Outside the box: an introduction embedded Linux and hardware interfacing using Snowball

Embedded Linux Conference Europe 2011

Date: Tuesday 25th October
Presenter: Chris Simmonds, chis@2net.co.uk

1. Overview

These notes are to help you get up and running with embedded Linux on the Snowball board.

At this point, you should have

- A laptop
- A bootable USB stick (from the instructor)
- A Snowball board
- 1 Ethernet cross-over cable

Boot your laptop from the USB stick and log with user name **ubuntu** and password **ubuntu**.

2. Flash the Snowball

This will load the Linaro 11.11 image into the on-board eMMC memory.

Connect the USB-serial cable to the PC, but **do not** connect the USB OTG cable yet and do not insert an SD card. Then

```
cd ~/elce-2011/images
sudo riff -m erase
```

You should see

```
$ sudo riff -m erase

 ---------------------- riff - Raw Image File Flasher
------------------------
 Version: 0.4.2
 Flash a device. Try `riff --help' for more information.
 ------------------------------------------------------------------------
-----
16:31:31  - Listening on USB for device connection...
```

Now plug the cable from the USB OTG port into the laptop and it will erase contents of flash memory.

When it is done, load the new image by typing

```
sudo riff -m flash -f snowball-image-V3-20110620
```

Press the reset button on the board. You should see the output below and a progress indicator. The flash load takes several minutes.

```
 ---------------------- riff - Raw Image File Flasher
------------------------
 Version: 0.4.2
 Flash a device. Try `riff --help' for more information.
 ------------------------------------------------------------------------
-----
14:34:53  - Listening on USB for device connection...
14:34:58  - Connected Device@0
14:34:58  - Flashing raw image...
14:34:58  - Filesize: 3221225472
100.0 %
14:43:25  - Flashing finished successfully
14:43:25  - Disconnected Device@0
```

3. Boot up!

Type `minicom -s` to start it in setup mode. Select “serial port setup” and press 'a' to change the Serial Device to /dev/ttyUSB0. Press 'e' and select speed 115200 no parity 8 data bits, one stop bit (1152008N1) and press enter. Select No hardware flow control and no software flow control.

Press enter, then select “Save setup as dfl” and press enter again. Finally “Exit from Minicom”.

Plug in the USB serial adapter cable

Plug your Ethernet cross-over cable between the board and your laptop

Press the reset button

Start minicom and check that the board boots

If all goes well, you should see the kernel load, boot and print a login prompt on the console.

Have a look round, for example:

- What is the kernel command line? (cat /proc/cmdline)
- What type of CPU? (cat /proc/cpuinfo)
- What processes are running? (ps)
- How much free memory? (free)
- What type of file systems are mounted? (cat /proc/mounts)

On your laptop, copy your hello-arm program to ~/rootdir/usr/bin and run it on the target.

Configure the network

Configure the Snowball to use a static IP address

```
ifconfig eth0 192.168.1.101
```

Likewise, configure your laptop to use an IP address on the same network

```
ifconfig eth0 192.168.1.1
```

Check that you can ping one from the other

Note that you will have to re-run the ifconfig if the board is reset

Finally, give a password to the root account so that you can log in using ssh

```
passwd
```

Make a note so you don't forget it

Check that you can log from your laptop using

```
ssh root@192.168.1.101
```

Write a program!

There is a simple "hello world" program in ~/elce-2011/hello. You should customise it a little and compile it using "make". Then copy it to the Snowball using scp:

```
scp hello root@192.168.1.101:/usr/bin
```

And check that it runs on the board.

4. GPIO

Configuring a pin

In the minicom session with the Snowball board, export GPIO pin 142, which is connected to the user LED

```
cd /sys/class/gpio
echo 142 > export
```

Note that directory gpio142 has been created and that it is currently configured as an

input:

```
cat  gpio142/direction
```

Make it an output and turn the led on and off

```
echo out > gpio142/direction
echo 1 > gpio142/value
echo 0 > gpio142/value
```

Generating interrupts

Ideally there would be a button that generated an interrupt on the board, but we can at least show that interrupts can be generated and handled using the gpio files.

Verify that GPIO 142 is not configured to generate interrupts

```
cat gpio142/edge
```

Now change it to trigger on a rising edge

```
echo rising > gpio142/edge
```

Note that interrupt 302 has been registered for "gpiolib" and there there are zero interrupts so far

```
grep gpio /proc/interrupts
```

Now turn the LED on and off again and check the interrupts. You should find that there has been one interrupt on 302

```
echo 1 > gpio142/value
echo 0 > gpio142/value
grep gpio /proc/interrupts
```

5. I2C peripherals

The Snowball has several devices connected to I2C adapters

Running "i2cdetect -l" shows that we have 4 I2C adapters.

"ls /sys/bus/i2c/devices" lists all the I2C peripherals by bus and address, as well as the 4 adapters found by "i2cdetect -l".

Another view of the hardware is to look in /sys/bus/i2c/drivers. Here you will find entries for all the i2c drivers that are configured and loaded in the kernel. You should see

lsm303dlh_a	Accelerometer
lsm303dlh_m	Magnetometer
l3g4200d	Gyroscope

The LSM303DLH accelerometer

You may want to look at the datasheet in ~/elce-2011/datasheets/LSM303DLH.pdf

Type ls /sys/bus/i2c/drivers/lsm303dlh_a and note the numbers:

```
  2-0018  bind  uevent  unbind
```

Set up a convenient alias, ?-???? in this case is 2-0018

```
ACC=$(echo /sys/bus/i2c/drivers/lsm303dlh_a/?-????)
```

Start the sensor

```
echo 1 > $ACC/mode
echo 1 > $ACC/sleep_wake
while true; do sleep 0.2; cat $ACC/data; done
```

Move the board to see the output change.

The LSM303DLH magnetometer

Do ls /sys/bus/i2c/drivers/lsm303dlh_m and note the numbers:

```
  2-001e  bind  uevent  unbind
```

Set up a convenient alias, ?-???? in this case is 2-0018

```
ACC=$(echo /sys/bus/i2c/drivers/lsm303dlh_m/?-????)
```

Start the sensor

```
while true; do
    sleep 0.2;
    grep 1 $ACC/mode > /dev/null && echo 0 > $ACC/mode
    echo 1 > $ACC/mode;
    cat $ACC/data
done
```

Move the board to see the output change.

6. Input events

There are two input devices that we can try out

- Power reset button (blue button near the 5V DC power connector
- LPS001WP barometer chip

To get a list of input devices, look in /dev/input and use input-events on each one:

```
ls -l /dev/input
input-events 0
input-events 1
```

```
input-events 2
```

The power reset button

Run input-events 1 and press the button

The LPS001WP barometer chip

Enable the i2c part of the device:

Set up a convenient alias, ?-???? in this case is 2-005c

```
D=$(echo /sys/bus/i2c/drivers/lps001wp_prs_sysfs/2-005c)
```

Start the sensor

```
echo 1 > $D/enable
```

Read from the sensor using

```
input-events 2
```

Wave something over the board to create an air stream, but be careful not to accidentally hit the board which could damage the chip.

7. Other things you can do

Load the Android image

On your laptop

```
cd ~/elce-2011-images
riff -m flash -f snowball-android-v5-20111012.img
```

Reset the board.

Compile your own kernel

Extract the files for the target board into your home directory:

```
cd ~
tar xf ~/elce-2011/kernel/igloo-kernel-2.6.38.7.tar.gz
```

Copy the mkimage utility that is needed to create the U-Boot header

```
sudo cp ~/elce-2011/kernel/mkimage /usr/bin
```

Select the default configuration for the target and build a kernel with the U-Boot header:

```
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-
make u8500_defconfig
make oldconfig
make uImage
```

Check:

vmlinux size = 86727542

arch/arm/boot/uImage = 3451580

8. Set up from scratch

Follow these instructions to recreate the environment on the USB stick

1. Install Ubuntu 11.04
2. Install extra Ubuntu packages

```
$ sudo apt-get install minicom nfs-kernel-server tftpd-hpa libncurses5-dev
patch quilt g++ manpages-dev
```

3. Install riff

```
$ sudo add-apt-repository 'deb http://igloocommunity.org/download tools
```

```
contrib main non-free'
$ sudo apt-get update
$ sudo apt-get install riff
```

4. Install Linaro toolchain

```
$ sudo apt-get install gcc-arm-linux-gnueabi
```