

Compiling and Running Programs With Linux and Qtopia 2.2.0

Industrial ARMWorks www.andahammer.com www.frindlyarm.us

Literal translations of the original Chinese user manual are fraught with imminent peril! In this text everything has been tested and corrected. We assume you are using the Linux 2.6.29 kernel and Qtopia 2.2.0 that ships on the Mini2440. This is tested with Debian system on the host PC. Under Redhat or Fedora some directories might be automatically created during uncompression of files. Making them ahead of time won't hurt anything.

Setting up the gcc cross compiler.

The current version (Aug 1009) of the FriendlyARM software uses the arm-linux-gcc-4.3.2 compiler for everything from the boot loader to Qtopia. The compressed compiler is in the /linux directory of the DVD. Copy it to a convenient place like /tmp. You will need to be the super user to do this. Then untar the

```
$ su [root password]
# cd /tmp
# tar xvzf arm-linux-gcc-4.3.2.tgz -C/ (with a capital 'C')
```

If this fails, make the directories first.

```
# cd ..
# mkdir /usr/local/arm
# mkdir /user/local/arm/4.3.2
```

and try again.

Now edit /root/bashrc and add this line:

```
PATH=$PATH:/usr/local/arm/4.3.2/bin
```

This adds the path for all users. You can also do this for a single user by editing the file in the home directory or use a directive in a config file. For this new path to be seen, you will have to log out and back in. Check to see if everything is copasetic.

```
# arm-linux-gcc -v
Reading..... tells you all about the compiler.
Thread model: posix
Gcc version 4.3.1
#
```

Hello World (There has to be something better to say!)

First, set a the directory for the example code in `/opt/FriendlyARM/mini2440`.

```
# cd opt
/opt# mkdir FriendlyARM
/opt# mkdir /FriendlyARM/mini2440
/opt# mkdir /FriendlyARM/mini2440/examples
/opt# cd ..
#
```

Copy the examples from the `/linux/examples.tgz` DVD directory to `/tmp` and uncompress as with the compiler. Check to see if the hello example and others are in the examples dir.

The Hello World source code is very simple.

```
#include<stdio.h>
int main(void){ printf("Good news everyone!\n"); }
```

To compile, just enter the directory and run Make.

```
# cd /opt/FriendlyARM/mini2440/examples/hello
/opt/FriendlyARM/mini2440/examples/hello# make
```

Or invoke the compiler from the command line.

```
/opt/FriendlyARM/mini2440/examples/hello# arm-linux-gcc-
4.3.2 -o hello main.c
```

Either way, a `hello.bin` file will be created (we hope). Now to get it into your Mini2440 and execute it. There are several alternatives.

FTP The kernel and file system as shipped has only the root user and no password. FTP wants a password! If you have a Mini2440 +LCD, open a terminal and stylus keyboard. If not, you will need to be connected by serial cable to a com program. 115200 8,1, none.

Since there is no password, use the `passwd` command with no arguments and follow the prompt.

```
~# passwd
```

If you need a user name, it is root. Choose a password with a few letters and numbers.

Connect to your PC with the Cat5 Ethernet crossover cable and open a terminal window on your host system (the PC). You will need to be *root* for at least some of this.

The default IP of this kernel on the Mini2440 is 192.168.1.230 with mask 255.255.255.0 and to get on the same subnet with your PC, add an IP address.

```
# ip add 192.168.1.2/24 eth0
```

Make sure it is not the same as something else on your network. The /24 provides the 255.255.255.0 subnet mask.

For some users, this does not take effect unless they check the configuration (?)

```
# ifconfig -a
```

Now FTP to the Mini2440 and 'put' hello on the Mini2440.

Do this from the /examples/hello directory. On our demo system it looks like this:

```
debian:/opt/FriendlyARM/mini2440/examples/hello# ftp 192.168.1.230
220 FriendlyARM FTP server (Version 6.4/OpenBSD-ftpd-0.17) ready.
Name (192.168.1.230:root) root
331 Password required for root.
Password: [whatever your password is]
230 user root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put hello
(transfer mode, bytes transferred, etc. are reported)
ftp> close
ftp> bye
debian:/opt/FriendlyARM/mini2440/examples/hello#
```

On the Mini2440 use the stylus and keyboard (or remote terminal) and enter ./hello to get the famous Farnsworth greeting.

```
~# ./hello
Good news everyone!
~#
```

That is all. Good work everyone!

Use a USB Storage Device This is a fairly straight forward and simple way to move compiled code to your Mini but is more time consuming when you are writing and troubleshooting.

An SD card USB adapter is a good choice for this. On many PC Linux distros it will auto-mount. If not, try this.

```
# mount /dev/sda1 /mnt
# cp hello /mnt
# umount /mnt
```

Of course, you have to be in the hello directory or use the full path to hello.bin.

Remove the USB device and plug it into the USB port on the Mini2440. The USB device will be auto-mounted as device /udisk. You can run the program directly from the device.

```
~# cd /udisk
/udisk# ./hello
Good news everyone!
/udisk cd ..
~#
```

Of course, you can copy hello to any directory on the Mini2440.