

라즈베리파이를 이용한 스마트센서 제어 실습1

한백전자 기술연구소



HANBACK ELECTRONICS CO.,LTD



학습5

라즈베리파이를 이용한 스마트센서 제어 실습1



- WiringPi
- GPIO Pi Map
- LED 제어 실습하기

1. WiringPi

- Raspberry Pi의 GPIO를 제어하기 위해서 C, C++, C#, Python, Perl, Ruby, Java 등 다양한 언어들이 사용된다. 이 언어들 중에서 C 언어를 사용하여 GPIO 제어 시 wiringPi 라이브러리를 사용한다면 GPIO를 쉽게 제어할 수 있다. wiringPi에 구현된 함수는 Arduino에서 사용하는 함수와 대부분이 동일하게 되어 있어서 Arduino를 사용해본 사용자라면 친숙할 것이다.

Wiring Pi

GPIO Interface library for the Raspberry Pi





1. GPIO Pi Map

- 라즈베리 파이에 설치되어 있는 WiringPi는 실제 라즈베리파이의 핀 맵과는 번호가 다르다. 이를 확인하기 위해서는 라즈베리 파이에서 “gpio readall”을 입력하면 된다.

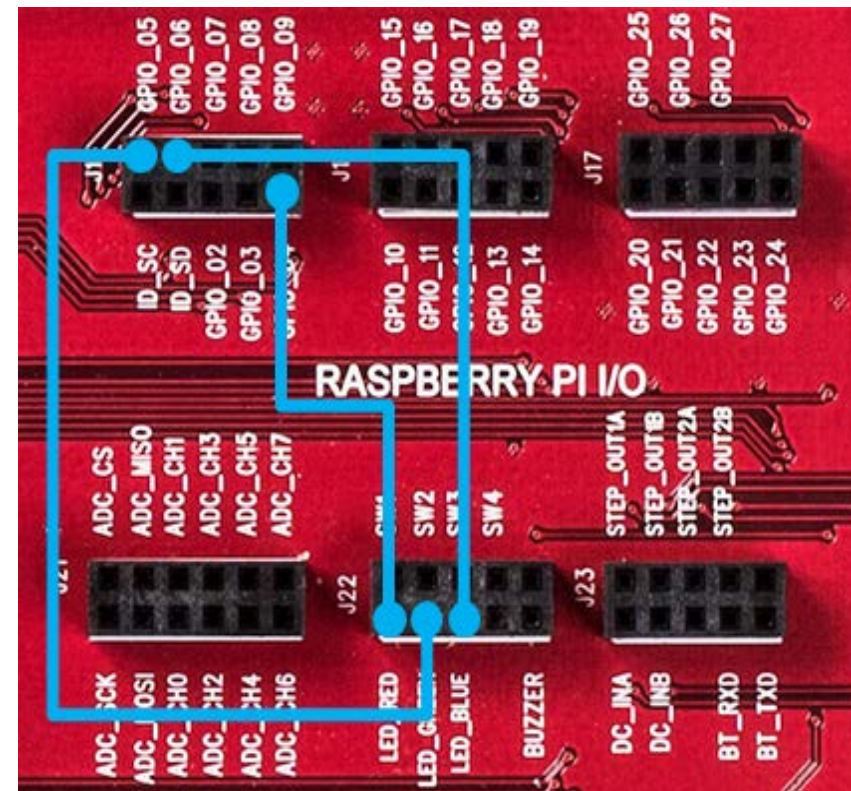
```
pi@raspberrypi ~ $ gpio readall
```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
BCM	wPi	Name	Mode	V	B Plus		V	Mode	Name	wPi	BCM
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											
		3.3v			1	2			5v		
2	8	SDA.1	IN	1	3	4			5V		
3	9	SCL.1	IN	1	5	6			0v		
4	7	GPIO. 7	IN	1	7	8	1	ALT0	TxD	15	14
		0v			9	10	1	ALT0	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	IN	0	19	20			0v		
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	1	IN	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	B Plus		V	Mode	Name	wPi	BCM
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+											

1. LED 핀 연결

- 두 모듈을 연결하기 위해서는 케이블로 그림과 같이 J4커넥터 와 J15커넥터를 케이블을 통해 연결해야 한다. 라즈베리파이의 4,5,6핀을 각각 RED, GREEN, BLUE 에 각각 연결한다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	LED 모듈 핀 번호
4	7	GPIO	LED_RED
5	21	GPIO	LED_GREEN
6	22	GPIO	LED_BLUE





1.LED 제어 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#define LED_RED 7
#define LED_GREEN 21
#define LED_BLUE 22
int main(void){
    if(wiringPiSetup () == -1)
        return 1;
    pinMode(LED_RED,OUTPUT);
    pinMode(LED_GREEN,OUTPUT);
    pinMode(LED_BLUE,OUTPUT);
```

```
digitalWrite(LED_RED,0);

    digitalWrite(LED_GREEN,0);
    digitalWrite(LED_BLUE,0);
    printf("3 Color LED Control Start !! \n");
    for(i=0;i<20;i++){
        printf("Red LED On !! \n");
        digitalWrite(LED_RED,1);
        usleep(500000);
        printf("Red LED Off !! \nGreen LED On !!\n");
        digitalWrite(LED_RED,0);
```



```
        digitalWrite(LED_GREEN,1);  
        usleep(500000);  
printf("Green LED Off !! \nBlue LED On !!\n");  
        digitalWrite(LED_GREEN,0);  
        digitalWrite(LED_BLUE,1);  
);  
        usleep(500000);  
printf("Blue LED Off !! \n");  
        digitalWrite(LED_BLUE,0);  
}  
return 0;  
}
```




1. 프로그램 컴파일 (in Raspberry Pi)

- 작성 한 소스코드를 실행 가능하도록 컴파일 한다.

```
pi@raspberrypi:~/led $ make
cc -O2 -c -o main.o main.c
cc -o main main.o -lwiringPi
pi@raspberrypi:~/led $ ls
main main.c main.o Makefile
```

2. 프로그램 실행 (in Raspberry Pi)

- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행하고 LED 모듈을 확인하면 RED, GREEN, BLUE 순서대로 LED를 각각 0.5 초간 켜고 끄는 동작을 확인할 수 있다.

```
pi@raspberrypi:~/led $ sudo ./main
3 Color LED Control Start !!
Red LED On !!
Red LED Off !!
Green LED On !!
Green LED Off !!
Blue LED On !!
```