

BÀI 3: LÀM QUEN VỚI MÔI TRƯỜNG LINUX

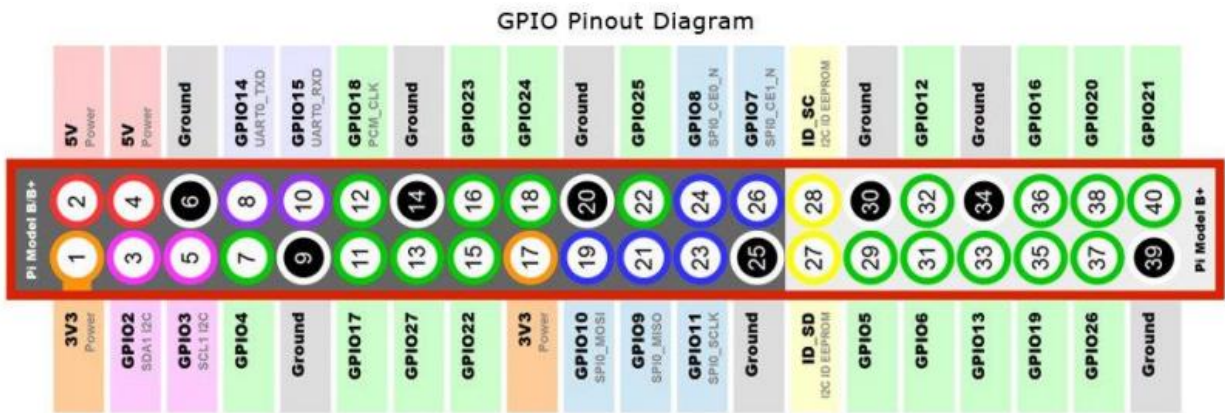
MỤC TIÊU BÀI TN

Trong bài TN này SV sẽ thực hành những phần sau:

- Viết chương trình điều khiển các cổng IO của Raspberry
- Viết chương trình sử dụng PWM, đọc tín hiệu analog
- Viết chương trình sử dụng ngắt ngoài

2. SƠ ĐỒ CHÂN GPIO CỦA RASPBERRY

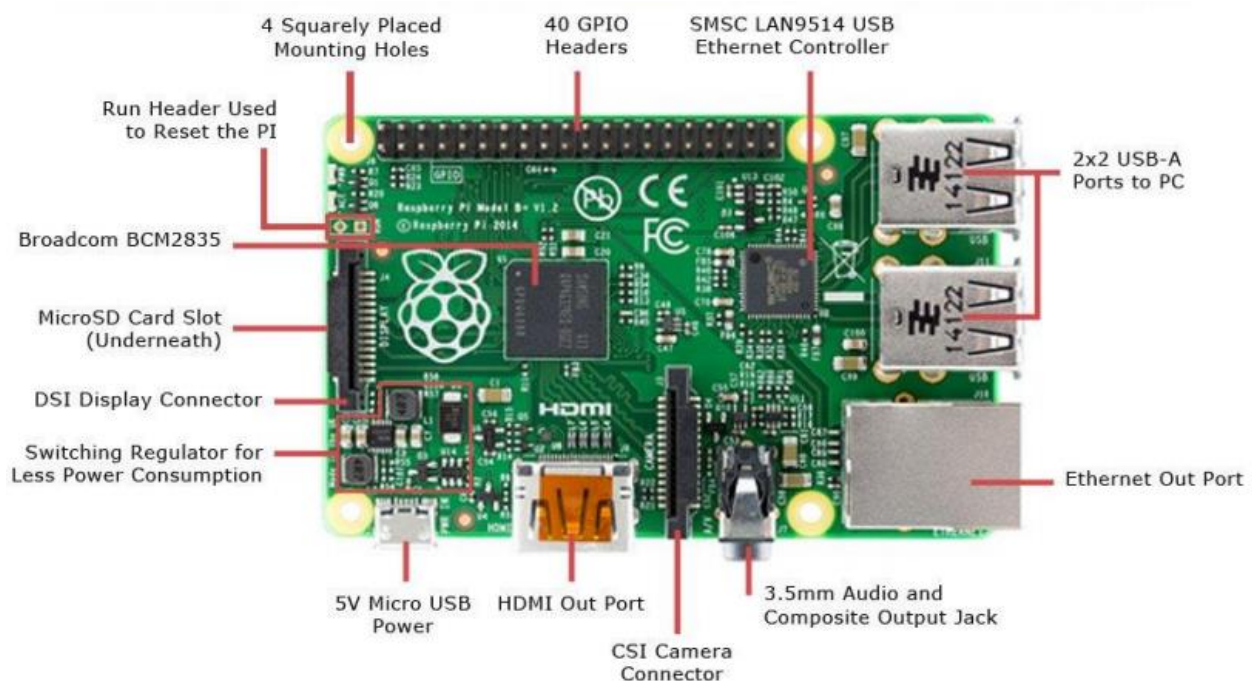
Raspberry Pi 3 có 40 chân GPIO mở rộng nằm ở cạnh trên của bo. Sơ đồ chức năng của các chân GPIO như sau:



Các chân GPIO này có thể được chia làm các nhóm:

- Chân cấp nguồn: Có 2 chân cấp nguồn 5V, 2 chân cấp nguồn 3.3V và 8 chân GND
- Chân IO: ký hiệu GPIOx, các chân này có điện áp mức cao là 3.3V và mức thấp 0V
- Các chân ngoại vi:
 - PWM (pulse-width modulation)
 - Software PWM ở tất cả các chân IO
 - Hardware PWM tại các chân GPIO12, GPIO13, GPIO18, GPIO19
- SPI: có 2 bộ SPI là SPI0 và SPI1
 - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
 - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

- I2C
 - Data: (GPIO2); Clock (GPIO3)
 - EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Serial
 - TX (GPIO14); RX (GPIO15)



3. ĐIỀU KHIỂN CHÂN GPIO VỚI WIRINGPI

3.1 Thư viện WiringPi

WiringPi là thư viện nguồn mở giúp cho việc điều khiển GPIO trên raspberry dễ dàng và thuận tiện.

Website: <http://wiringpi.com/>

WiringPi được tích hợp sẵn tron raspbian. Gõ câu lệnh **gpio -v** trên Terminal để kiểm tra xem hệ điều hành có WiringPi chưa,

```

pi@raspberrypi:~$ gpio -v
gpio version: 2.44
Copyright (c) 2012-2017 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
Type: Pi 3, Revision: 02, Memory: 1024MB, Maker: Sony
* Device tree is enabled.
*--> Raspberry Pi 3 Model B Rev 1.2
* This Raspberry Pi supports user-level GPIO access.
pi@raspberrypi:~$ █

```

Trạng thái của các chân IO có thể được kiểm tra với câu lệnh: ***gpio readall***

```

pi@raspberrypi:~$ gpio readall

```

-----Pi 3-----											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	ALT0	1	3	4		5v			
3	9	SCL.1	ALT0	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT5	TxD	15	14
		0v			9	10	1	ALT5	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19	20		0v			
9	13	MISO	ALT0	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23	24	1	OUT	CE0	10	8
		0v			25	26	1	OUT	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	

Điều khiển chân gpio trên terminal

Xuất chân gpio ra userspace bằng lệnh: ***gpio export <pin> in/out***

Thiết lập output/input cho chân IO: ***gpio mode <pin> out/in***

Xuất điện áp ra chân output: ***gpio write <pin> 1/0***

Bài tập:

- Viết code điều khiển đèn led (được kết nối với chân 13) on/off trên terminal.
- Nối 1 nút nhấn vào chân số 7, sử dụng điện trở kéo lên. Viết code sử dụng nút nhấn để on/of đèn led ở chân số 13 trên terminal

3.2 Các lệnh cơ bản của WiringPi

Tham khảo thêm tại website: <http://wiringpi.com/reference/core-functions/>

- ***pinMode (int pin, int mode)***: chọn chế độ in/out cho chân tương ứng
 - *mode*: INPUT, OUTPUT, PWM_OUTPUT
- ***pullUpDnControl (int pin, int pud)***: lựa chọn điện trở kéo xuống/lên cho chân input
 - *pud*: PUD_OFF, PUD_DOWN và PUD_UP
- ***digitalWrite (int pin, int value)***: đặt giá trị cho chân output
 - *value*: 0 hoặc 1
- ***pwmWrite (int pin, int value)***: xuất PWM ở chân output
 - *value*: 0 - 1024
- ***digitalRead (int pin)***: đọc giá trị chân input
- ***analogRead (int pin)***: đọc giá trị analog
- ***analogWrite (int pin, int value)***: xuất giá trị analog

3.2.1 Chương trình mẫu

Tạo file gpio_demo.c với nội dung như sau:

```
#include <wiringPi.h>          //gọi thư viện WiringPi

int main (void)
{
    wiringPiSetup () ;          // Khởi tạo WiringPi
    pinMode (0, OUTPUT) ;      // Thiết lập GPIO0 là output
    for (;;)
    {
        digitalWrite (0, HIGH) ; delay (500) ;
        digitalWrite (0, LOW) ; delay (500) ;
    }
    return 0 ;
}
```

Code bên trên sẽ điều khiển chân GPIO0 (chân 11 trên header) chớp nháy mỗi 500ms.

Để biên dịch chương trình trên, gõ lệnh:

gcc -Wall -o gpio_demo gpio_demo.c -lwiringPi

-Wall : tất cả warning sẽ được in lên terminal

chạy chương trình bằng lệnh:

sudo ./gpio_demo

3.3 Bài tập trên lớp

1. Viết code điều khiển đèn led (được kết nối với chân 13) on off với tần số 10Hz.
2. Nối 1 nút nhấn vào chân số 7, sử dụng điện trở kéo lên. Viết code sử dụng nút nhấn để on of đèn led ở chân số 13
3. Nối 2 nút nhấn lần lượt vào chân 7 và 11. Viết code để hai nút nhấn này có thể tăng giảm độ sáng của đèn led ở chân số 13.

4. EXTERNAL INTERRUPT

4.1 Ngắt ngoài của Raspberry

Tất cả các chân GPIO của Raspberry đều hỗ trợ ngắt ngoài. Việc cấu hình và xây dựng hàm xử lý ngắt có thể thực hiện với thư viện wiringPi thông qua các bước sau:

Bước 1: Khởi tạo thư viện wiringPi với hàm wiringPiSetup()

Bước 2: Khai báo ngắt với hàm wiringPiISR(pin, edgeType, interrupt_function)

Bước 3: Viết hàm xử lý ngắt.

Cấu trúc cơ bản của một chương trình ngắt:

```
void interrupt_func_name (void)
{
    // your code
}
int main(void)
{
    wiringPiSetup();
    wiringPiISR(pin, edge_type, &interrupt_func_name);
}
```

Các điểm cần lưu ý:

- Raspberry có thể phát hiện xung cạnh lên, cạnh xuống hay cả cạnh lên và cạnh xuống tùy thuộc vào giá trị của edge_type trong hàm wiringPiISR
 - INT_EDGE_FALLING: Phát hiện xung cạnh xuống
 - INT_EDGE_RISING: Phát hiện xung cạnh lên
 - INT_EDGE_BOTH: phát hiện cả cạnh lên và cạnh xuống
- Tên của hàm ngắt có thể đặt tùy ý, nhưng phải giống với khai báo trong hàm wiringPiISR
- Trước tên hàm ngắt trong hàm wiringPiISR phải có dấu “&”

4.2 Chương trình mẫu

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#define BUTTON_PIN 7
int counter = 0;
// Hàm xử lý ngắt
void myInterrupt(void) {
    counter++;
}

int main(void) {
    // Khởi tạo thư viện wiringPi
    wiringPiSetup ();
    // Khai báo ngắt ngoài tại chân số 7, phát hiện cạnh xuống
    wiringPiISR (BUTTON_PIN, INT_EDGE_FALLING, &myInterrupt);

    while ( 1 ) {

    }

    return 0;
}
```

4.3 Bài tập

Sử dụng ngắt thực hiện lại bài tập 2 trong mục 3.3