In your home direction(/home/xxx):

mkdir arm

cd arm

1. Serial port

minicom:

(1) install

sudo apt-get install minicom

or:

cd ../ForStudent/apt/

sudo dpkg -i minicom\_2.5-2\_i386.deb

minicom -v

(2) configuire

sudo minicom -s

--------------------------------------------

Serial port setup

A - Serial Device : /dev/ttyUSB0

F - Hardware Flow Control : No

Save setup as dfl

Exit from Minicom

--------------------------------------------

(3) run

ls -l /dev/ttyUSB0

sudo chmod 666 /dev/ttyUSB0

minicom

switch to NOR, power up board.

minicom

Device /dev/ttyUSB0 is locked.

rm /var/lock/LCK..ttyUSB0

2. Update Board System

cd ForStudent/MiniTools-20130513/

chmod 777 MiniTools

chmod 777 start.sh

sudo ./start.sh

裸机程序:

Install to Nand Flash

superboot: images/Superboot2440.bin

2440test\_W35.bin

![A description...](data:None;base64,)

Linux:

![A description...](data:None;base64,)

mini2440用户手册.pdf:

read and test:

1.1

1.2

2.4

***2.6***

**Target**:

[root@FriendlyARM /]#

**Host**:

xfliao@xfliao:~$

3. Linux Development Enviorment

(1) Network

watch sudo ifconfig eth0 192.168.0.1

Target:

ifconfig eth0 192.168.0.230

ping 192.168.0.1

(2) TFTP

Host:

(1) install

sudo apt-get install tftpd-hpa

sudo apt-get install tftp-hpa

or:

cd ../ForStudent/apt/

sudo dpkg -r tftp

sudo dpkg -r tftpd

sudo dpkg -i tftp-hpa\_5.2-1ubuntu1\_i386.deb

sudo dpkg -i tftpd-hpa\_5.2-1ubuntu1\_i386.deb

(2) configuire

sudo vim /etc/default/tftpd-hpa

--------------------------------------------

TFTP\_DIRECTORY="**/tftpboot**"

TFTP\_ADDRESS="0.0.0.0:69"

--------------------------------------------

or:

sudo vim /etc/xinetd.d/tftp

--------------------------------------------

server\_args = -s **/tftpboot**

flags = IPv4

--------------------------------------------

cd /

sudo mkdir tftpboot

sudo chmod 777 /tftpboot/

cd /tftpboot/

vi tftptest

(3) run

sudo service tftpd-hpa restart

tftpd-hpa start/running, process 11633

Or:

sudo /usr/sbin/in.tftpd restart

netstat -anu|grep 69

udp 0 0 0.0.0.0:69 0.0.0.0:\*

**Target**:

Network is good.

ifconfig eth0 192.168.0.230

ping 192.168.0.1

tftp -r tftptest -g 192.168.0.1

cat tftptest

4. Embed Linux Development

(1) Edit code

mkdir 02

cd 02/

vi hello.c

(2) Compile

gcc hello.c -o hello1

(3) **download** & Run

cp hello1 /tftpboot/

Host:

watch sudo ifconfig eth0 192.168.0.1

sudo service tftpd-hpa restart

Target:

ifconfig eth0 192.168.0.230

ping 192.168.0.1

tftp -r hello1 -g 192.168.0.1

ls -l hello1

chmod u+x hello1

./hello1

error reason:

gcc: Compiler which runs a x86 platform, compiles a executed program runing on X86.

Cross-Compiler (arm-linux-gcc):

Compiler which runs a x86 platform, compiles a executed program runing on ARM.

cd ..

cp ../ForStudent/Src/arm-linux-gcc-4.4.3.tar.gz .

tar zxvf arm-linux-gcc-4.4.3.tar.gz

cd 4.4.3/bin/

pwd

/home/xfliao/arm/4.4.3/bin

cd ~/

pwd

/home/xfliao

vi .bashrc

-----------------------------------------

export PATH=$PATH:/home/xfliao/arm/4.4.3/bin

-----------------------------------------

exit

**New a terminal.**

-----------------------------------------

64 bit OS:

14.04:

sudo apt-get install lib32z1

sudo apt-get install lib32stdc++6

------------------------------------------

arm-linux-gcc -v

cd arm/02/

arm-linux-gcc hello.c -o hello2

file hello1

file hello2

cp hello2 /tftpboot/

**Target**:

ifconfig eth0 192.168.0.230

ping 192.168.0.1

tftp -r hello2 -g 192.168.0.1

ls -l hello2

chmod u+x hello2

./hello2

5. Embed Linux Software

(1) Boot Loader

(2) Linux Kernel

(3) Root File System

BootLoader:

Initate hardware, loader OS.

Migrate bootloader (u-boot) to board:

pwd

/home/xfliao/arm

cp ~/ForStudent/Src/bootloader.tar .

tar xvf bootloader.tar

cd bootloader/u-boot/

make distclean

make mini2440\_config

make

ls -l u-boot.bin

Fire into board using MiniTools. (Switch to NOR)

cd ForStudent/MiniTools-20130513/

sudo ./start.sh

![A description...](data:None;base64,)

Low format NAND flash

Linux BootLoader:

/home/xfliao/arm/bootloader/u-boot/u-boot.bin

Switch to S2, Restart board,Hit any key.

[u-boot@MINI2440]#

NO OS, Only U-boot.

6. U-boot Input and Output

Target:

printenv

ping 192.168.0.1

tftp 0x30000000 hello2

go 0x30000000

WRONG!!!!

vi include/common.h

void printf(const char \*fmt, ...)

vi u-boot.map

0x33f963a8 printf

in arm/02:

mkdir 2\_hello

cd 2\_hello/

vi hello.c

vi Makefile

Target:

tftp 0x30000000 hello.bin

go 0x30000000

---------------------------------------------------

sudo chmod 666 /dev/ttyUSB0

minicom

watch sudo ifconfig eth0 192.168.0.1

sudo service tftpd-hpa restart

---------------------------------------------------

cp -r 2\_hello/ 3\_hello

cd 3\_hello/

vi include/common.h

int getc(void);

vi u-boot.map

0x33f965f0 getc

Target:

ping 192.168.0.1

tftp 0x30000000 hello.bin

go 0x30000000

7. ARM underlayer hardware development(裸机程序)

(1) analysis hardware [电路图](http://cn.bing.com/images/search?q=电路图&qpvt=电路图&qpvt=电路图&FORM=IGRE), make clear how it works.

mini2440原理图.pdf

蜂鸣器

MCU GPB0 pin: output mode

Output 0: buzzer off

Output 1: buzzer on

(2)refer chipset manual, make clear how to control hardware.

S3C2440.pdf

GPIO:general purple input/output

9. I/O Ports

PORT CONFIGURATION REGISTER GPBCON

GPBCON 0x56000010

GPB0 [1:0] =01

00 = Input

10 = TOUT0

01 = Output

11 = reserved

PORT DATA REGISTER GPBDAT

GPBDAT 0x56000014

data can be written to the corresponding bit of GPBDAT

GPB0 [0]

(3)program

define two variable which point to 0x56000010 and 0x56000014

variable x, n bit set 0:

x &= ~(0x01<<n)

xxxx xxxx

~ 0010 0000

& 1101 1111

-------------------------

xx0x xxxx

variable x, n bit set 1:

x |= (0x01<<n)

xxxx xxxx

| 0010 0000

-------------------------

xx1x xxxx

(4)run & test

===LED============================================

(1)

pin: Output mode

Led 1 2 3 4

GPB 5 6 7 8

output '0', led ON

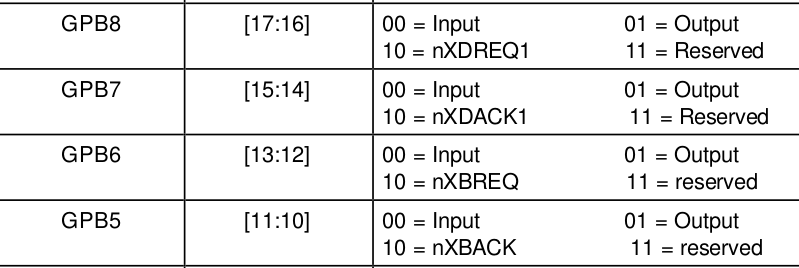
output '1', led OFF

i) how to configure GPB5-8 as output mode.

ii) how to output high/low electronic level

(2)

i)GPBCON 0x56000010



01:Output mode

ii)GPBDAT 0x56000014



GPB[5] GPB[6] GPB[7] GPB[8]

========key===================================

(1)pin: Input mode

Key 1 2 3 4 5 6

GPG 0 3 5 6 7 11

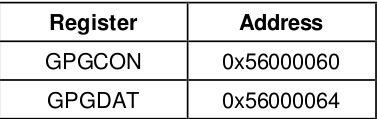
Press: low level,'0'

Not Press: High level,'1'

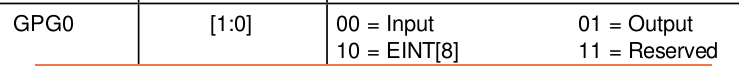
i) how to configure GPGx as input mode.

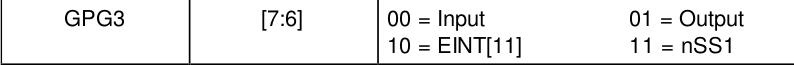
ii) MCU how to read high/low electronic level

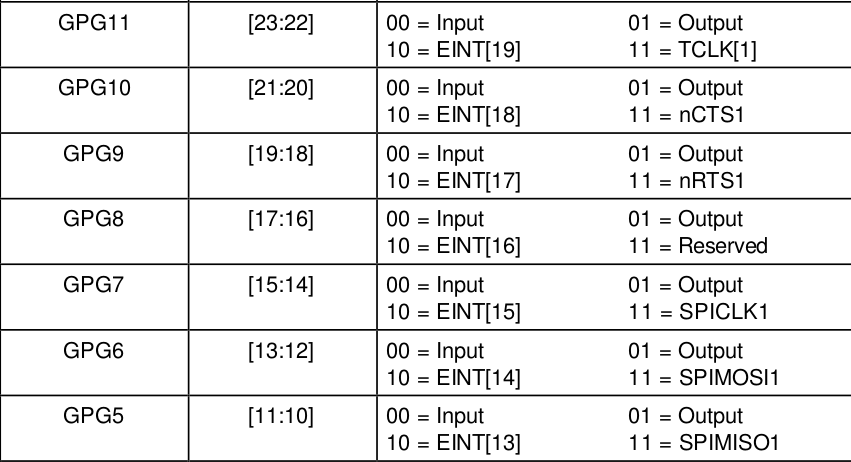
(2)



00:Input mode







When the port is configured as an input port, the corresponding bit is the

pin state



val, judge n bit 0 or 1?

val & (0x01<<n)

result bit

not 0 1

0 0

=========================================

8. Interrupt

Key 1 2 3 4 5 6

EINT 8 11 13 14 15 19

GPG 0 3 5 6 7 11

EINT8\_23: External interrupt 8 – 23

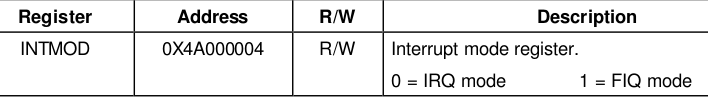
(1) SRCPND: (auto set 1, you must clear it after interruption handler)

A description...

A description...

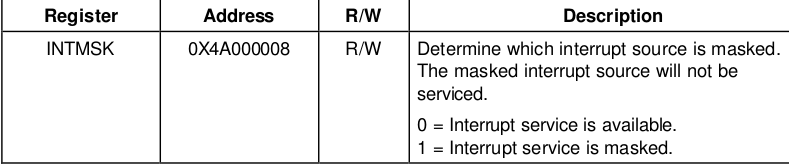
bit 5: 1, clear it: SRCPND |= 0x01<<5

(2) INTMOD: configure bit 5 to 0.



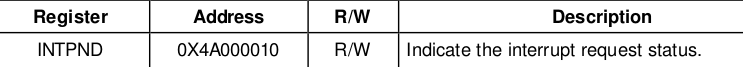
A description...

(3) INTMSK: configure bit 5 to 0.



A description...

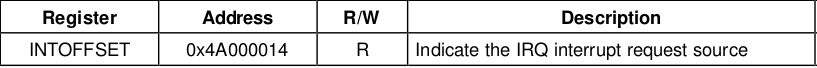
(4) INTPND (auto set 1, you must clear it after interruption handler))



A description...

After interrupt handler, clear bit 5: INTPND |= 0x01<<5

(5)INTOFFSET(euqal to 5)



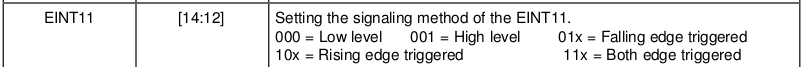
A description...

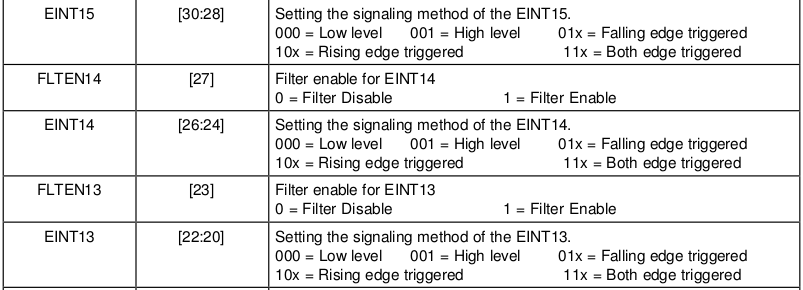
External Interuption 8-23 configuration.

(1) EXTINT1 (set 000: low level mode)

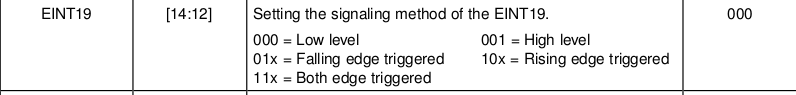
A description...

A description...



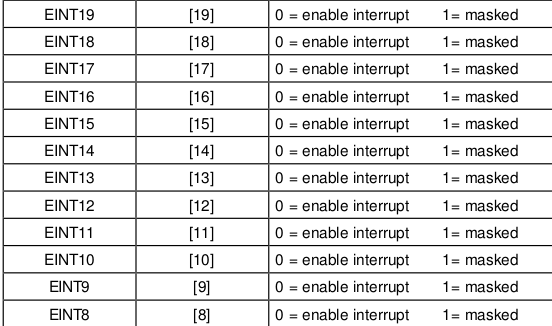


EXTINT2



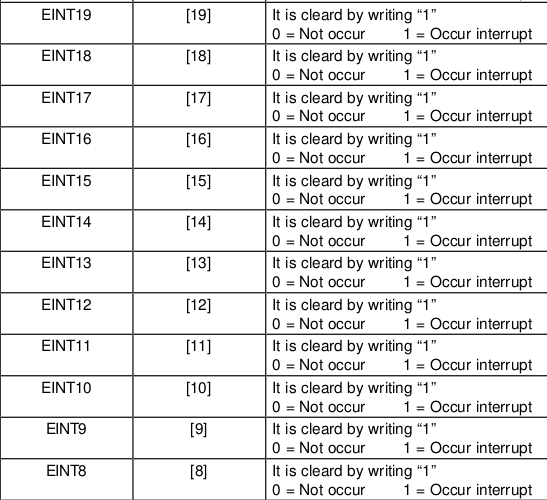
(2) EINTMASK (set to 0)

A description...



(3) EINTPEND (Auto set to 1, you must clear by |=0x01<<n)

A description...



(4)GPGCON (set to 10 as EINT mode)

A description...



CPSR 的第七位的比特为总的控制开关,1 不允许中断,0 允许中断