

# Operating Systems for Embedded Systems

## Preparing a minimal Linux system for i.MX53QSB

Massimo Violante, Politecnico di Torino, Dip. Automatica e Informatica  
massimo.violante@polito.it

### Purpose

The purpose of this document is to describe how to create a minimal Linux system for the i.MX53QSB to be used in the lab for the course “Operating System for Embedded System”.

The content of this document is coming in part from the following page, authored by Robert Nelson: <http://eewiki.net/display/linuxonarm/i.MX53+Quick+Start>

### Prerequisites

In order to compile U-Boot, and the Linux Kernel the following items are needed: ARM cross compiler, ncurses, mkimage, U-Boot, the Linux kernel, and a root file system:

```
sudo apt-get install gcc-arm-linux-gnueabi
sudo apt-get install uboot-mkimage
sudo apt-get install libncurses5-dev
sudo apt-get install git
git clone git://git.denx.de/u-boot.git
git clone git://github.com/RobertCNelson/imx-devel.git
wget -c http://rcn-ee.net/deb/minfs/squeeze/debian-6.0.6-minimal-armel-2012-10-22.tar.xz
wget https://raw.githubusercontent.com/RobertCNelson/Bootloader-Builder/master/patches/v2012.10/0001-enable-bootz-support.patch
wget https://raw.githubusercontent.com/RobertCNelson/Bootloader-Builder/master/patches/v2012.10/0002-imx-convert-to-uEnv.txt-n-fixes.patch
```

### Compile U-Boot

Move in the U-Boot directory, create a new tree:

```
cd u-boot/
git checkout v2012.10 -b tmp
```

Apply the patches and compile U-Boot:

```
patch -p1 < 0001-enable-bootz-support.patch
patch -p1 < 0002-imx-convert-to-uEnv.txt-n-fixes.patch
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- distclean
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- mx53loco_config
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
```

### Compile Linux

Move in the Linux directory, create a new tree:

```
cd imx-devel
git checkout origin/imx-bsp -b imx-bsp
./build_kernel.sh
```

Upon completion of the kernel compilation the kernel (2.6.35.3-imx3.6.zImage) and modules (2.6.35.3-imx3.6-modules.tar.gz) will be under the imx-devel/deploy.

## Prepare the root file system

First, decompress the downloaded file system:

```
tar xJf debian-6.0.6-minimal-armel-2012-10-22.tar.xz
```

## Set-up the SDCARD

Let us suppose that the SDCARD is associated to the `/dev/sdb` Linux device. We need to start from a blank SDCARD, so incase it comes pre-formatted, remove any partition from it using `fdisk`.

After removing any partition from the SDCARD, we have to copy U-Boot first:

```
sudo dd if=./u-boot/u-boot.imx of=/dev/sdb seek=1 bs=1024
```

This operation will writing the boot loader in the first sector of the SDCARD, and thus when inserted in the i.MX53QSB it will be loaded and executed.

We then have to prepare two Linux primary partitions in the remaining space of the SDCARD:

- `/dev/sdb1`, 10 Megabytes, it will host the Linux kernel and the bootstrap script for U-Boot.
- `/dev/sdb2`, using the remaining free space, it will host the Linux root file system.

For setting-up the SDCARD, `fdisk` is highly recommended.

Upon completion of the SDCARD set-up, we need to format and mount the two partitions:

```
sudo mkfs.ext2 /dev/sdb1
sudo mkfs.ext4 /dev/sdb2
mount /dev/sdb1 /media/boot
mount /dev/sdb2 /media/rootfs
```

## Set-up the boot partition

Copy the Linux kernel in the boot partition

```
sudo cp -v ./imx-devel/deploy/2.6.35.3-imx3.6.zImage /media/boot/zImage
```

Then, we need to prepare the bootstrap script that must be placed in `/media/boot/uEnv.txt`, and must have the following content:

```

initrd_high=0xffffffff
fdt_high=0xffffffff

console=ttymx0,115200

mmcroot=/dev/mmcblk0p2 ro
mmcrootfstype=ext4 rootwait fixrtc

optargs=console=tty0
video=mxcdi1fb:1280x720M@60

mmc_load_image=ext2load mmc 0:1 0x70010000 zImage
mmc_load_initrd=ext2load mmc 0:1 0x72000000 initrd.img; setenv initrd_size
${filesize}
mmc_load_dtb=ext2load mmc 0:1 0x71ff0000 /dtbs/${dtb_file}

mmccargs=setenv bootargs console=${console} ${optargs} root=${mmcroot}
rootfstype=${mmcrootfstype} ${video}

#Just: zImage
xyz_mmcboot=run mmc_load_image; echo Booting from mmc ...
loaduimage=run xyz_mmcboot; run mmccargs; bootz 0x70010000

#zImage and initrd
#xyz_mmcboot=run mmc_load_image; run mmc_load_initrd; echo Booting from mmc ...
#loaduimage=run xyz_mmcboot; run mmccargs; bootz 0x70010000
0x72000000:${initrd_size}

```

This is read by U-Boot and specifies how to boot the Linux kernel, and where the root file system is located.

## Set-up the rootfs partition

First populate the root file system:

```
sudo tar xfv ./-*-minimal-arm*/armel-rootfs-*.tar -C /media/rootfs/
```

Then, copy Kernel modules:

```
sudo tar xfv ./imx-devel/deploy/3.6.6-imx2-modules.tar.gz -C /media/rootfs/
```

A few configuration files need to be modified in order to complete the configuration of the Linux system.

Edit `/media/rootfs/etc/fstab` and modify it as follows:

```

/dev/mmcblk0p2 /          auto    errors=remount-ro 0 1
/dev/mmcblk0p1 /boot/uboot auto    defaults           0 2

```

To enable the Ethernet connection, edit `/media/rootfs/etc/network/interfaces` and uncomment the `eth0` configuration, so that the content of the file contains the following lines:

```

auto eth0
iface eth0 inet dhcp

```

Finally, the serial console must be enabled by editing `/media/rootfs/etc/inittab`, and modifying its contents so that it contains the following line:

```
T2:23:respawn:/sbin/getty -L ttymx0 115200 vt102
```

## **Finalizing the work**

We can now sync the files on the SDCARD, unmounts the volumes and place it in the i:MX53QSB.

```
sync
sudo umount /media/boot
sudo umount /media/roofs
```

## **Bootstrapping**

Run your preferred terminal emulation, e.g., `minicom`, and configure it for a connection with the following parameters: 115200, 8n1.

Run it, power on the i.MX53QSB and look at the bootstrap (root password is root).