

Establish Development Environment
HL
S2014

1. Create a director /opt/FriendlyARM/mini6410/linux by using

sudo mkdir command, all the tar files will be installed from issuing commands from within this directory later.

2. Create a temporary /tmp/linux and copy all the tar files from the linux directory of your distribution CD.
3. Build cross compiler (arm-linux-gcc-4.5.1). First, make sure you have already copied arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz to the directory tmp/ then

```
#cd /tmp  
#tar xvzf arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz -C /
```

Note C for changing directory and is upper case and followed by a space before / . As result this will install arm-linux-gcc to /opt/FriendlyARM/toolschain/4.5.1

Before you can use it, change .bashrc file as follows:

```
#gedit /root/.bashrc
```

add last line to this file as

```
PATH=$PATH: /opt/FriendlyARM/toolschain/4.5.1/bin
```

then update by

```
#source ~/.bashrc
```

You can check by

```
#arm-linux-gcc -V
```

It will return the version info for you, so it is installed good.

4. Installation of the rest tar files:

(1) U-boot; at the directory /opt/FriendlyARM/mini6410/linux do the following:

```
#tar xvzf /tmp/linux/ u-boot-mini6410-20101106.tar.gz
```

which will produce u-boot-mini6410 directory with U-boot source coed.

(2) linux

```
#tar xvzf /tmp/linux/ linux-2.6.38-20110325.tar.gz
```

which will produce linux-2.6.38 directory with all source distribution code.

(3) File system

```
#tar xvzf /tmp/linux/ rootfs_qtopia_qt4-20101120.tgz
```

which generate directory rootfs_qtopia_qt4

(4) qtopia (for embedded gui and graphics applications etc.)

```
#tar xvzf /tmp/linux/ x86-qtopia-20100420.tar.gz
```

```
#tar xvzf /tmp/linux/ arm-qtopia-20101105.tar.gz
```

which generates two directories x86-qtopia and arm-qtopia.

(5) qt-extended-4.4.3 (for graphics)

```
#tar xvzf /tmp/linux/ x86-qt-extended-4.4.3-20101003.tgz
```

```
#tar xvzf /tmp/linux/ arm-qt-extended-4.4.3-20101105.tgz
```

which forms two directories x86-qt-extended-4.4.3 and arm-qt-extended-4.4.3

(6) QtE-4.7.0

```
#tar xvzf /tmp/linux/x86-qte-4.6.1-20100516.tar.gz
```

```
#tar xvzf /tmp/linux/ arm-qte-4.7.0-20101105.tar.gz
```

which produces x86-qte-4.6.1 and arm-qte-4.7.0 two directories. x86-qte-4.6.1 is for Creator development platform.

(7) busybox (busybox version 1.13.3, you can get the newest version from (<http://www.busybox.net>))

```
#tar xvzf /tmp/linux/ busybox-1.17.2-20101120.tgz
```

which produces busybox-1.17.2 directory.

(8) Examples

```
#tar xvzf /tmp/linux/ examples-mini6410-20101110.tgz
```

which produces examples directory.

(9) rootfs_qtopia_qt4-20101120.tgz and rootfs_qtopia_qt4-s-20101120.tgz

where '-s' for professional touch screen color LCD (4 wires), the one without 's' is for ARM default touch screen (1 wire). The difference between two rootfs is “/etc/friendlyarm-ts-input.conf” has different configurations.

```
#tar xvzf /tmp/linux/ rootfs_qtopia_qt4-20101120.tgz
# tar xvzf /tmp/linux/ rootfs_qtopia_qt4-s-20101120.tgz
```

which forms two directories rootfs_qtopia_qt4 and rootfs_qtopia_qt4-s. Note qtopia-2.2.0, Qtopia4 and QtE-4.7.0, busybox, support NFS, auto detect if touch screen display is attached, auto detect if SD card (up to 32G) is high speed version, auto detect USB mouse.

(10) (optional, file system tool) mktools: mkyaffs2image will make file system image (when linux kernel starts up, depends on the line command, the kernel will start with yaffs2, ubifs, ext2, nfs etc.) for different Nand Flash file (size of block and page), for example:

(10.1) mkyaffs2image :

For “1 Page= 512 Byte,1 Block=16K” SLC Nand Flash, yaffs2, (hardware part: K9F1208).

(10.2) mkyaffs2image-128M:

For “1 Page= 2K Byte,1 Block=128K” SLC Nand Flash, yaffs2, hardware : K9F2G08,K9F4G08,K9K8G08. (in this distribution 10.1 and 10.2 both shares the same command mkyaffs2image-128M).

(10.3) mkubimage-slc

For “1 Page= 2K Byte,1 Block=128K” SLC Nand Flash, UBIFS, hardware K9F2G08,K9F4G08,K9K8G08.

(10.4) mkubimage-mlc2

For “1 Page= 8K Byte,1 Block=1M” MLC Nand Flash, UBIFS, hardware K9GAG08U0E.

(10.5) mkext3image

For individual EXT3 to use FAT32/FAT format for SD Card installation of Linux OS.

```
#tar xvzf /tmp/linux/mktools.tar.gz -C /
```

which builds tools under directory /usr/sbin

(11) (Optional) LogoMaker

LogoMaker is for making Linux Logo, which can convert bmp, jpg, png file converted to Linux Logo.

```
#tar xvzf /tmp/linux/logomaker.tgz -C /
```

LogoMaker then is installed in the directory /usr/sbin once this is done use command logomaker to test it out, you should see a window with a logo to appear.

(12) (optional) U-boot

(12.1) For NAND use U-boot

```
#cd /opt/FriendlyARM/mini6410/linux/u-boot-mini6410
#make mini6410_nand_config-ram128
#make
```

then down load to SD Card or USB to use it.

Distribution CD has provided already built Uboot at the directory: images/linux/u-boot_nand-ram128.bin

For 256M U-boot:

```
#cd /opt/FriendlyARM/mini6410/linux/u-boot-mini6410
#make mini6410_nand_config-ram256
#make
```

which will build the U-boot, the distribution CD has already provided the already built CD at images/linux/u-boot_nand-ram256.bin

(12.2) For SD card U-boot

For different sizes of DDR RAM build U-boot accordingly.

For 128M DDR RAM:

```
#cd /opt/FriendlyARM/mini6410/linux/u-boot-mini6410
#make mini6410_sd_config-ram128
#make
```

(12.3) (Optional) SD-Flasher.exe

(13) Kernel

(13.1) Config and build

First for different LCD use one of the following configuration file from directory /opt/FriendlyARM/mini6410/linux/linux-2.6.38

```
config_mini6410_x35 –For Sony 3.5”LCD, 240x320
config_mini6410_n43 –For NEC4.3”LCD, 480x272
config_mini6410_l80 – For Sharp 8”LCD, 640x480
config_mini6410_a70 –For 7” display, 800x480
config_mini6410_vga1024x768 –for 1024x768 VGA module board
config_mini6410_vga800x600 – for 800x600 VGA module board
config_mini6410_vga640x480 – for 640x480 VGA module board
config_mini6410_ezvga800x600 – for simple VGA module board, 800x600
```

Example: use config_n43 for kernel compile and build, make sure you are at directory /opt/FriendlyARM/mini6410/linux/linux-2.6.38 then

```
#cp config_mini6410_n43 .config
#make zImage (or use make)
```

Reference: 1. <http://www.friendlyarm.net/> 2. there are few commands may be useful while you need them (ldd to print sheared library dependencies; echo \$PATH to find our ./.bashrc path settings; rm -Rf unwanted_directory just in case you need it).

(END)