라즈베리파이를 이용한 스마트센서 제어 실습2

한백전자 기술연구소

HANBACK ELECTRONICS CO.,LTD

학습5

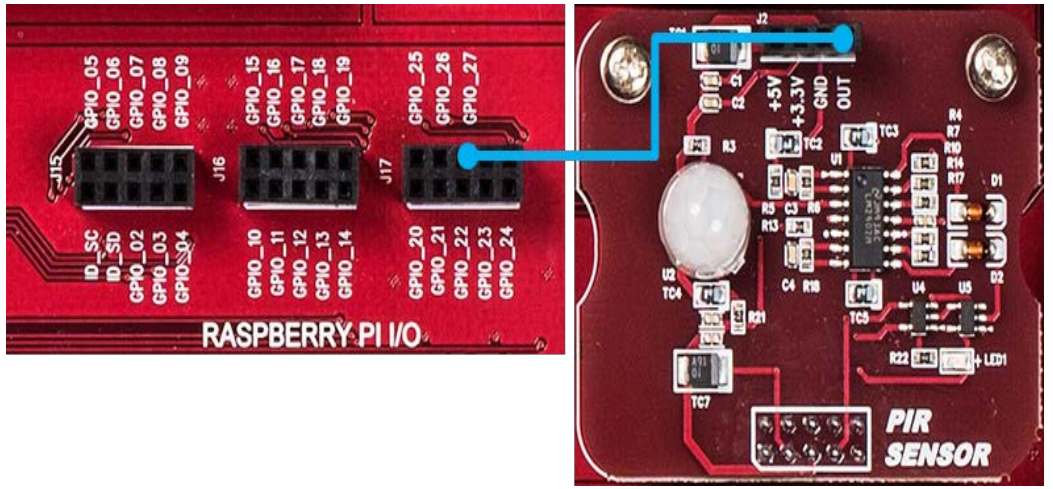
라즈베리파이를 이용한 스마트센서 제어 실습2

목차

1. 인체 감지 센서 핀 연결

- 두 모듈을 연결하기 위해서는 케이블로 그림과 같이
  인체 감지 센서 모듈 과 라즈베리파이를 케이블로 연결한다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	PIR 모듈 핀 번호
27	2	GPIO	PIR_D



1.인체 감지 센서 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#define PIR_D 2 // 인체 감지 센서를 2번에 연결
char pir_flag = 0; // 센서감지를 나타내는 flag
void PIR_interrupt()
{
        pir_flag = 1 ;
}
int main(void){

        if(wiringPiSetup()==-1)
         return -1;
```

```
pinMode(PIR_D,INPUT);
        wiringPiISR(PIR_D,
INT_EDGE_RISING, &PIR_interrupt);

while(1){
        if(pir_flag == 1) {
        printf("PIR Detected !! ₩n");

        pir_flag = 0;
         }
        else {printf("PIR Not detect !! ₩
n");
         }
        usleep(500000);          }
        return 0;
}
```

1. 프로그램 컴파일 (in Raspberry Pi)

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

2. 프로그램 실행 (in Raspberry Pi)

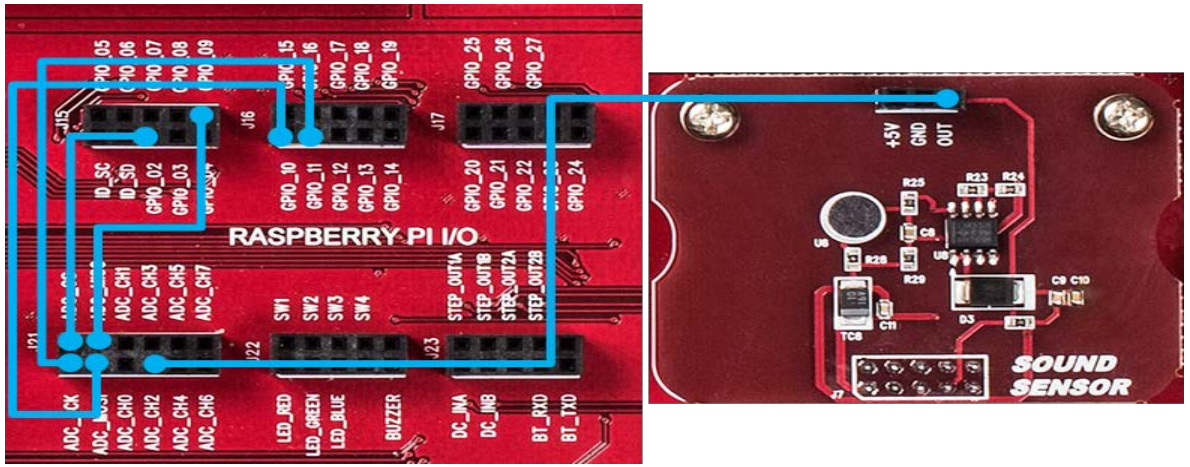
- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행한다.
  인체 감지 센서의 상태에 따라 출력되는 데이터가 다르게 표기된다.
  “PIR Detected”가 표기되면 인체 감지 센서에 인체가 감지된 상태이고
  “PIR Not detect”가 표기되면 인체 감지 센서에 어떠한 것도 감지되지 않은 상태이다.

```
pi@raspberrypi:~/pir $ sudo ./main
PIR Not detect !!
PIR Detected !!
PIR Not detect !!
PIR Detected !!
PIR Not detect !!
```

1. 소리 감지 센서 핀 연결

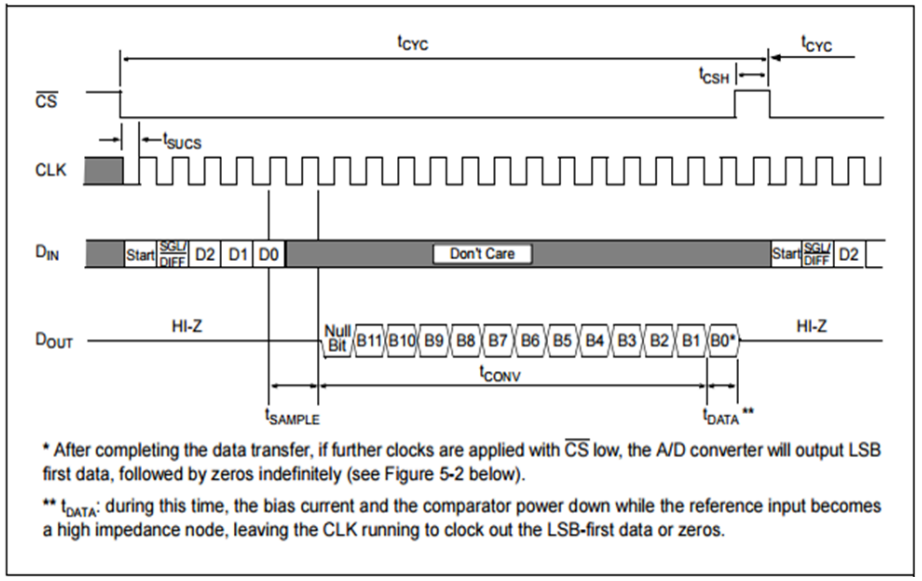
- 라즈베리파이 에는 ADC 기능을 지원하는 핀이 없다. 따라서 외부 ADC 칩을 통해 소리 감지 센서의 센서 값을 읽어 온다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	MCP3208 핀 명칭
2	8	GPIO	ADC_CS
9	13	MISO	ADC_MISO
10	12	MOSI	ADC_MOSI
11	14	SCLK	ADC_SCK



1. MCP3208

- MCP3208은 아날로그 신호를 디지털 신호로 변환해주는 칩이다.
  총 8개의 채널을 지원하고 12 bit의 해상도를 지원한다.
  라즈베리파이에서 제어는 SPI를 통해서 제어한다.



< MCP3208 타이밍도 >

1. 소리 감지 센서 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#define SPI_CH 0 // SPI 채널
#define ADC_CH2 2 // AD 변환기 채널
#define ADC_CS 8
#define SPI_SPEED 500000 // SPI Speed
int main(void){
int adcValue=0, i;
char adChannel = ADC_CH2;
```

```
unsigned char buf[3];

if(wiringPiSetup () == -1)
                return 1;
pinMode(ADC_CS,OUTPUT);

if(wiringPiSPISetup(SPI_CH,SPI_SPEED) =
= -1){
printf("wiringPi SPI Setup failed!₩n");
exit(0);
}
```

소리 감지 센서 프로그램 작성


```
while(1){
buf[0] = 0x06 | ((adChannel & 0x07)>>2
);
buf[1] = ((adChannel & 0x07)<<6);
buf[2] = 0x00;
digitalWrite(ADC_CS,0);
wiringPiSPIDataRW(SPI_CH,buf,3);
buf[1] = 0x0F & buf[1];
adcValue = (buf[1] << 8) | buf[2];
digitalWrite(ADC_CS,1);
printf("Sound ADC Value -> %d₩n",adc
Value);
usleep(100000);
}
return 0;
}
```

결과 확인

1. 프로그램 컴파일 (in Raspberry Pi)

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

2. 프로그램 실행 (in Raspberry Pi)

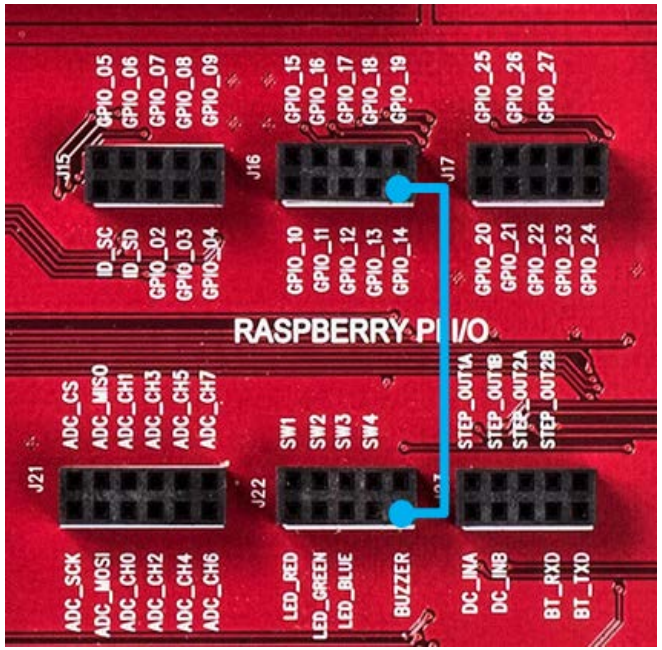
- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행한다. 소리 감지 센서에 소리가 감지되면 그림과 같이 출력되는 수치가 크게 변화 하는 것을 확인 할 수 있다.

```
pi@raspberrypi:~/sound $ sudo ./main
Sound ADC Value -> 176
Sound ADC Value -> 134
Sound ADC Value -> 144
Sound ADC Value -> 154
Sound ADC Value -> 148
Sound ADC Value -> 142
Sound ADC Value -> 2890
Sound ADC Value -> 2930
Sound ADC Value -> 2487
Sound ADC Value -> 2923
Sound ADC Value -> 1472
Sound ADC Value -> 665
Sound ADC Value -> 388
Sound ADC Value -> 277
Sound ADC Value -> 186
Sound ADC Value -> 183
Sound ADC Value -> 195
Sound ADC Value -> 165
Sound ADC Value -> 140
Sound ADC Value -> 145
Sound ADC Value -> 126
Sound ADC Value -> 128
```

1. Buzzer 핀 연결

- 두 모듈을 연결하기 위해서는 케이블로 그림과 같이 Buzzer 모듈을 라즈베리파이와 연결한다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	Buzzer 모듈 핀 번호
14	15	GPIO	Buzzer



1. Buzzer 제어 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#define BUZZER 15
int main(void){
if(wiringPiSetup () == -1)
return 1;
```

```
pinMode(BUZZER,OUTPUT);
printf("Buzzer Control Start !! ₩n");
while(1){
printf("BUZZER ON !!!₩n");
digitalWrite(BUZZER,1);
sleep(1); // 1초시간 지연
printf("BUZZER OFF !!!₩n");
digitalWrite(BUZZER,0);
sleep(1); // 1초 시간 지연
                }
                return 0;
}
```

1. 프로그램 컴파일 (in Raspberry Pi)

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

2. 프로그램 실행 (in Raspberry Pi)

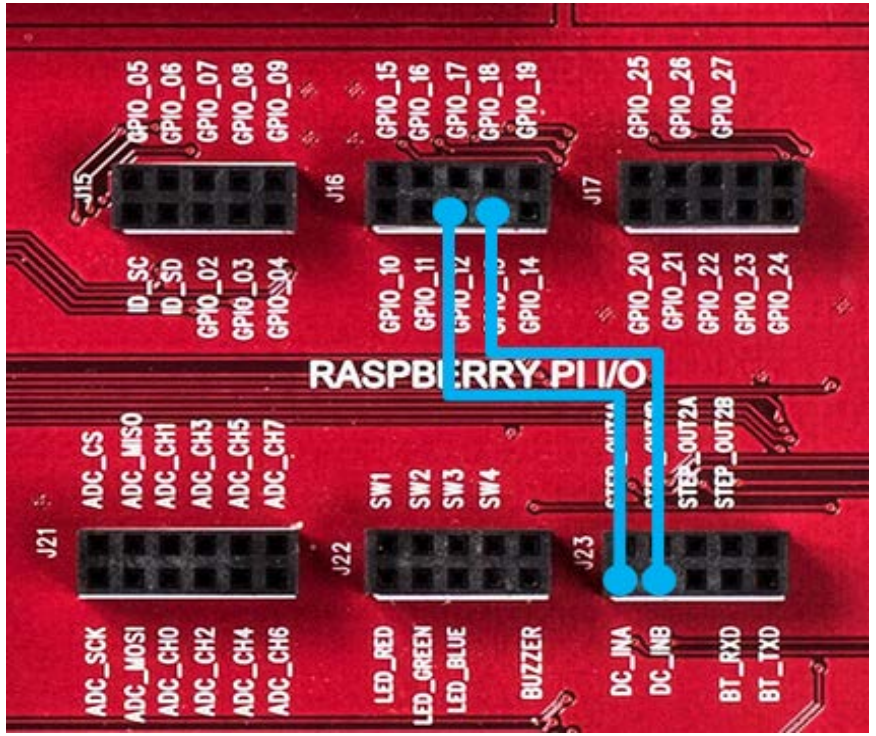
- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행한다. 1초 마다 Buzzer를 켜고 끈다. 총 20초간 동작하게 되고 소리를 울릴 때는 "BUZZER ON"을 소리 울림을 정지할 때는 “BUZZER OFF"를 출력한다.

```
pi@raspberrypi:~/buzzer $ sudo ./main
Buzzer Control Start !!
BUZZER ON !!!
BUZZER OFF !!!
BUZZER ON !!!
BUZZER OFF !!!
BUZZER ON !!!
```

1. DC Motor 핀 연결

- 두 모듈을 연결하기 위해서는 케이블로 그림과 같이 DC Motor 과 라즈베리파이를 케이블로 연결한다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	DC 모터 핀 번호
12	26	GPIO	DC_INA
13	23	GPIO	DC_INB



1. DC Motor 제어 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#define DC_INA 26
#define DC_INB 23
int main(void){
int direction,i;
                if(wiringPiSetup () == -1)
                                return 1;
                pinMode(DC_INA,OUTPUT);
                pinMode(DC_INB,OUTPUT);
                digitalWrite(DC_INA,0);
                digitalWrite(DC_INB,0);
                printf("DC motor Control Start !!
₩n");
                while(1){
                printf("1. forward, 2. reverse 3.exit
₩n select : ");
                scanf("%d",&direction);
                                if(direction==1){

                printf("forward Selected !!₩n");
```

```
                for(i=0; i<5; i++){

                digitalWrite(DC_INB,1);
                sleep(1);
                digitalWrite(DC_INB,0);
                sleep(1);
                }
                }else if(direction==2){
                printf("reverse Selected !!₩n");
                for(i=0; i<5; i++){
                digitalWrite(DC_INA,1);
                sleep(1);
                digitalWrite(DC_INA,0);
                sleep(1);
                }
                }else if(direction==3){
                printf("exit Selected !!₩n");
                break;
                }
                }
                return 0;
}
```

결과 확인

1. 프로그램 컴파일 (in Raspberry Pi)

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

2. 프로그램 실행 (in Raspberry Pi)

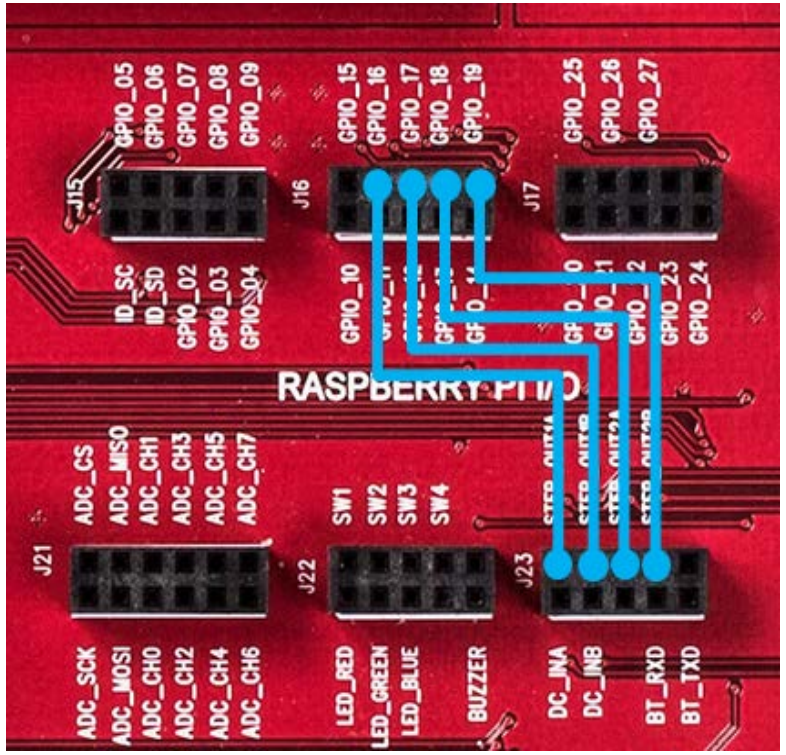
- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행한다.
  사용자가 모터를 회전 시키고 싶은 방향에 따라 '1' 또는 '2'를
  입력하면 해당 방향으로 모터가 회전하게 된다.

```
pi@raspberrypi:~/dc_motor $ sudo ./main
DC motor Control Start !!
1. forward, 2. reverse 3.exit
 select : 1
forward Selected !!
Motor Encode -> 1
1. forward, 2. reverse 3.exit
 select : 2
reverse Selected !!
Motor Encode -> 2
Motor Encode -> 3
1. forward, 2. reverse 3.exit
 select : 3
exit Selected !!
```

1. Step Motor 핀 연결

- 두 모듈을 연결하기 위해서는 케이블로 그림과 같이 스텝 모터 모듈과 연결한다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	Step 모터 핀 번호
16	27	GPIO	STEP_OUT1A
17	0	GPIO	STEP_OUT1B
18	1	GPIO	STEP_OUT2A
19	24	GPIO	STEP_OUT2B



Step Motor 제어 프로그램 작성

1. Step Motor 제어 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#define STEP_OUTA 27
#define STEP_OUTB 0
#define STEP_OUT2A 1
#define STEP_OUT2B 24
int main(void){
int i;
                if(wiringPiSetup () == -1)
                                return 1;
                pinMode(STEP_OUTA,OUTPUT);
                pinMode(STEP_OUTB,OUTPUT);
                pinMode(STEP_OUT2A,OUTPUT);
                pinMode(STEP_OUT2B,OUTPUT);


digitalWrite(STEP_OUTA,0);
digitalWrite(STEP_OUTB,0);
digitalWrite(STEP_OUT2A,0);
digitalWrite(STEP_OUT2B,0);
printf("Step Motor Control Start !! ₩n");
                for(i=0;i<2000;i++){
digitalWrite(STEP_OUTA,1);
usleep(2000);
```

```
digitalWrite(STEP_OUTA,0);
digitalWrite(STEP_OUTB,1);
usleep(2000);
digitalWrite(STEP_OUTB,0);
digitalWrite(STEP_OUT2A,1);
usleep(2000);
digitalWrite(STEP_OUT2A,0);
digitalWrite(STEP_OUT2B,1);
usleep(2000);
digitalWrite(STEP_OUT2B,0);
}
return 0;
}
```

1. 프로그램 컴파일 (in Raspberry Pi)

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

2.프로그램 실행 (in Raspberry Pi)

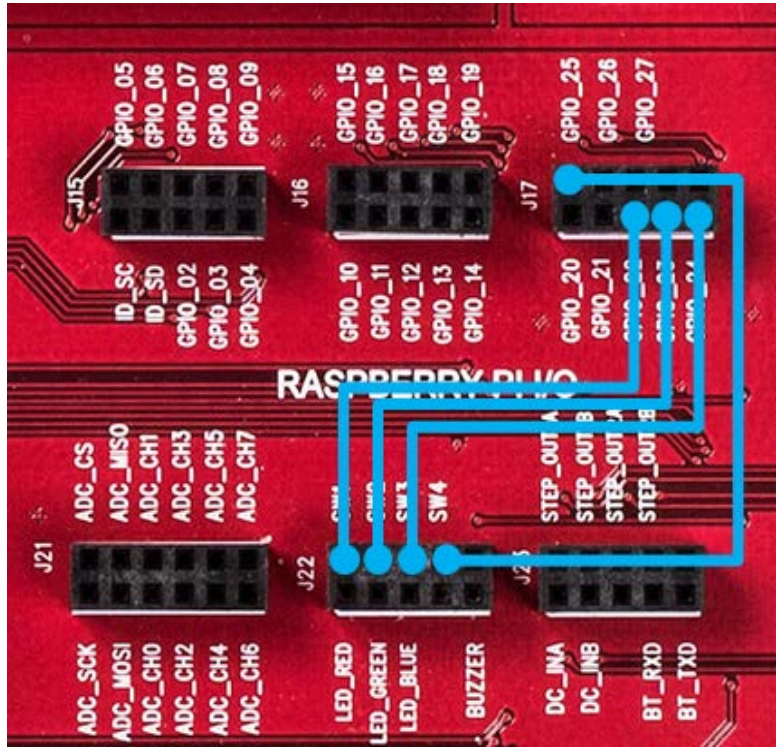
- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행한다.
  프로그램이 실행되면 Step 모터가 한 방향으로 회전하는 것을 확인 할 수 있다.


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

1. Input 모듈 핀 연결

- 두 모듈을 연결하기 위해서는 케이블로 아래 그림과 같이 버튼의 핀 연결을 수행한다.

라즈베리파이 핀 번호	Wiring Pi 핀 번호	핀 정보	Input shield 핀 번호
22	3	GPIO	SW1
23	4	GPIO	SW2
24	5	GPIO	SW3
25	6	GPIO	SW4



1. Input shield 프로그램 작성

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#define SW1 3
#define SW2 4
#define SW3 5
#define SW4 6
int main(void){
        int i,ret=2;
        if(wiringPiSetup () == -1)
                return 1;
        pinMode(SW1,INPUT);
        pinMode(SW2,INPUT);
        pinMode(SW3,INPUT);
        pinMode(SW4,INPUT);
        for(i=0;i<20;i++){
        ret = digitalRead(SW1);
        ret = digitalRead(SW1);
        if(ret==0)
        printf("SW1 Button push !!₩n");
        ret = digitalRead(SW2);
        if(ret==0)
        printf("SW2 Button push !!₩n");
        ret = digitalRead(SW3);
        if(ret==0)
        printf("SW3 Button push !!₩n");
        ret = digitalRead(SW4);
        if(ret==0)
        printf("SW4 Button push !!₩n");
        sleep(1);
        }
        return 0;
}
```

결과 확인

1. 프로그램 컴파일 (in Raspberry Pi)

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

2. 프로그램 실행 (in Raspberry Pi)

- 프로그램 컴파일이 완료 되면 "./main" 명령을 통해 프로그램을 실행한다.
  프로그램이 실행되면 현재 Input shield 에 Key 의 눌림 상태를 출력한다.

```
pi@raspberrypi:~/input $ sudo ./main
SW1 Button push !!
SW2 Button push !!
SW3 Button push !!
SW4 Button push !!
SW3 Button push !!
SW2 Button push !!
```