

uControl Platform SL

User Manual

Hardware and Software Overview



Introduction

uControl Platform is an Embedded Computer (Single Board Computer) based on the 667MHz Samsung S3C6410 (ARM11) microcontroller. *uControl* Platform embodies abundant of built-in resources and powerful video processing capacity, which make it reliable for the development of higher-end products.

Together with the *uControl* Platform we provide BSP (Board Support Package) for Windows CE and Linux Embedded including basic drivers for all the components on the board and demonstration programs, which we believe can help the users in understanding the ARM architecture and shortening their development circle.

uControl Platform package includes the following:

- *uControl* Platform touchscreen and Control boards
- 5V power supply
- Software DVD

When using the Development Board for the first time, please do read and follow the user manual to prevent unnecessary troubles and damages.

Every time before powering on the Development Board, please touch anyone of the metallic interface with your fingers to unload the Electrostatic. Do not touch the chips with your fingers!

OrderOptions

uControl Platform can be ordered in the following standard configurations:

uControl O-M (EXAMPLE *uControl L-S*)

O – Operating system

- A – Android 2.1
- L – Linux Embedded 2.6
- W – Windows CE6.0
- E – without OS

M – NAND RAM

- A – Android 2.1
- L – Linux Embedded 2.6
- W – Windows CE6.0
- E – without OS

TABLE OF CONTENTS

1.	<i>General Description</i>	5
2.	<i>Boot Mode</i>	8
3	<i>uControl Interface Board Schematics</i>	10
3.1	DC Power Supply	10
3.2	MMC1	10
3.3	USB Host	10
3.4	Audio Line-in and microphone	11
3.5	CAN BUS	11
3.6	UART Interfaces	12
3.7	RS485	12
3.8	Keypad GPIO	13
3.9	SPI&I2C	13
3.10	RTC Battery Connector	13
3.11	GPIO1	14
3.12	JTAG	14
3.13	Mini USB	15
3.14	Ethernet RJ45	15
3.15	uSD	16
3.16	Key Inputs	17
3.17	Reset Button	17
3.18	LCD Connectors	17
3.19	User LEDs	18
4.	<i>Software Instalation</i>	20
4.1	Installing Cross-Compiler	20
4.2	Compiling u-boot-1.1.6	20
4.3	Compiling Linux Kernel	21
4.4	Burning U-BOOT	22
4.5	Burning Linux Kernel	27
4.6	Burning File System	29

5	<i>Configuring and Testing functions in Linux</i>	32
5.1	Configuring LCD Resolution	32
5.2	Recalibrating Touch Screen	32
5.3	Testing SD/MMC Card	33
5.4	USB Host Test	33
5.5	Network Test	35
6	<i>Writing simple applications</i>	36
6.1	Create a .c file	36
6.2	Cross compiling	37

1. General Description

The *uControl* Platform and its software package provide LCD touch interface for industrial, medical and consumer appliances. The board provides various physical means of communication as well as support for popular industrial and commercial protocols. It is high performance Single Board Computer based on the ARM11 architecture micro-processor Samsung S3C6410. The board support wide range of peripherals and interfaces such as USB, SD, LCD, Ethernet, RS485, industrial CAN Bus and powerful video controller. The *uControl* Platform is the ideal product for developing and implementing different types of industrial applications and controls.

uControl Platform include S3C6410 CPU, mobileDDR RAM, NAND Flash and NOR Flash. We provide BSP (Board Support Packages) for Android 2.1, Embedded Linux 2.6 and Windows CE6.0 which provide drivers for all the bottom interfaces, devices and utility applications.

The S3C6410 is one of the highest performance 32bit RISC microprocessors designed to provide high performance and low-power capabilities. It includes powerful hardware accelerators for motion video processing, audio processing, 2D graphics acceleration, display manipulation and scaling. Integrated Multi Format Codec supports encoding and decoding of MPEG4/H.263/H.264 and decoding of VC1.

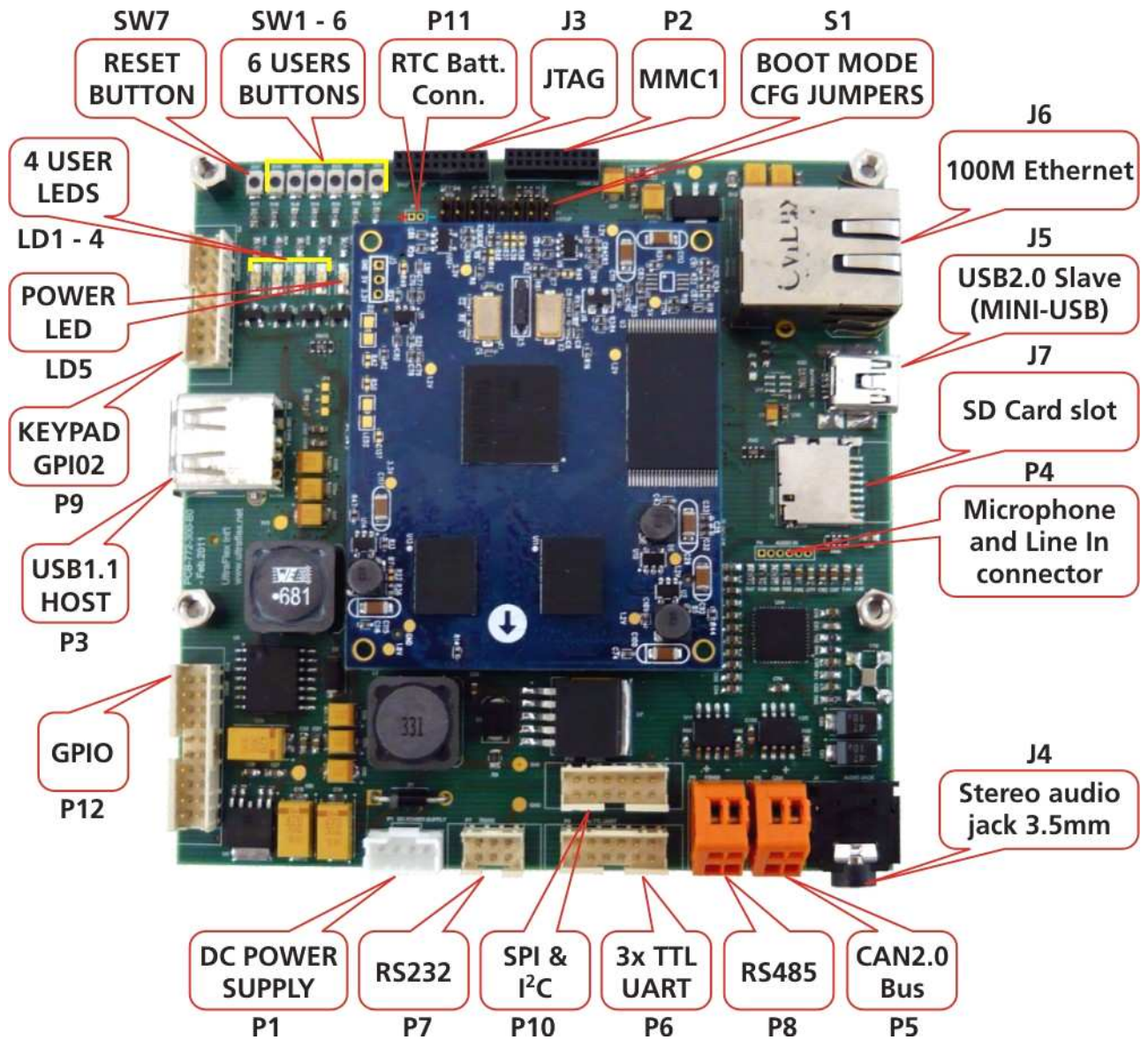
The microprocessor includes:

- ARM1176ZJF 667MHz VFP/SIMD;
- Java acceleration engine 16KB/16KB I/D Cache and 16KB/16KB I/D TCM
- 65nm low-power process;
- 133MHz expansion bus frequency;
- 32bit data bus and 32bit external address bus;
- 32 channel DMA controller;
- Built-in hardware and Multi-Format Codec;
- AC-97 audio codec interface and PCM serial audio interface
- SRAM/ROM/NOR Flash Interface with x8 or x16 data bus.
- Muxed OneNAND Interface with x16 data bus.
- NAND Flash Interface with x8 data bus.
- SDRAM Interface with x32(Port1) data bus.
- Mobile SDRAM Interface with x32(Port1) data bus
- DDR Interface with x32(Port1) data bus
- Mobile DDR Interface with x32(Port1) data bus

The *uControl* Platform have 128MBytes DDR RAM, NAND Flash, audio, USB, UART and Ethernet. Board introduces most signals that provided by the ARM11 S3C6410, like Matrix Keypad, USB OTG, USB HOST, SDIO, LCD, Touch Screen, Camera, AC97, UART, SPI, I2C, ADC, DAC, PWM, EXT INT, GPIO and so on. Additional ISP protection is provided for the touchscreen, USB and the ADC0-3 modules.

Platform Features:

- 667MHz Samsung S3C6410X microcontroller;
- 128/256MBytes mDDR RAM;
- 266MHz system bus;
- 256/1024MBytes NAND Flash;
- WM9714 audio codec, AC97 interface;
- 10/100M DM9000A Ethernet chip;
- One reset button implemented with specific reset chip;
- 8-bit jumper for selecting boot mode;
- Four serial ports, including 1 5-wire RS232 (P7) and 3 3-wire TTL (P6 - 12pin 2.0mm-pitch connector);
- One RS485 (P8);
- One CAN2.0 Bus (P5);
- One 100M Ethernet port with DM9000AEP and indicators;
- One USB Host interface supporting USB1.1 protocol (P3);
- One USB Slave (MINI-USB) interface supporting USB2.0 protocol (J5);
- One high-speed SD card slot supporting SD Memory and SDIO (J7);
- One 3.5mm standard Stereo audio jack (J4);
- One connector with microphone and line-in (P4);
- LCD and touch screen interface supporting 3.5", 4.3", 5.6", 5.7", 7", 8" TFT LCD and 10" LVDS LCD;
- Built-in RTC (Real Time Clock) with back-up battery;
- One JTAG interface using 2*10pin connector (J3);
- Six user buttons & four user LEDs;
- One 2*10pin expansion interface (P12) 4 ADC input, 2 DAC output, 1 PWM, 1 Clock output, 3 GND, 3.3V output and others as normal IOs (P12);
- 1 x SPI and 1 x I2C interfaces (P10);
- 1 Keypad connector (P9);



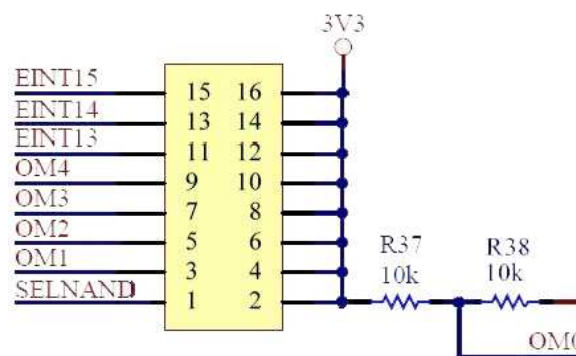
2. Boot Mode

The S3C6410 microcontroller supports booting from either NAND Flash, Nor Flash or SD card. The boot mode can be set by setting 8-bit toggle switch S1 Setup.

SETTING	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
Layout	SEL NAND	OM4	OM3	OM2	OM1	GPN15	GPN14	GPN13
NAND Flash	1*	0	0	1	1	X	X	X
NOR Flash	X	0	1	0	1	X	X	X
SD card	1	1	1	1	1	0	0	0

* “1” indicates “ON” while “0” refers to “OFF”, “X” indicates high/low power level

The OM0 is a signal that selects the clock source. When the OM0 is set as “0”, XTlpll is selected as the clock source; while when the OM0 is set as “1”, EXTCLK is selected. On the *uControl* Platform SBC we selected XTlpll.

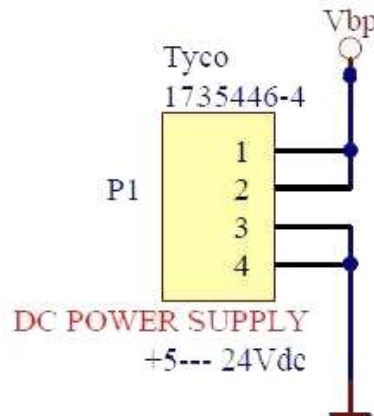


The signal SELNAND determines the type of NAND memory. It should be “1” (high power level) when using NAND Flash while “0” (low) when using ONENAND. On the *uControl* Platform, we use NAND Flash; therefore the power level of the SELNAND signal is high.

EINT13, EINT14 and EINT15 are pins that set the boot device for IROM boot mode. When booting in IROM mode, the S3C6410 microcontroller first runs the in-chip ROM firmware, reads the statuses of EINT15, EINT14 and EINT13, then, according to the statuses of the pins, selects the boot device. OM1-OM4 are the pins that set the boot mode of the S3C6410.

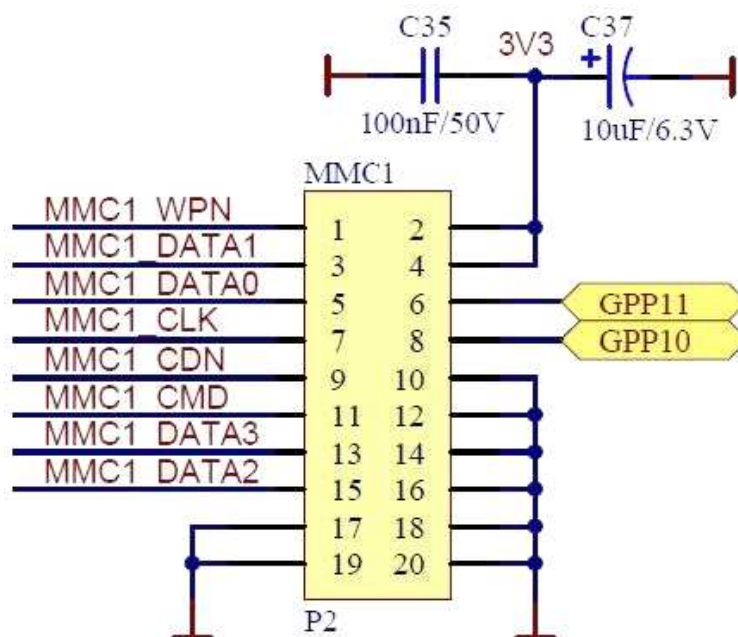
3 uControl Interface Board Schematics

3.1 DC Power Supply



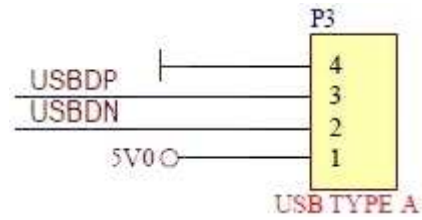
3.2 MMC1

Optional interface for WiFi 802.11 b/g SD/MMC adapter.



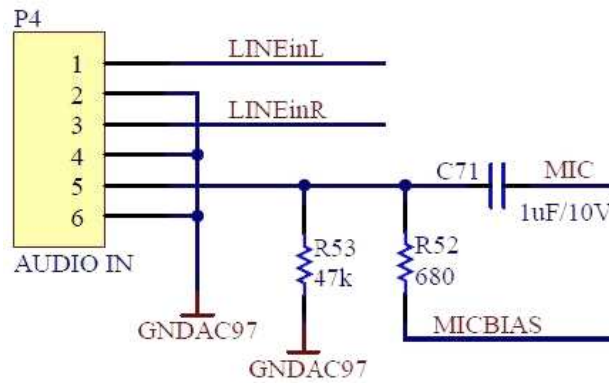
3.3 USB Host

The USB Host interface on the *uControl* Platform supports USB1.1 protocol; it can be used to connect USB keyboard, USB hard disk, USB mouse, USB memory stick, etc.



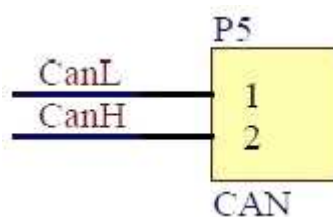
3.4 Audio Line-in and microphone

The *uControl* Platform SBC takes the AC97 bus on the S3C6410 to implement audio I/O. The WM9714 sound chip realizes audio output, Line in and MIC input.



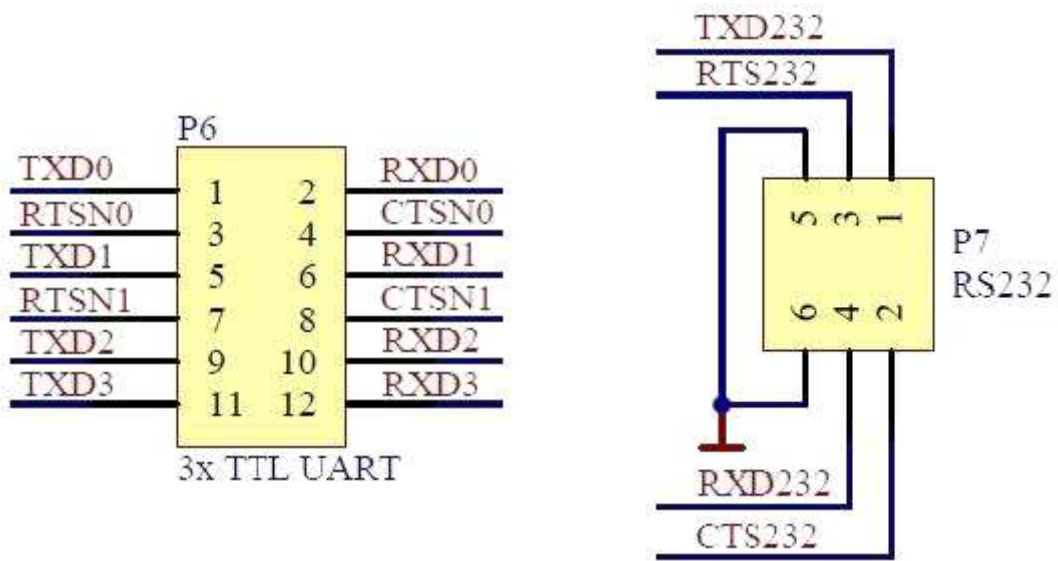
3.5 CAN BUS

The CAN Bus on the *uControl* Platform is implemented by the MCP2515 chip, which supports CAN2.0 standard.

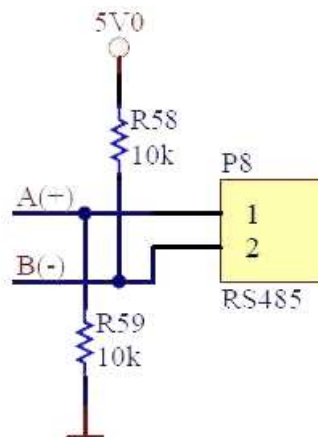


3.6 UART Interfaces

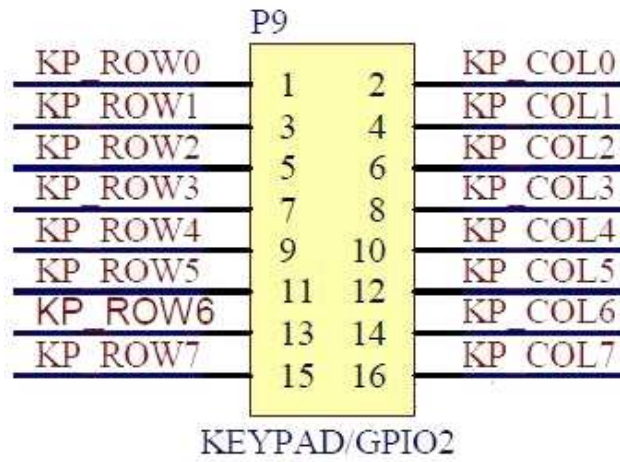
There are a total of four serial ports on the *uControl* Platform: one 5-wire RS232 serial port and three 3-wire TTL serial ports. The RS232 has been used as debug port by default, which can be connected to the PC to display debug information. By default P7 is connected to UART0, but there is an option to connect UART1 instead, if the resistor matrix from RN16 is replaced to RN17.



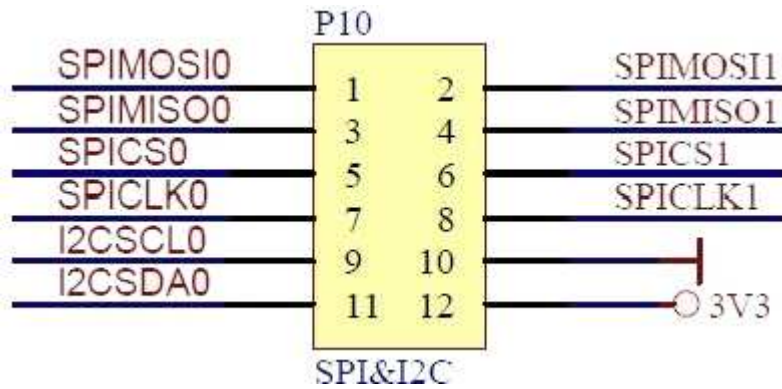
3.7 RS485



3.8 Keypad GPIO

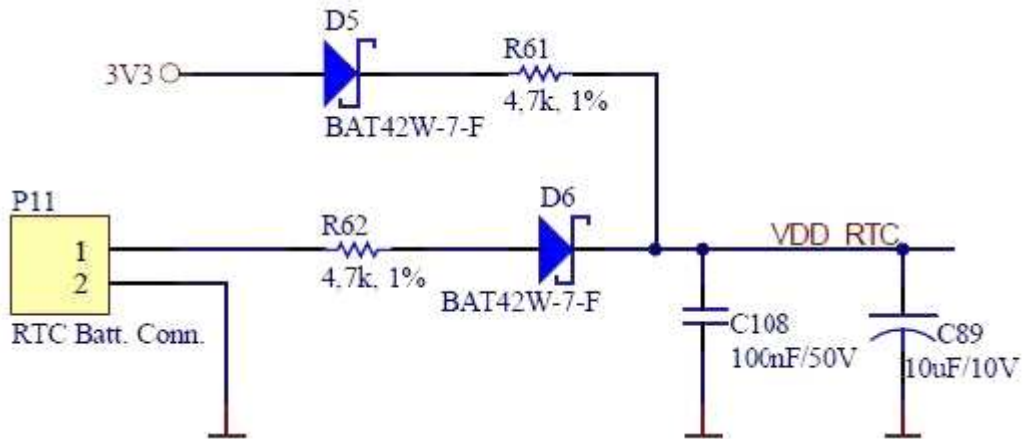


3.9 SPI&I2C



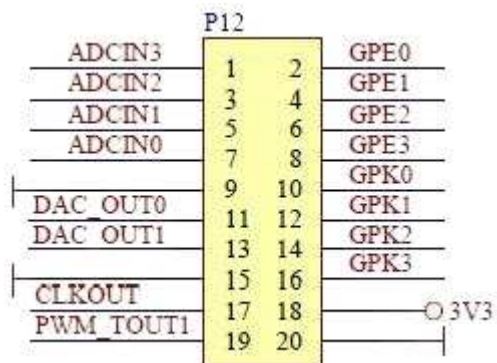
3.10 RTC Battery Connector

The Real Time Clock on the *uControl* Platform is powered by a button battery.



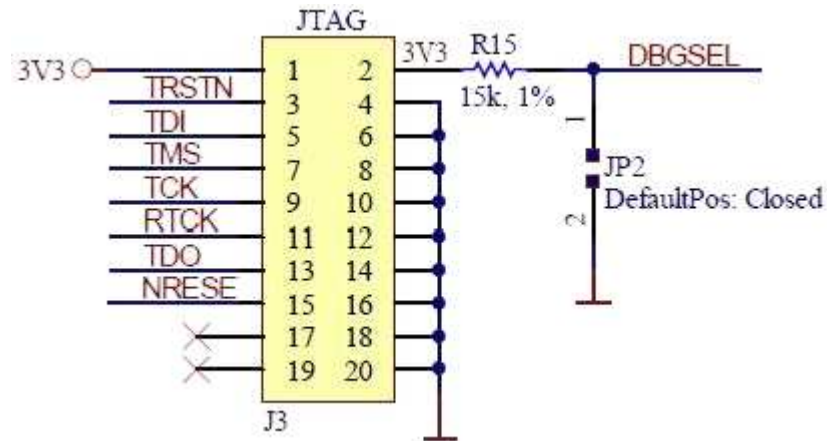
3.11 GPIO1

2 x 10pin IO pins are reserved on the *uControl* Platform for the users, which includes 4 ADC input, 2 DAC output, 1 PWM, 1 Clock output, 3 GND, 3.3V output and others as normal IOs.



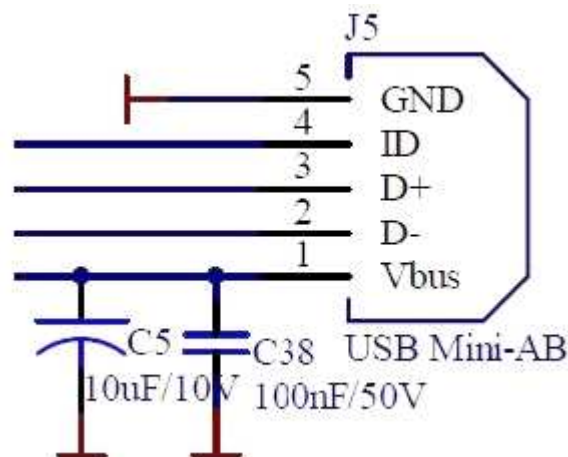
3.12 JTAG

The JTAG interface on the *uControl* Platform provides access to both the ARM11 core and S3C6410 in-chip peripherals. The DBGSEL signal determines which device to access: When the power level of DBGSEL is high, the JTAG interface provides access to the S3C6410 in-chip peripherals; when the power level of DBGSEL is low, the JTAG interface provides access to the ARM11 core. The power level of DBGSEL can be set by the jumper JP2.



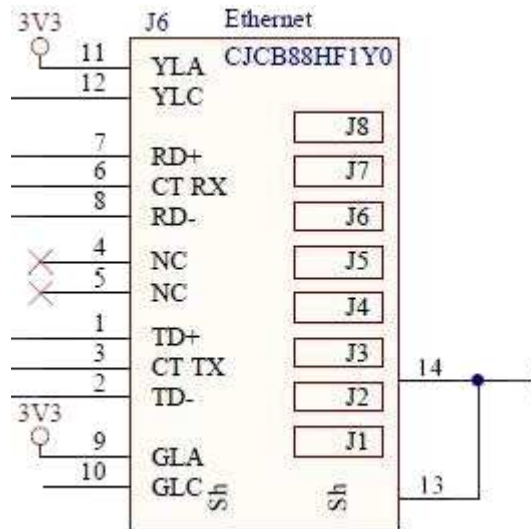
3.13 Mini USB

The USB OTG interface (Mini USB A/B port) supports USB 2.0 protocol with the maximum speed up to 480Mbps. The USB OTG interface can be used for downloading programs.



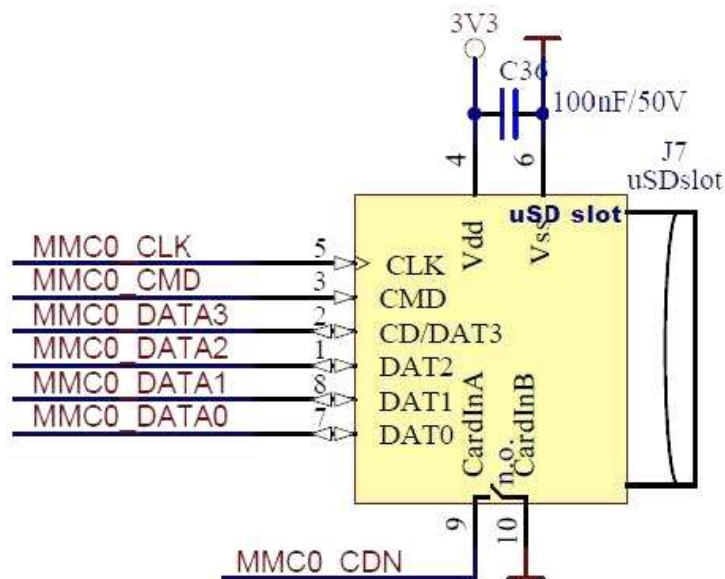
3.14 Ethernet RJ45

DM9000 Ethernet chip is used on the *uControl* Platform to implement 100M Ethernet. During the development, the Ethernet port can be used to download WinCE binaries, mount NFS in Linux. The interrupt signal takes the EINT7 on the S3C6410.

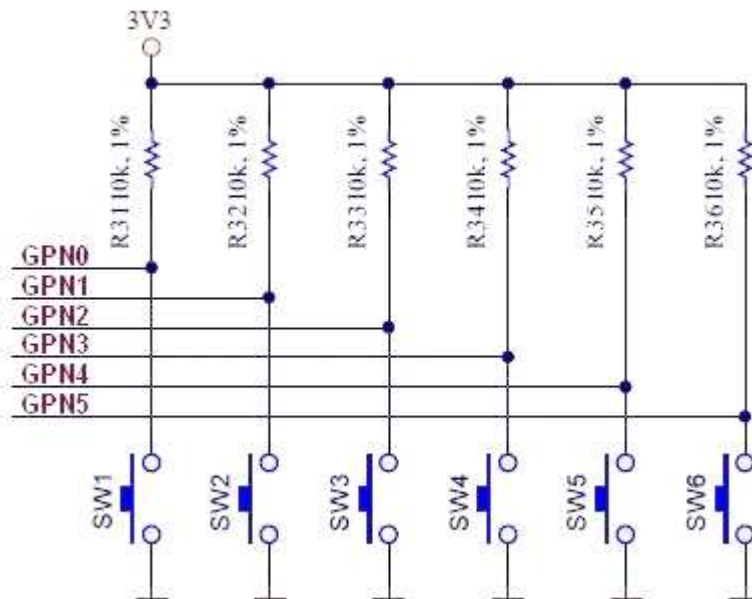


3.15 uSD

The 4-wire SD card slot supports both SD Memory 2.0 and SDIO 1.0 protocol. As SD memory interface it supports up to 8GB SD card; as SDIO, it can be used to connect WIFI module, GPS module, and etc.

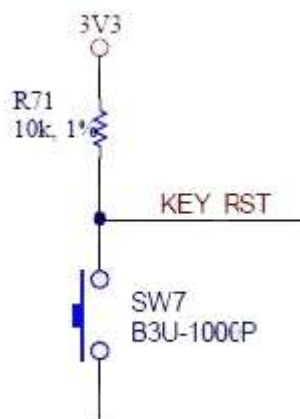


3.16 Key Inputs

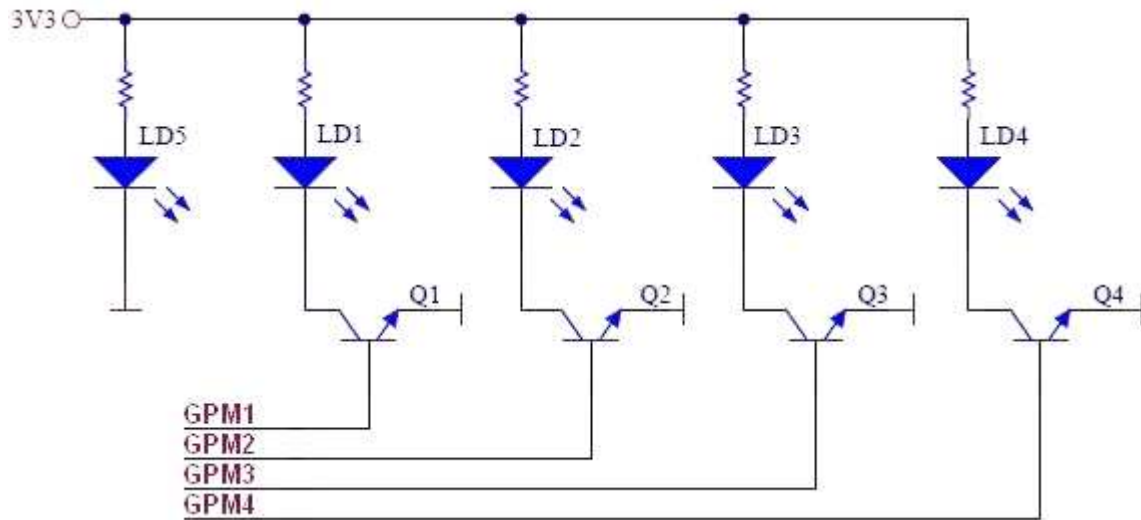


3.17 Reset Button

Masked HW reset input



3.18 LCD Connectors



4. Software Instalation

4.1 Installing Cross-Compiler

Cross compiling simply means to compile executable code for a platform on another platform (a PC for instance). Before cross compiling, we have to install cross compiler tool chains first. A cross compiler named as “Cross-4.2.2-eabi.tar.bz2” has been enclosed in the DVD, to install the cross compiler: Transmit the file “Cross-4.2.2-eabi.tar.bz2” to Ubuntu through FTP and execute the commands below:

- `cd /ultragui`
- `mkdir /usr/local/arm`
- `tar xjvf cross-4.2.2-eabi.tar.bz2 -C /usr/local/arm`

The cross compiler will be installed in “/usr/local/arm/4.2.2-eabi”.

4.2 Compiling u-boot-1.1.6

U-boot, short for Universal Bootloader, is an open source bootloader that boots Linux on the *uControl* Platform. The source code of u-boot can be found in the folder “\linux-2.6.36\u-boot-1.1.6” in the DVD.

Copy the source code “uboot1.1.6_UltraGUI_v1.0.tgz” to the directory “/ultragui” and un-zip it:

- `tar xzf u-boot-1.1.6-2636.tgz`

The commands below compile the u-boot bootloader:

- `cd uboot1.1.6`
- `make smdk6410_config`
- `make`

When the compilation is done, the u-boot binary named as “u-boot.bin” will be found in the directory “uboot1.1.6”, which we can download to the *uControl* Platform and use as Linux bootloader.

4.3 Compiling Linux Kernel

For the *uControl* Platform we provide the Linux-2.6.36.2 kernel. Find the kernel source code named as “linux2.6.36.2.tgz” in the folder “linux2.6.36.2\kernel” in the shipped DVD, copy it to the directory “ultrgui” and un-zip it: `tar xzf linux2.6.36.2.tgz`

Configurations

We might need to install the “libncurses5” before we can execute the `make menuconfig` command.

- `sudo apt-get install libncurses5-dev`
Restore the default configuration:
- `cd linux2.6.36.2`
- `cp OK6410_CFG .config`
(note the blank space between the “G” and “.”)

Configure LCD type: The *uControl* Platform Linux system supports 3.5”, 4.3”, 5.6”, 7”, and 8” LCDs. The LCD types are selected in:
“Device Drivers-> Graphics support->Support for frame buffer devices”

Compiling Kernel

The command below compiles the Linux kernel into “zImage” binary:

- `make zImage`

When the compilation completes the binary image “zImage” will be found in the directory “arch/arm/boot”.

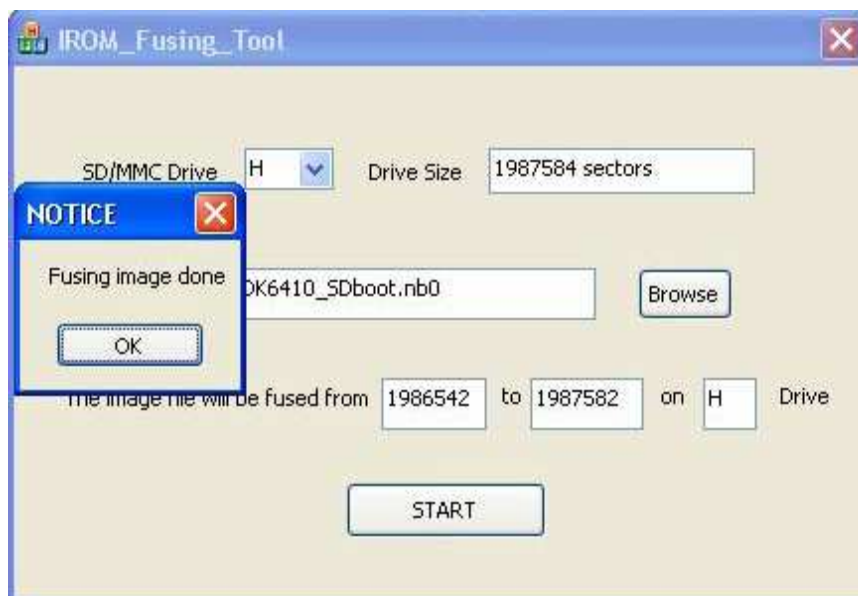
4.4 Burning U-BOOT

Fusing SDbboot to SD card

Insert a SD card formatted as FAT32 to a card reader and connect the card reader to the PC. Find the IROM_Fusing_Tool.exe in the folder “Tools \SDboot” in the DVD, double click to run it. In the IROM_Fusing_Tool window, click “Browse” and open the file “OK6410_SDbboot.nb0” (or TE6410_Sdbboot_sdhc.nb0 if you are using a SDHC card higher than 2GB) in the folder “Tools\SDboot”, select the SD drive and then click the “START” button to start fusing the SDbboot image to the SD card. A dialog box as shown in the picture below indicates that the fusing is done.

Note that there are two SD bootloaders in the directory “Sdbboot”. Use:

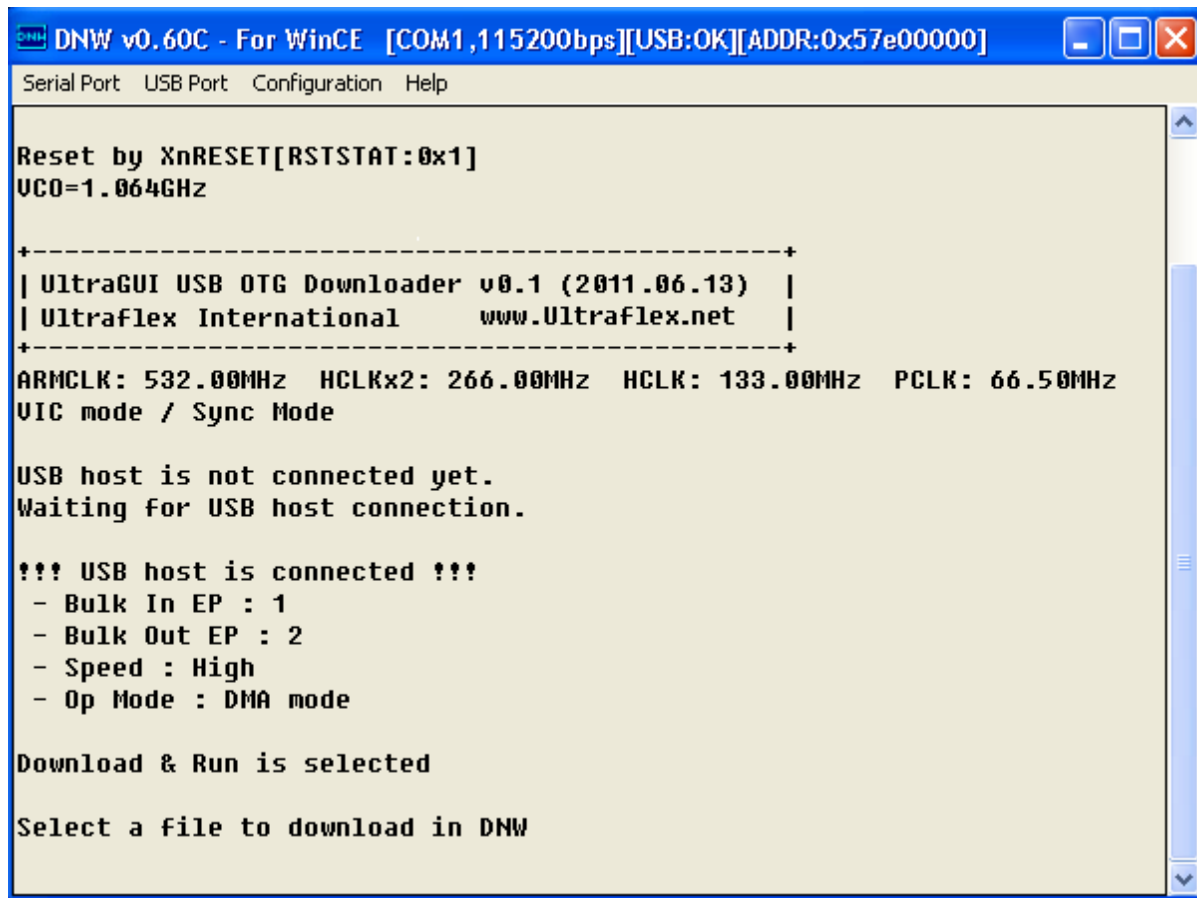
- “OK6410_SDbboot.nb0” for 2GB or lower SD card;
- “TE6410_Sdbboot_sdhc.nb0” for SDHC card bigger than 2GB.



Setting Boot Mode

- Insert the micro SD card to the uSD card slot on the *uControl* Platform.
- Set up the S1 Jumper block to SD card boot mode (as shown chapter 2).
- Connect the RS232 (P7) serial port cable, USB cable and power supply to the *uControl* Platform.
- Find the DNW program in the DVD folder “Tools”, copy it to a local folder and run it.
- In the DNW window, click the menu item “Serial Port->Connect” to connect to the *uControl* Platform via serial port, when it is connected, we will see something like “COM1, 115200bps” on the DNW title bar.

- Power on the *uControl* Platform, we will see the SDboot message in the DNW window

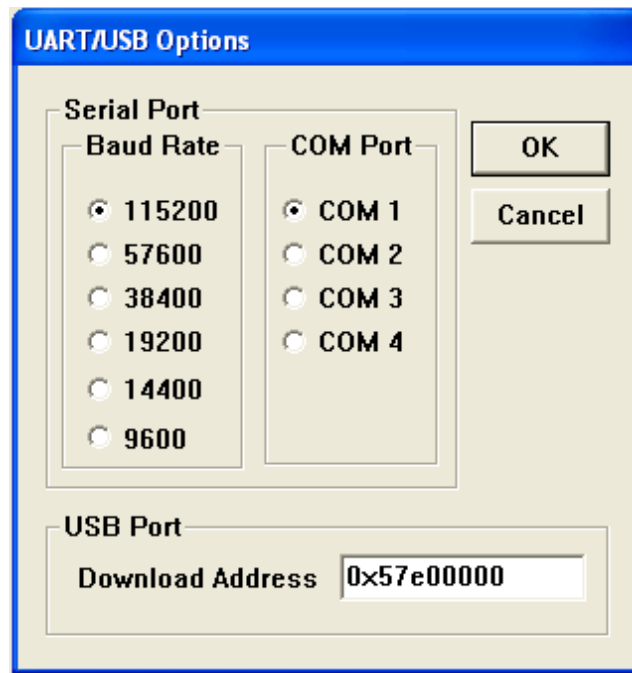


Installing USB Driver

We will get Found New Hardware Wizard if it is the first time that we run the SDboot, Follow the wizard to install the USB driver for DNW (DVD:\Drivers\USB Driver\USB Driver for DNW). As soon as the USB driver is installed, we will see the information like [USB:OK] on the DNW title bar.

Burning U-BOOT to NAND Flash

- Click the DNW menu "Configuration->Options" and configure the download address to "0x57e00000", as shown in the picture below:



- Click “USB Port->Transmit->Transmit”, and open the file “u-boot.bin” in the folder “Linux2.6.36\demo_images” in the DVD:

The “u-boot” boot loader will be downloaded to the *uControl* Platform and booted as soon as it is downloaded. Press the Space Bar during the boot delay to enter the u-boot command menu. As shown below:



```

DNW v0.60C - For WinCE [COM1,115200bps][USB:x][ADDR:0x57e00000]
Serial Port  USB Port  Configuration  Help

CPU:      S3C6410 @532MHz
          Fc1k = 532MHz, Hc1k = 133MHz, Pc1k = 66MHz, Serial = CLKUART
Mode)
Board:    SMDK6410
DRAM:     128 MB
Flash:    0 kB
NAND:     256 MB
In:       serial
Out:      serial
Err:      serial

Hit any key to stop autoboot:  0

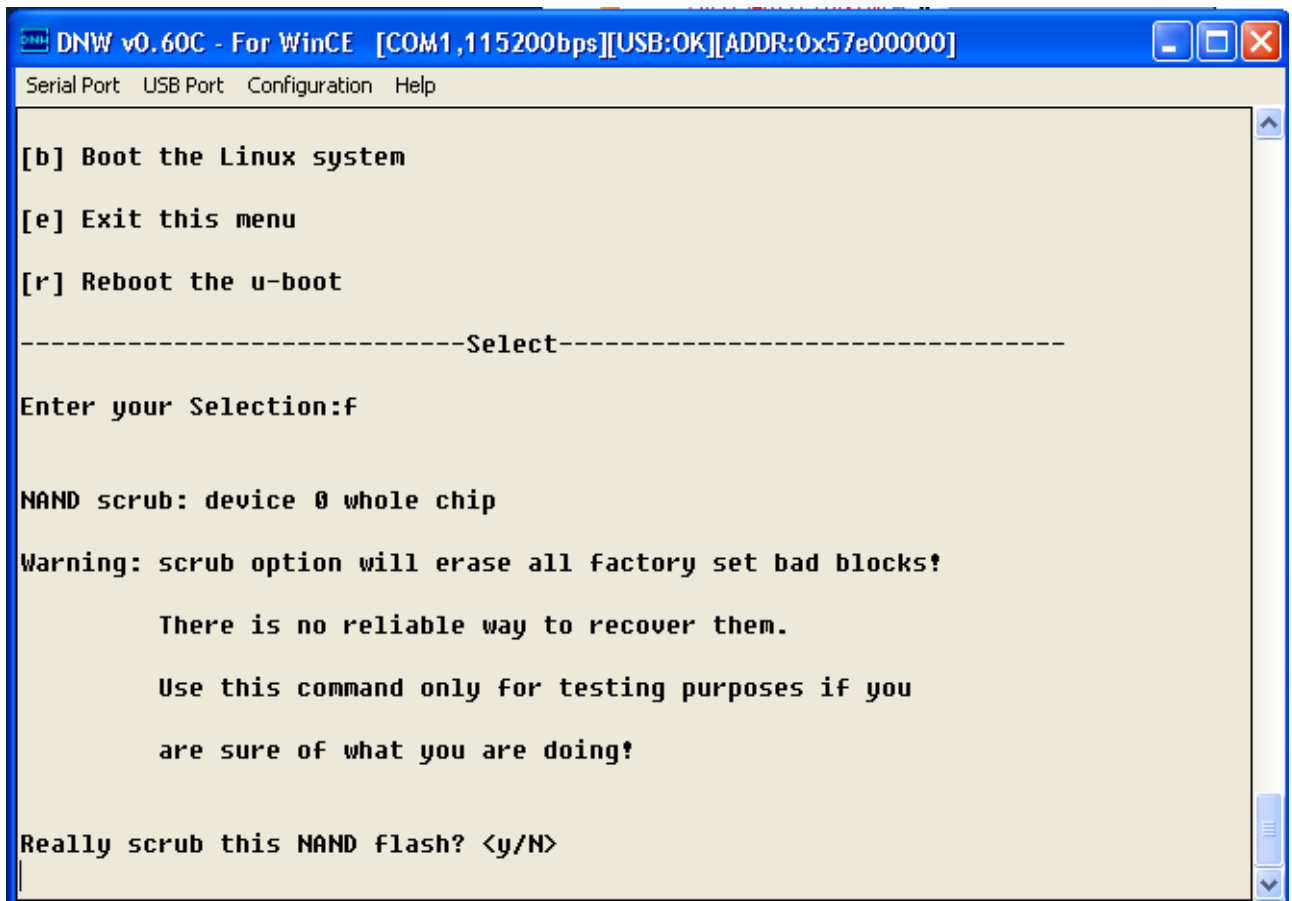
##### User Menu for UltraGUI #####

[f] Format the nand flash
[u] Download u-boot.bin
[k] Download zImage
[c] Download cramfs
[b] Boot the Linux system
[e] Exit this menu
[r] Reboot the u-boot

-----Select-----

Enter your Selection:
  
```

- In the u-boot: command menu, press “f” to format the NAND Flash:

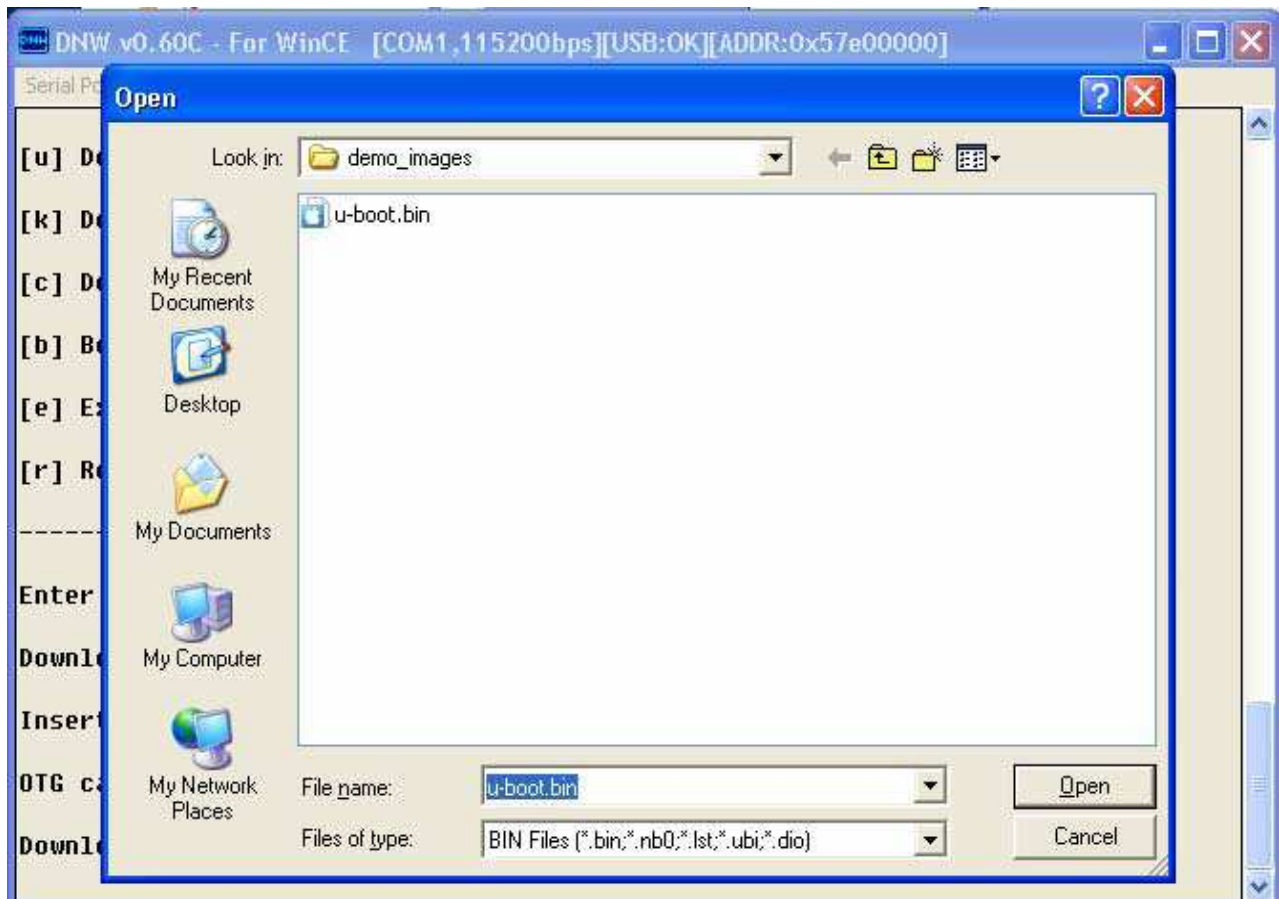


Press “y” and “Enter”:

- Then press “u” to download the u-boot to the NAND Flash:



Click the DNW menu “USB Port-> Transmit-> Transmit” and open “u-boot.bin” in the coming pop-up window to download the u-boot to the Flash:



Setting Nand Flash Boot

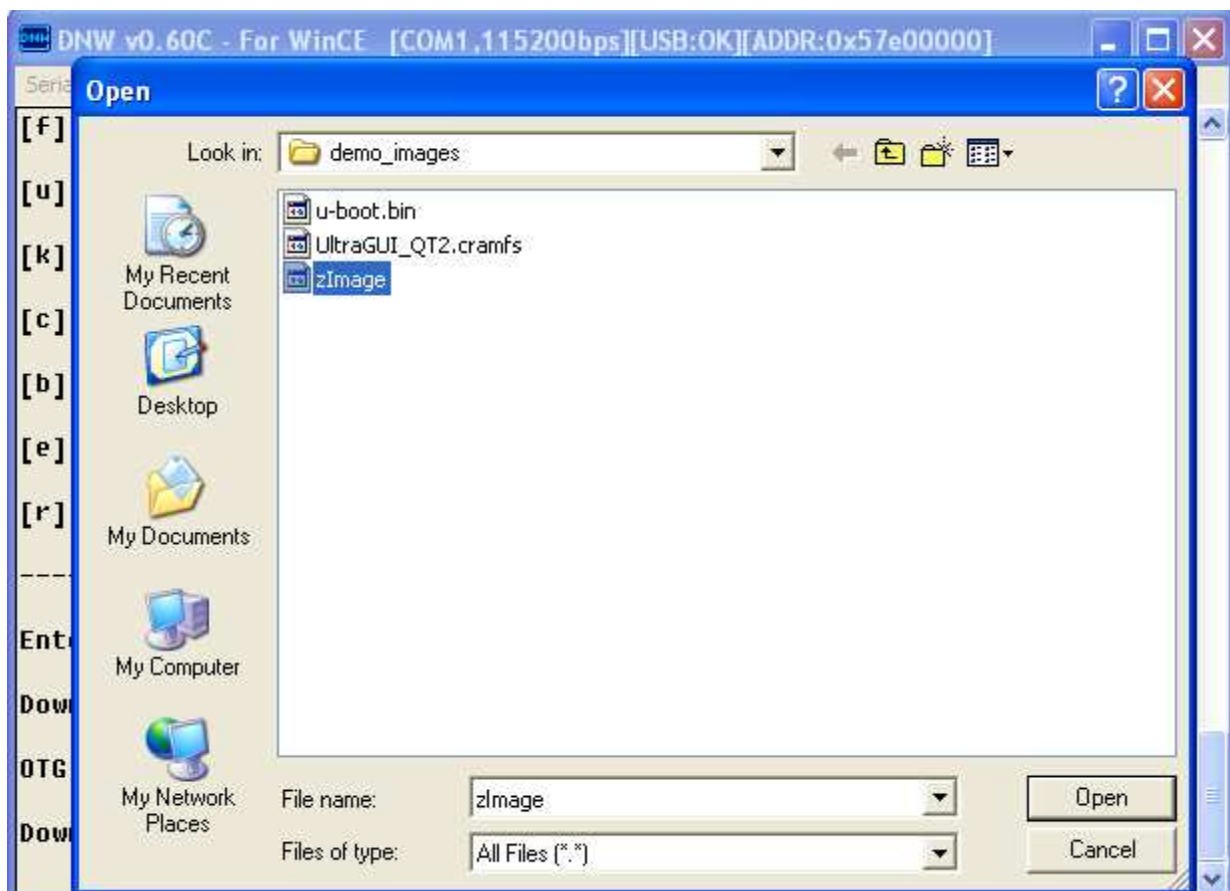
Set the 8-bit toggle switch S1 Setup to boot from Nand Flash (as shown in Chapter 2).

4.5 Burning Linux Kernel

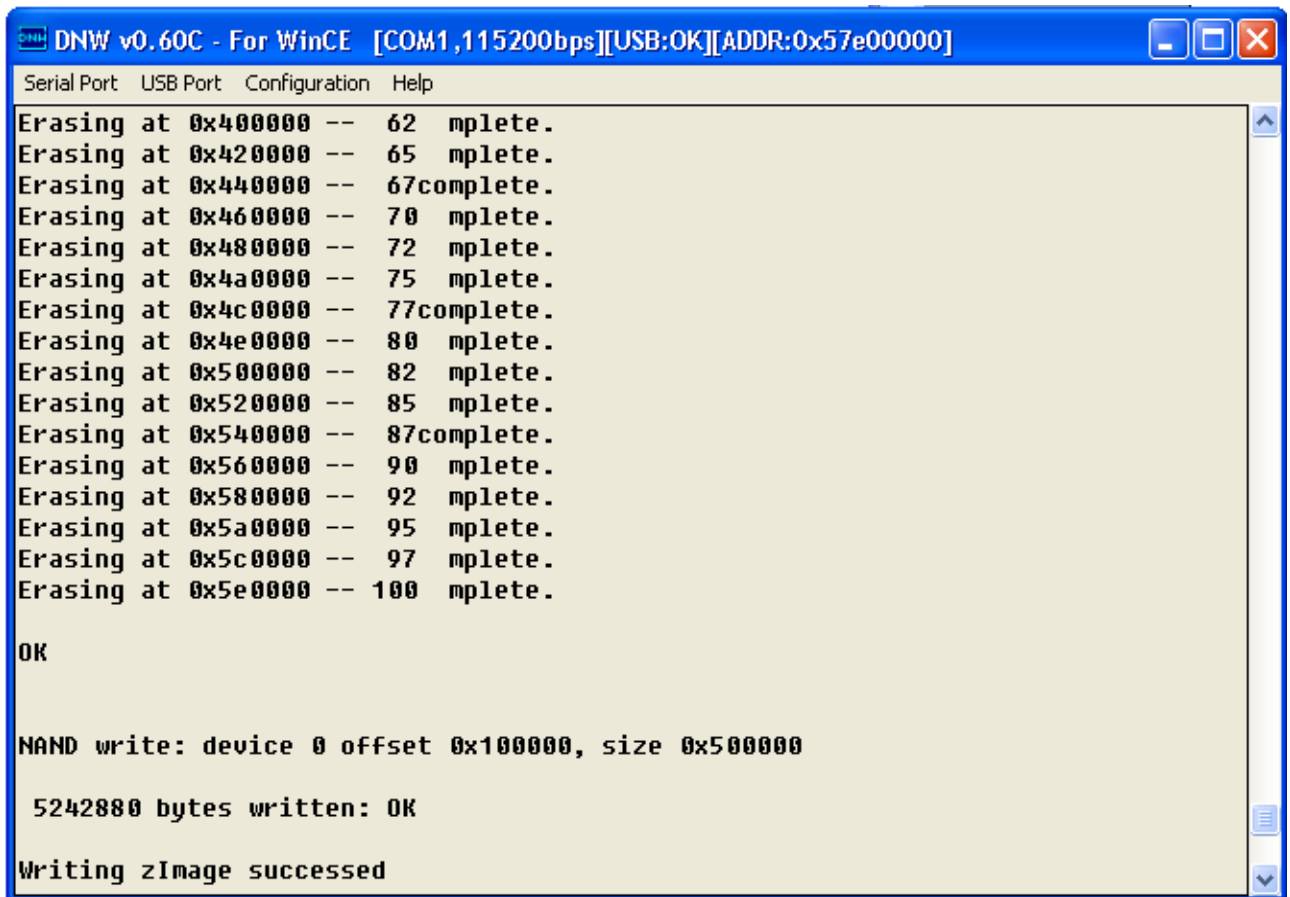
- Before burning the Linux kernel, make sure that the steps in the chapter 4.4 have all been

done, i.e., make sure that the u-boot has been burnt, and the *uControl* Platform boots from NAND Flash.

- Once the above mentioned steps are finished, restart the *uControl* Platform and press any key during the boot delay to enter the u-boot command menu.
- Press “k” and then click “USB Port->Transmit->Transmit”, and open the Linux kernel at “DVD:\Linux2.6.36\demo_images\zImage”:



The Linux kernel will be downloaded and automatically burnt to the NAND Flash.



```

Serial Port  USB Port  Configuration  Help
Erasing at 0x400000 -- 62 mplete.
Erasing at 0x420000 -- 65 mplete.
Erasing at 0x440000 -- 67complete.
Erasing at 0x460000 -- 70 mplete.
Erasing at 0x480000 -- 72 mplete.
Erasing at 0x4a0000 -- 75 mplete.
Erasing at 0x4c0000 -- 77complete.
Erasing at 0x4e0000 -- 80 mplete.
Erasing at 0x500000 -- 82 mplete.
Erasing at 0x520000 -- 85 mplete.
Erasing at 0x540000 -- 87complete.
Erasing at 0x560000 -- 90 mplete.
Erasing at 0x580000 -- 92 mplete.
Erasing at 0x5a0000 -- 95 mplete.
Erasing at 0x5c0000 -- 97 mplete.
Erasing at 0x5e0000 -- 100 mplete.

OK

NAND write: device 0 offset 0x100000, size 0x500000

5242880 bytes written: OK

Writing zImage succeeded
  
```

4.6 Burning File System

Two kinds of filesystems are provided for the *uControl* Platform: cramfs and yaffs2.

Burning CRAMFS Filesystem

- Press “c” in the U-BOOT command menu and download the “DVD:\Linux2.6.36\demo_images\OK6410_Qtopia47.cramfs”:



The cramfs filesystem will be downloaded and automatically burned to the NAND Flash. The burning of cramfs filesystem will take some time, please wait in patience. We will see the u-boot menu again as soon as the burning is done.

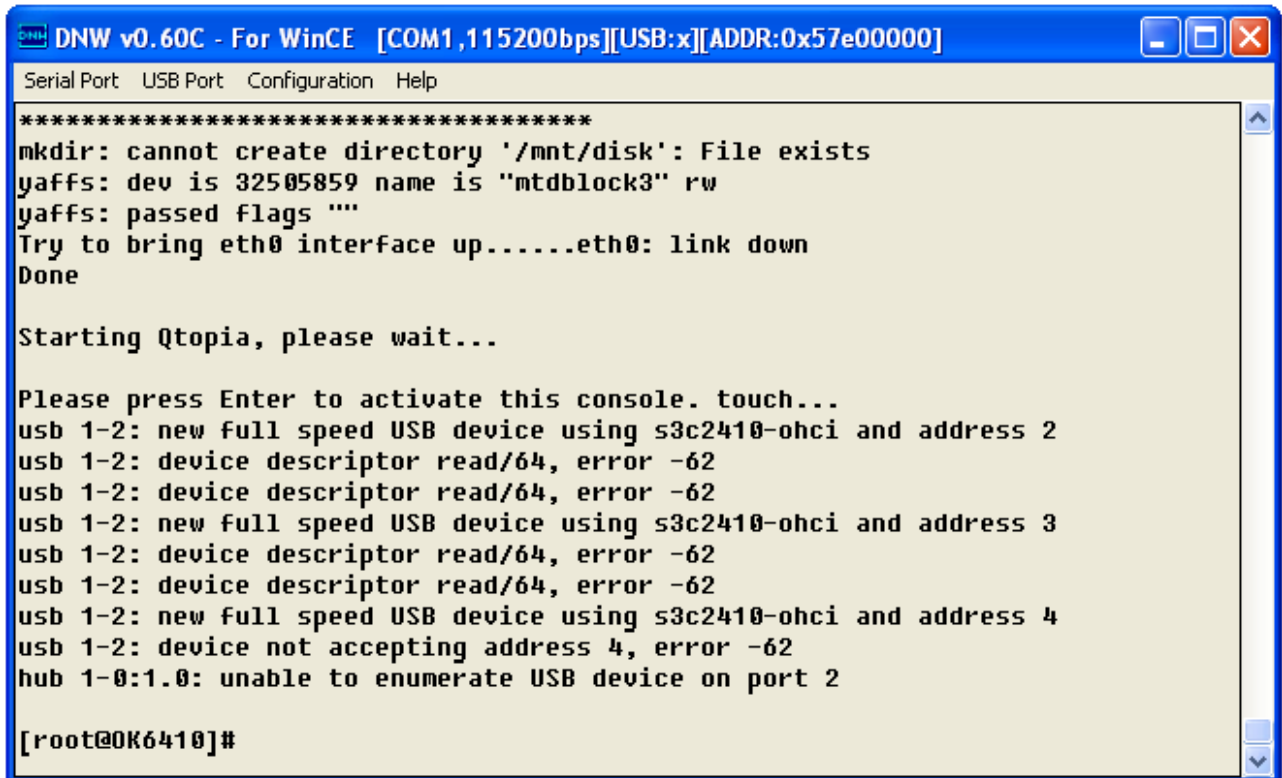
By now we have burned everything needed for a Linux system: bootloader, kernel and filesystem. We can now boot the Linux system by pressing “b” in the u-boot menu. Qtopia interface will be shown on the LCD as soon as the Linux is booted, tap the LCD to enter the application interface.

Burning YAFFS2 Filesystem

CRAMFS filesystem is read-only filesystem, in case that we need to write something to the filesystem while using the *uControl* Platform, we can use the YAFFS2 filesystem.

- YAFFS2 filesystem is provided at “DVD:\Linux2.6.36\filesystem\ OK6410_yaffs2_v3.0.tar”. Copy it to SD card and connect the SD card to *uControl* Platform.

- Boot the Linux system on the *uControl* Platform and enter the Linux console:



```

DNW v0.60C - For WinCE [COM1,115200bps][USB:x][ADDR:0x57e00000]
Serial Port  USB Port  Configuration  Help
*****
mkdir: cannot create directory '/mnt/disk': File exists
yaffs: dev is 32505859 name is "mtdblock3" rw
yaffs: passed flags ""
Try to bring eth0 interface up.....eth0: link down
Done

Starting Qtopia, please wait...

Please press Enter to activate this console. touch...
usb 1-2: new full speed USB device using s3c2410-ohci and address 2
usb 1-2: device descriptor read/64, error -62
usb 1-2: device descriptor read/64, error -62
usb 1-2: new full speed USB device using s3c2410-ohci and address 3
usb 1-2: device descriptor read/64, error -62
usb 1-2: device descriptor read/64, error -62
usb 1-2: new full speed USB device using s3c2410-ohci and address 4
usb 1-2: device not accepting address 4, error -62
hub 1-0:1.0: unable to enumerate USB device on port 2

[root@OK6410]#

```

- The SD card should be automatically mounted to `"/sdcard"`, we can check this by executing the command below:

– `ls /sdcard`

We know it is mounted if files in the SD card are listed.

In case that it is not, we can mount it with:

– `mount /dev/mmcblk0 /sdcard`

- As soon as the SD card is mounted, we can decompress the YAFFS2 filesystem to the NAND Flash:

```
#tar zxvf /sdcard/OK6410_yaffs2_v3.0.tar -C /mnt/disk
```

The decompressing will take some time. Please wait until the command line shows up again.

- Reset the *uControl* Platform and press any key during the boot delay to enter the u-boot

menu, press “e” in the menu to enter the u-boot command line, and execute the command below to configure the boot parameter:

```
#setenv bootargs "root=/dev/mtdblock3 rootfstype=yaffs2
```

```
console=/dev/ttySAC0,115200"
```

and the one below to save the modification:

```
# saveenv
```

- Reset the *uControl* Platform, the YAFFS2 filesystem will be started.

5 Configuring and Testing functions in Linux

In this chapter we test the drivers and services in the Linux system, Hyper-Terminal is recommend to be taken as the serial port console.

5.1 Configuring LCD Resolution

In Linux 2.6.36 we can configure the LCD resolution by modifying the u-boot parameter.

- Press any key during the boot delay to enter the u-boot menu, press “e” in the menu to enter the u-boot command line, and execute the command below to configure the boot parameter:

```
# setenv bootargs "root=/dev/mtdblock2 rootfstype=cramfs console=ttySAC0,115200 lcdsize=35"
```

The command above configures the LCD size to 3.5 inch.

Use the one below for 4.3” LCD:

```
# setenv bootargs "root=/dev/mtdblock2 rootfstype=cramfs console=ttySAC0,115200 lcdsize=43"
```

For 5.6” LCD:

```
# setenv bootargs "root=/dev/mtdblock2 rootfstype=cramfs console=ttySAC0,115200 lcdsize=56"
```

For 7” LCD:

```
# setenv bootargs "root=/dev/mtdblock2 rootfstype=cramfs console=ttySAC0,115200 lcdsize=70"
```

The parameters need to be saved after making the configuration:

```
# saveenv
```

```
# reset
```

5.2 Recalibrating Touch Screen

A touch screen calibration program has been included in the filesystem, we can run it to recalibrate the touch screen.

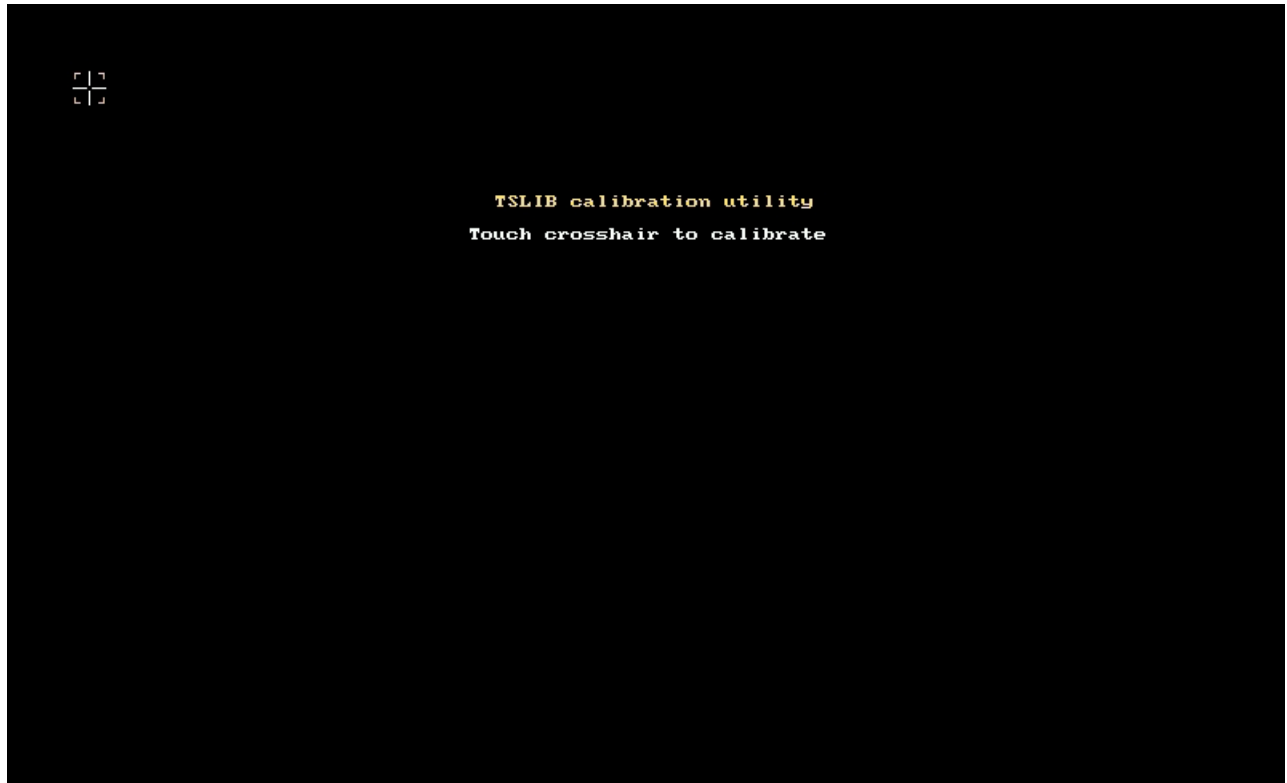
Before running the calibration program we have to terminate the Qtopia:


```
# killall qpe
```

Then execute:

```
# calibrate
```

to run the calibration program. Five cross-hairs will be shown on the LCD, click the center of the cross-hair to calibrate the touch screen.



5.3 Testing SD/MMC Card

As soon as a SD card is inserted into the SD card slot, it will be mounted automatically to the directory “/sdcard” and we should be able to view the files in the card with File Manager.

We can also mount/unmount the SD card using the commands below:

```
# mount /dev/mmcblk0 /sdcard
```

```
# umount /sdcard
```

5.4 USB Host Test

USB Mouse

USB mouse and the touch screen could not be supported at the same time by the QTE. The Filesystem installed on the *uControl* Platform supports touch screen by default, by executing the command below we can switch to USB mouse:

mouseinput and touchinput switches back to the touch screen.

We can also modify the Filesystem to provide support to the USB mouse:

In the file “/etc/init.d/tcS”, find the line “/bin/qtopia &” to “/bin/qtopia_mouse &”, to do this:

```
# cd /etc/init.d
```

```
# vi rcS
```

Move the cursor to the character behind “/bin/qtopia ”, press “i” to insert “_mouse”, then ESC to quit the Insert Mode, and then execute :wq to save the modification. The *uControl* Platform needs to be restored in order to validate such modification.

USB Keyboard

USB Keyboard is Plug-and-Playable on the *uControl* Platform, we will see the information as shown in the picture below when a USB keyboard is connected to the *uControl* Platform.

```
[root@TE6410 /]# usb 1-1: new low speed USB device using s3c2410-ohci and address 7
usb 1-1: configuration #1 chosen from 1 choice
input: USB USB Keyboard as /class/input/input2
generic-usb 0003:1C4F:0002.0002: input: USB HID v1.10 Keyboard [USB USB Keyboard] on usb-s3c24xx-1/input0
drivers/hid/usbhid/hid-core.c: usb_submit_urb(ctrl) failed
generic-usb 0003:1C4F:0002.0003: timeout initializing reports
input: USB USB Keyboard as /class/input/input3
1.10 Device [USB USB Keyboard] on usb-s3c24xx-1/input1
```

USB Disk

Just like the SD card, USB disk will be automatically mounted to the directory “/udisk” as soon as it is connected:

```
usb 1-1: new full speed USB device using s3c2410-ohci and address 2
usb 1-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
scsi 0:0:0:0: Direct-Access    USB 2.0  SD MMC Reader          PQ: 0 ANSI: 0 CCS

sd 0:0:0:0: [sda] 3842048 512-byte hardware sectors: (1.96 GB/1.83 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 3842048 512-byte hardware sectors: (1.96 GB/1.83 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0
```

The device file for the USB disk can be found at “/dev/udisk”, we can unmount the disk by typing the following commands:

- *umount /udisk*
- *mkdir /tmp/udisk*
- *mount /dev/udisk /tmp/udisk*

Upto 32GB USB Disk is supported by *uControl* Platform.

5.5 Network Test

Configuration

- Configure the IP address with the command *ifconfig* we can setup the IP address, for example:
 - *ifconfig eth0 192.168.1.99 up*

192.168.1.99 in the command above is the IP address, you may set it to your own one

- Configure Gateway:
 - *route delete default*
 - *route add default gw 192.168.1.12*
- Modify name server The name server is configured in the file *etc/resolv.conf*, it should be something like this in the file: *nameserver 192.168.1.11*

PING Test

We can test the network connecting by using the *ping* command, for example:

- *ping 192.168.1.100 -s 10000*

6 Writing simple applications

We can create a simple application to control the User LEDs of the *uControl* Platform board.

6.1 Create a .c file

We are going to create a led.c file and we will copy and save the following code:

```
int main(void)
{
    int on=1;

    int led;

    int fd;

    fd = open("/dev/leds",0);

    if(fd<0)
    {
        perror("open device leds");
        exit(1);
    }

    printf("leds test show.press ctrl+c to exit\n");

    while(1)
    {
        for(led=0;led<4;led++)
        {
            ioctl(fd,on,led);
            usleep(60000);
        }
        on = !on;
    }
    close(fd);
    return 0;
}
```

6.2 Cross compiling

Cross compiling means to compile a program that targets another platform, for example, to compile the `led.c` program on the PC and run it on the *uControl* Platform. Use the cross compiler “`arm-linux-gcc-4.2.2-eabi`” installed in chapter 4.1 to cross compile the “`led.c`” for *uControl* Platform.

```
— /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-gcc -o leds-arm led.c
```

The command above is executed on the Linux PC and compiles the `led.c` into executable program “`leds-arm`”, copy the resulted executable to the *uControl* Platform and execute `./leds-arm` command to run it.