Control the GPIO in Linux system

Using sunxi-gpio kernel module

1. Make sure that the newest firmware has loaded the module gpio-sunxi.ko by default and it has added the command include fex2bin bin2fex. (Use the command 'lsmod' to see the module in the list.)

2. Modify script.bin that configuration file of AZD-A7N-1GB.

- modify script.bin

```
   $ mount /dev/nanda /mnt
   $ cd  /mnt
   $ bin2fex script.bin script.fex
   $ vi script.fex
```

Default GPIO paragraph setup:

```
[gpio_para]
gpio_used =1
gpio_num = 2
gpio_pin_1 = port: PH20<1><default><default><1>      //PH20 is connecting  the
green LED
gpio_pin_2 = port: PH21<1><default><default><1>      //PH21 is connecting  the
blue LED
```

Modify GPIO port want to use e.g.:

```
[gpio_para]
gpio_used =1
gpio_num = 3
gpio_pin_1 = port: PD01<1><default><default><1>
gpio_pin_2 = port: PD02<1><default><default><1>
gpio_pin_3 = port: PD03<1><default><default><1>
```

Save the file, then

```
  $ fex2bin script.fex script.bin
  $ reboot
```

After rebooted the AZD-A7N-1GB, the new setup will be effective.

3. Operate GPIO port e.g.:

- open the GPIO PD01~03

```
  $ echo 1 > /sys/class/gpio/export
  $ echo 2 > /sys/class/gpio/export
  $ echo 3 > /sys/class/gpio/export
```

Below the /sys/class/gpio has appeared directories named gpio1_pd01, gpio2_pd02, gpio3_pd03.

- set the PD01 por as output

```
  $ cd /sys/class/gpio/gpio1_pd01
  $ echo out > direction
```

- set the PD01 port as high

```
  $ echo 1 > value
```

- set the PD01 port as low

```
  $ echo 0 > value
```

Using C program without driver

The demo control the PD01 port blink led :

```
#include <stdlib.h>
#include <stdio.h>

#include "gpio_lib.h"
#define PD0    SUNXI_GPD(0)
#define PD1    SUNXI_GPD(1)
#define PD2    SUNXI_GPD(2)
#define PD3    SUNXI_GPD(3)
#define PD4    SUNXI_GPD(4)
#define MISO   SUNXI_GPE(3)
#define MOSI   SUNXI_GPE(2)
```

```
#define SCK    SUNXI_GPE(1)
#define CS     SUNXI_GPE(0)

int main()
{
    if(SETUP_OK!=sunxi_gpio_init()){
        printf("Failed to initialize GPIO\n");
        return -1;
    }

    if(SETUP_OK!=sunxi_gpio_set_cfgpin(PD01,OUTPUT)){
        printf("Failed to config GPIO pin\n");
        return -1;
    }

    int i;
    for(i=0;i<5;i++){
        if(sunxi_gpio_output(PD01,HIGH)){
            printf("Failed to set GPIO pin value\n");
            return -1;
        }

        usleep(500000);
        if(sunxi_gpio_output(PD01,LOW)){
            printf("Failed to set GPIO pin value\n");
            return -1;
        }
        usleep(500000);
    }

    sunxi_gpio_cleanup();

    return 0;

}
```

Save as gpio.c,download gpio_lib,then

```
$sudo apt-get install gcc build-essential
$tar -xf gpio.tar
$cd gpio/
$gcc gpio_lib.c -c
$gcc gpio.c -c
$gcc gpio.o gpio_lib.o -o gpio
$./gpio
```

If you have used a led connect GND and PD01 port,the led is going to blink in cycle time.
Of course you can define and use other port .
Use above the method like high-delay-low-delay to simulate PWM output .

Using mmap mapping IO address

```

////////////////////////////////////////////////////////////////////////////////
    #include <ctype.h>
    #include <string.h>
    #include <stdlib.h>
    #include <stdio.h>
    #include <math.h>
    #include <time.h>
    #include <signal.h>
    #include <sys/types.h>
    #include <sys/stat.h>
    #include <fcntl.h>
    #include <sys/mman.h>
    #include <sys/select.h>
    #include <pthread.h>
    #include <unistd.h>
    #include <sched.h>
    #include <errno.h>



    #define SW_PORTC_IO_BASE  0x01c20800
    int main() {
       unsigned int * pc;
       int fd, i;
       char * ptr;
       unsigned int addr_start, addr_offset, PageSize, PageMask, data;

       PageSize = sysconf(_SC_PAGESIZE);
       PageMask = (~(PageSize-1));
       addr_start = SW_PORTC_IO_BASE & PageMask;
       addr_offset = SW_PORTC_IO_BASE & ~PageMask;

       fd = open("/dev/mem", O_RDWR);
       if(fd < 0) {
          perror("Unable to open /dev/mem");
          return(-1);
       }

       pc = mmap(0, PageSize*2, PROT_READ|PROT_WRITE, MAP_SHARED, fd,
addr_start);

       if(pc == MAP_FAILED) {
          perror("Unable to mmap file");
          printf("pc:%lx\n", (unsigned long)pc);
          return(-1);
       }

printf("PageSize:%8.8x\tPageMask:%8.8x\naddr_start:%8.8x\taddr_offset:%8.8x\n",Pa
geSize,PageMask,addr_start,addr_offset);
       printf("pc:%8.8x\n", *(unsigned int *)pc);
       ptr = (char *)pc + addr_offset;
       data = *(unsigned int *)(ptr+0x10c);
       for(i=0;i<1000;i++){
          data |= 1<<20;                              //green led connect PH20
          *(unsigned int *)(ptr+0x10c) = data;
          usleep(100000);
```

```
            data &= ~(1<<20);
            *(unsigned int *)(ptr+0x10c) = data;
            usleep(500000);
        }

        return 0;
    }
```

save as test.c,then

```
$sudo apt-get install gcc build-essential
$gcc test.c -o test
$./test &
```

The green LED is blinking in cycle time. You can modify the `data |= 1«20;` and `data &= ~(1«20);` to use PH15 like `data |= 1«15;` and `data &= ~(1«15);`.If you have used a led connect VCC and PH15 port, the led is going to blink in cycle time.

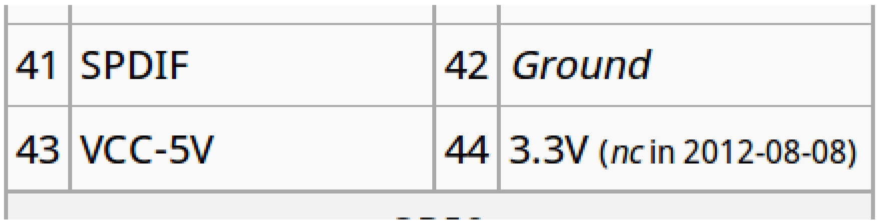
Using Python program

Install python and lib

```
$sudo apt-get update
$sudo apt-get install python-dev
$wget http://www.azdisplays.com/download/devboard/tools/pySUNXI-0.1.12.tar.gz
$tar zxf pySUNXI-0.1.12.tar.gz
$sudo python setup.py install
```

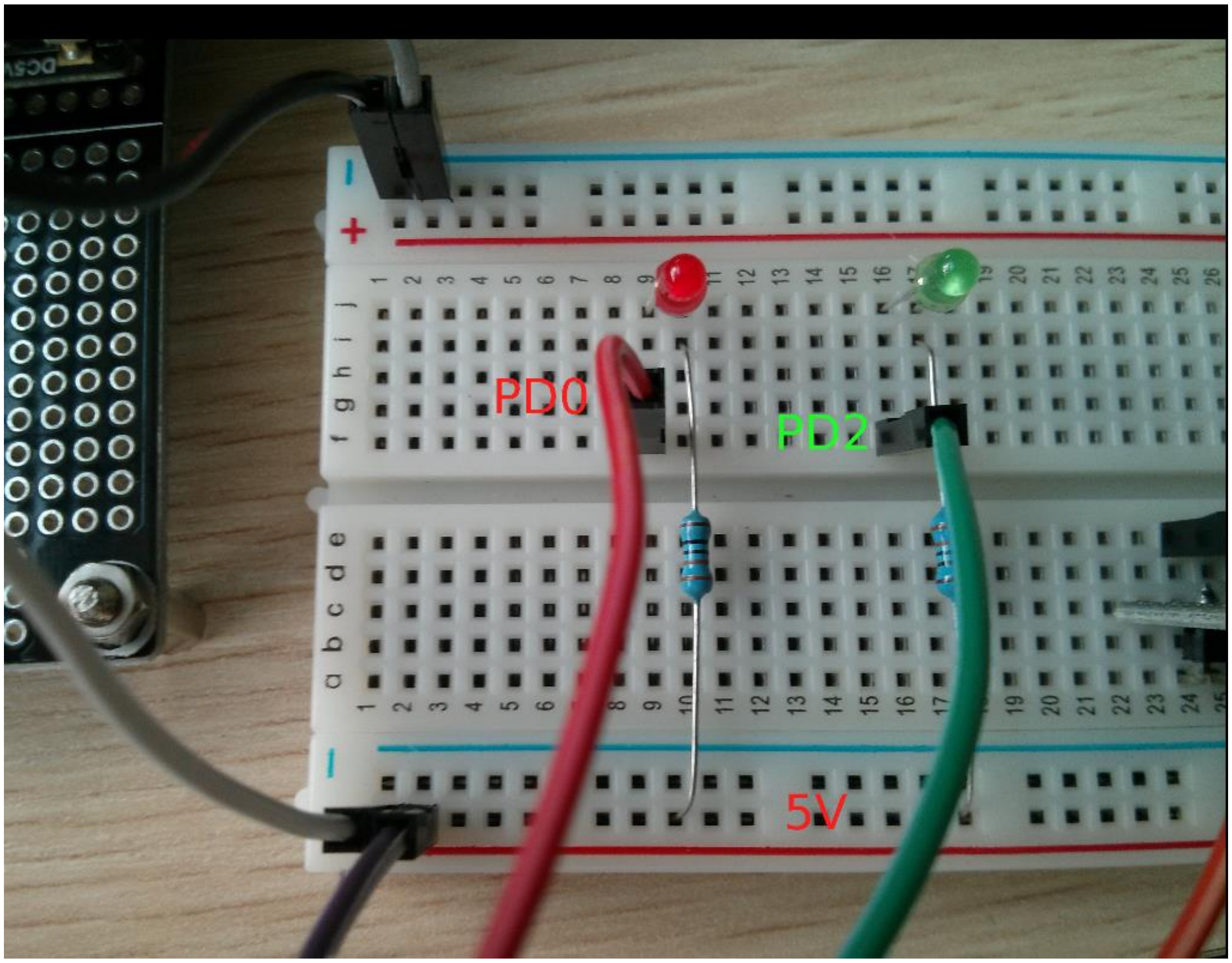
Connect the GPIO

Using the PD0 and VCC-5V for test.



41	SPDIF	42	*Ground*
43	VCC-5V	44	3.3V (*nc* in 2012-08-08)

U14 (Next to SATA connector)			
LCD			
1	PD0 (LCDD0/LVDSP0)	2	*Ground*
3	PD2 (LCDD2/LVDS0P1)	4	PD1 (LCDD1/LVDS0N0)
5	PD4 (LCDD4/LNVS0P2)	6	PD3 (LCDD3/LVDS0N1)
7	PD6 (LCDD6/LVDS0PC)	8	PD5 (LCDD5/LVDS0N2)
9	*Ground*	10	PD7 (LCDD7/LVDS0NC)



Write a sample program

```
#!/usr/bin/env python

import SUNXI_GPIO as GPIO
import time

RED_LED = GPIO.PD0

GPIO.init()
GPIO.setcfg(RED_LED, GPIO.OUT)

while True:
        GPIO.output(RED_LED, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(RED_LED, GPIO.LOW)
        time.sleep(1)
```

Run program

```
$chmod +x blink.py
$sudo ./blink.py
```

The red LED will blink in cycle time.