core) Board
Starting on-board ethernet with TCP/IP stack...
Getting network address with DHCP...
fec0: flags=843<UP,BROADCAST,RUNNING,SIMPLEX> mtu 1500
        address: XX:XX:XX:XX:XX:XX
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet xxx.xxx.xxx.xxx netmask 0xfffff000 broadcast xxx.xxx.xxx.xxx
#
```

You can test the OS simply by executing any shell builtin command or any command residing within the OS image (e.g. **`ls`**).

Getting a filesystem onto the SD Card#

You can use fdisk to put a filesystem on your SD card as follows:

```
fdisk -l
```

Read through the output until you find the section on the connected SD card, which should look something like this:

```
Disk /dev/sdd: 1967 MB, 1967128576 bytes
61 heads, 62 sectors/track, 1015 cylinders
Units = cylinders of 3782 * 512 = 1936384 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdd doesn't contain a valid partition table
```

The line we are interested in is

```
Units = cylinders of 3782 * 512 = 1936384 bytes
```

We will use the number of cylinders (in this case, 3782) to calculate the start cylinder of the filesystem. Note that since U-boot and the QNX image are stored in the first 4 megabytes, the filesystem must start after that. First, divide the number of cylinders into 8192 (8192/3782 = 2.166049...). Then round that number up to the nearest whole number (2.166049 rounded up = 3). Last, add 1 (3 + 1 = 4). So in this case, the start cylinder for the filesystem will be cylinder #4.

Now we can use fdisk to create a new partition. Do the following commands

```
fdisk /dev/sdd

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x1a249dd3.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help): n
Command action\\
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1015, default 1): 4
```

Note that the '4' entered here is the number we calculated earlier. Substitute in your own calculated value.

```
Last cylinder, +cylinders or +size{K,M,G} (4-1015, default 1015):
```

Hit enter here to use the default value.

```
Using default value 1015
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): c
Changed system type of partition 1 to c (W95 FAT32 (LBA))

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
```

Done now with fdisk, just one last command:

```
mkfs.vfat /dev/sdd1
```

And your filesystem is created.

Step 4: Boot the IFS image using QNX IPL

Booting using the QNX IPL#

After building the BSP Package, you will find the elf32 executable "ipl-mx53evk" in the folder src/hardware/ipl/boards/mx53evk/arm/le/

Change the directory, to the IPL folder using the following command

```
cd src/hardware/ipl/
```

Convert the elf32 executable to the binary format using the following command

```
ntoarm-objcopy --output-format=binary boards/mx53evk/arm/le/ipl-mx53evk boards/mx53evk/arm/le/ipl-mx53evk.bin
```

Place the SD Card, in which you have to burn the IPL, into the SD Slot of the Linux HOST.

Now zero fill the first 10MB of the SD Card (you can fill lesser size also) using the following command

```
sudo dd if=/dev/zero of=/dev/mmcblk0 bs=1M count=10
```

Format the SD card using the following command

```
sudo fdisk /dev/mmcblk0
```

After giving the format command to the SD card, give the following Command Options

```
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xa17f8c2d.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help): p

Disk /dev/mmcblk0: 1977 MB, 1977614336 bytes
4 heads, 16 sectors/track, 60352 cylinders
Units = cylinders of 64 * 512 = 32768 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa17f8c2d

      Device Boot      Start         End      Blocks   Id  System

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
```

```
p
Partition number (1-4): 1
First cylinder (1-60352, default 1): 100
Last cylinder, +cylinders or +size{K,M,G} (100-60352, default 60352): 60352

Command (m for help): a
Partition number (1-4): 1

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): c
Changed system type of partition 1 to c (W95 FAT32 (LBA))

Command (m for help): p

Disk /dev/mmcblk0: 1977 MB, 1977614336 bytes
4 heads, 16 sectors/track, 60352 cylinders
Units = cylinders of 64 * 512 = 32768 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa17f8c2d

      Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1   *         100       60352     1928096    c  W95 FAT32 (LBA)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
```

Create a FAT32 filesystem, on the formatted SD Card, using the following command

```
sudo mkfs.vfat -F 32 /dev/mmcblk0p1
```

Now Burn the IPL, to the raw partition of the formatted SD Card, using the following command

```
sudo dd if=boards/mx53evk/arm/le/ipl-mx53evk.bin of=/dev/mmcblk0 skip=2 seek=2 bs=512 && sync && sync
```

Copy the images/ifs-mx53evk.raw, which was built in the BSP package, to the SD card's Mounted partition (mmcblk0p1) and rename it to "QNX-IFS" (As the IPL needs the file to be named as "QNX-IFS")

Now place the SD card, in to the SD/MMC slot of the iMX53 target and apply power to the board. The target runs the IPL as below and waits for the command 'M' to boot the QNX IFS image. After giving the 'M' command, it boots the QNX IFS image as below:

```
Welcome to QNX Neutrino Initial Program Loader for Freescale iMX53 Board
Command:
Press 'M' for SDMMC download, file QNX-IFS assumed.
Load QNX image from SDMMC...
Found image             @ 0x80800008
Jumping to startup      @ 0x90100660

CPU0: L1 Icache: 512x64
CPU0: L1 Dcache: 512x64 WB
```

```
CPU0: L2 Dcache: 4096x64 WB
CPU0: VFP 410330c2
CPU0: 412fc085: Cortex A8 rev 5 800MHz

System page at phys:70011000 user:fc404000 kern:fc404000
Starting next program at vfe044e74
cpu_startnext: cpu0 -> fe044e74
VFPv3: fpsid=410330c2
coproc_attach(10): replacing fe066c84 with fe066444
coproc_attach(11): replacing fe066c84 with fe066444
Welcome to QNX Neutrino 6.4.1 on the i.MX53 EVK (ARM Cortex-A8 core) Board
Starting on-board ethernet with TCP/IP stack...
Getting network address with DHCP...
fec0: flags=843<UP,BROADCAST,RUNNING,SIMPLEX> mtu 1500
        address: 00:04:9f:cc:9a:b5
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet 10.90.74.70 netmask 0xffffff00 broadcast 10.90.74.255
Starting SD1 memory card driver...
Starting SD3 memory card driver...
# Path=0 - FREESCALE MX35 e
 target=0 lun=0     Direct-Access(0) - SD:3 SD02G Rev: 8.0
Path=0 - FREESCALE MX35 e

#
```

Driver Command Summary#

The following table summarizes the commands to launch the various drivers.

Component	Buildfile Command	Required Binaries	Required Libraries	Source Location
Startup	`startup-mx53evk -d1 -v`	.	.	`src/hardware/startup/boards/mx53evk`
Serial	`devc-sermx1 -e -F`	devc-sermx1	.	`src/hardware/devc/sermx1`
FEC Network	`io-pkt-v4 -d mx53 mac=xxxxxxxxxxxx`	io-pkt-v4 ifconfig if_up dhcp.client	devnp-mx53.so libsocket.so	`src/hardware/devnp/mx53`
SD card	`devb-mmcsd-mx53 mmcsd ioport=0x50004000,irq=1` `devb-mmcsd-mx53 mmcsd ioport=0x50020000,irq=3`	devb-mmcsd-mx53	libcam.so fs-dos.so cam-disk.so	`src/hardware/devb/mmcsd`
I2C	`i2c-mx35 -u0,0` `i2c-mx35 -u1,1`	i2c-mx35	.	`src/hardware/i2c/mx35`
SPI	`spi-master -u1 -d mx51ecspi base=0x50010000,irq=36,waitstate=0,clock=54000000,burst,gpiocsba`	spi-master	spi-mx51ecspi.so	`src/hardware/spi/mx51ecspi`

USB	`io-usb -d ehci-mx31 ioport=0x53F80300,irq=14`	io-usb usb devb-umass	devu-ehci-mx31.so libusbdi.so	`binary`
CAN	`dev-can-mx35 can0 can1`	dev-can-mx35 canctl*	.	`src/hardware/can/mx35`
Graphics	`io-display -dvid=0,did=0`	io-display imx53.conf	devg-imx51.so devg-soft3d.so libgf.so.1 libGLES_CM.so.1 libffb.so.2	`src/hardware/devg/imx51`
WDT	`wdtkick`	wdtkick	.	`src/hardware/support/wdtkick`
SPI DATAFLASH	`fs-etfs-imx53 -e fs-etfs-imx53 -m /fs/etfs`	fs-etfs-imx53 etfsctl		`src/hardware/etfs/nand512/`
NAND FLASH	`fs-etfs-mx53evk -e fs-etfs-mx53evk -m /nand`	fs-etfs-mx53evk etfsctl		`src/hardware/etfs/nand2048/`
Audio	`io-audio -d mx-mx53evk ssibase=0x50014000,rxevt=25,txevt=24,srchn=4,rate=48000,mixe`	io-audio wave wave,ute mix_ctl	deva-ctrl-mx-mx53evk.so deva-util-mixer.so libasound.so.2 libdma-sdma-imx53.so	`src/hardware/deva/ctrl/mx`
DMA			libdma-sdma-imx53.so	`src/lib/dma/sdma`

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.

Startup, Ethernet, Dataflash, Can, Audio, NAND Flash and Graphics have additional details:

Startup#

`startup-mx53evk [`*`startup-options`*`]`

Some modules aren't enabled after boot up, so you need to use command line options to startup to enable them. These options must be passed before any other startup options.

To enable	Use this option	Comment
WDT	-W	Enable WatchDogTimer

Ethernet#

`io-pkt-v4-hc -d mx53 mac=xxxxxxxxxxxx`

MAC address labelled on i.mx53 evk board must be provided with "mac" command line option.

Dataflash#

Before starting the dataflash driver , spi driver must be started.

To start spi driver run:

```
spi-master -u1 -d mx51ecspi
base=0x50010000,irq=36,waitstate=0,clock=54000000,burst,gpiocsbase=0x53f8c000,g
```

To erase dataflash run :

```
fs-etfs-imx53 -e
```

To mount etfs filesystem on a mount point run:

```
fs-etfs-imx53 -m /fs/etfs
```

CAN#

There are two CAN interfaces on the i.MX53 EVK board: CAN0 and CAN1 connected to expansion port.

- dev-can-mx35 can0

or can1 if can1 is used or both if both are used.

Graphics#

With the current revisions of the iMX53 EVK (rev B) the VGA output is not supported due to a silicon errata. (ENGcm11165 TVE: TV DAC is not functional - TV DAC is also used for VGA) To run graphics the 'MCIMX51LCD - Hardware Only - i.MX51 WVGA LCD DAUGHTER' from Freescale is required. (see Freescale website for details)

To run the Graphics Driver, please issue the command: `io-display -dvid=0,did=0`

The `display.conf` config file, maps the driver config file imx53.conf, to get the output display timing and configuration settings.

Accelerated Graphics#

An archive containing an experimental driver for the GPU (OpenVG & OpenGLES) can be found on the Foundry 27 graphics project:

http://community.qnx.com/sf/go/projects.graphics/frs.imx51_gpu_driver

Please obtain the latest version and follow the corresponding setup instructions.

Audio#

Before starting Audio driver i2c driver must be started.

To start i2c driver run:

```
i2c-mx35 --u1,1
```

```
i2c-mx35 --u0,0
```

To start audio driver run:

```
io-audio -d mx-mx53evk
ssibase=0x50014000,tevt=25,tchn=3,revt=24,rchn=4,rate=48000,mixer=i2cdev=1:adr0
```

NAND Flash#

Run

```
fs-etfs-mx53evk -e
```

to erase the NAND device and to create an empty file system that is ready to use. The factory marked bad blocks are not erased. Blocks that become bad during normal use are also skipped during the erasing.

After the erase of the NAND device, run the following command

```
fs-etfs-mx53evk -m /nand
```

This command sets the directory /nand as the mount point.

Known Issues for This BSP#

- Support for most features is either limited or disabled at this time. These limited features include:
  - Video
- To be determined if these issues remain
  - The audio driver `deva-ctrl-mx-mx53evk.so` doesn't support Syncronous data playback ability since the DMA library doesn't have position information. (Ref# 75488)
  - The i.MX35 (ARM1136) processor doesn't support unaligned accesses in hardware. If an application (e.g. pwmopts) tries to access data that isn't aligned on a 32-bit boundary, a bus error will occur. To avoid this memory fault, you can enable the software emulation of unaligned accesses by starting procnto with the -ae option. (Ref# 71252)