

Raspberry Pi를 활용한 IoT 프로젝트

라즈베리 파이 GPIO와 센서 동작하기

4일차

2021.07.15 10:00~13:00

담당교수 : 조도은



1일차 : 라즈베리파이 소개와 환경 구축(3H)

2일차 : 라즈베리 파이를 위한 리눅스 기초 배우기(3H)

3일차 : 파이썬 기초 명령어 익히기(3H)

4일차 : 라즈베리 파이 GPIO와 센서 동작하기(3H)

5일차 : 나만의 가상비서 만들기(구글 어시스턴트)(3H)



- 라즈베리 파이 GPIO 및 센서 사용하기
 - GPIO 핀 구조
 - LED
 - Button
 - Button과 LED
 - Servo Motor
 - PIR(HC-SR501)



■ GPIO

- General Purpose Input Output의 약자
- 라즈베리 파이에서 **범용 입출력**으로 많이 사용
- 총 26개 포트가 있음 (GPIO 2~GPIO 27)
- 기본 입출력 외에 하나의 핀에 다른 기능의 입출력도 사용 가능
- 전원은 micro-USB 단자를 통해 5V 전원을 제공 (내부 동작은 3.3Volt기반)
- GPIO 핀 당 3.3V/50mA의 최대 허용 전원이내에서 사용할 것

GPIO 핀구조 : <https://pinout.xyz/>

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	Red	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	Blue	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Blue	Ground	06
07	GPIO04 (GPIO_GCLK)	Green	(TXD0) GPIO14	08
09	Ground	Black	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	Green	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Green	Ground	14
15	GPIO22 (GPIO_GEN3)	Green	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	Red	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Purple	Ground	20
21	GPIO09 (SPI_MISO)	Purple	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	Purple	(SPI_CE0_N) GPIO08	24
25	Ground	Black	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	Yellow	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12	32
33	GPIO13	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40



■ GPIO 라이브러리

- GPIO 사용을 위해서는 센서와 라즈베리 파이의 GPIO 핀을 연결해야 함
- 프로그램으로 센서를 제어하기 위해 라즈비안에 포함된 **RPi.GPIO** 라이브러리가 필요
- **GPIO.setmode** 함수
 - GPIO 세팅을 위해 사용하는 함수로 BCM 옵션은 “브로드컴 SOC 채널” 번호에 의한 핀 번호를 사용할 경우 세팅하는 옵션
 - **BOARD** 옵션은 그림의 물리적 구성(핀)을 핀 번호로 사용할 경우 세팅하는 옵션
 - **BCM**으로 세팅하는 경우 GPIO 4에 해당하는 핀은 BOARD로 세팅하는 경우 7번 핀에 해당

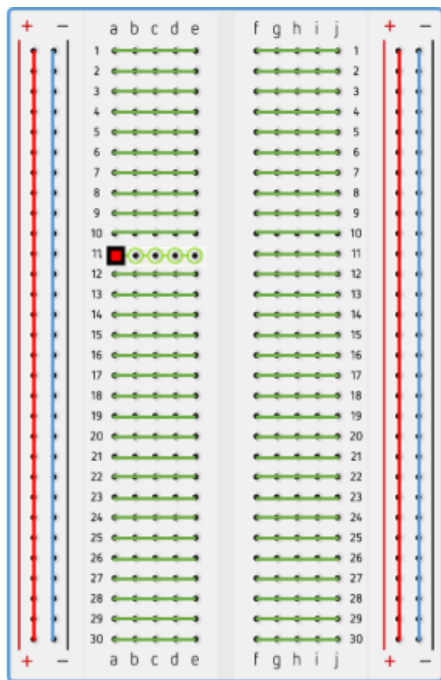
```
GPIO.setmode(GPIO.BCM) # BCM 핀번호
```

```
GPIO.setmode(GPIO.BOARD) # 물리적 핀번호
```



■ 브레드보드

- 브레드보드는 납땜하지 않고도 각종 전자부품을 쉽게 꽂아 전자회로를 구성할 수 있는 보드

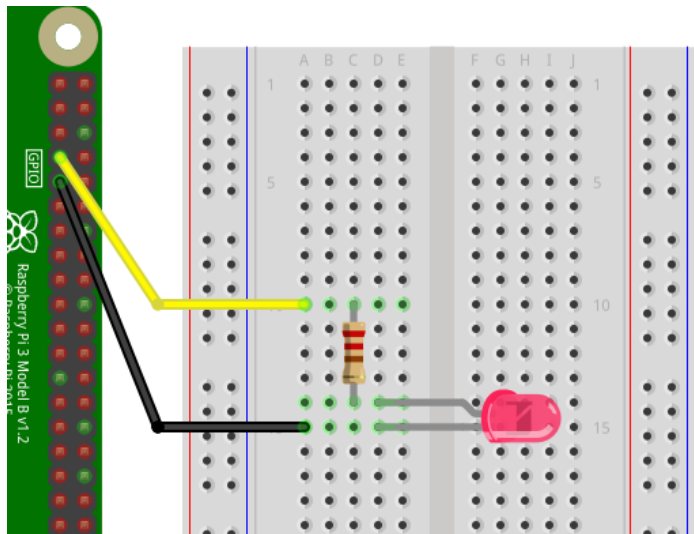




■ LED 실습

라즈베리파이의 GPIO 출력을 제어하기 위해 LED를 브레드 보드에 연결하여 ON/OFF 실습을 진행한다.

(1) 브레드 보드 연결하기



준비물 : LED × 1, 저항(220Ω) × 1, 점퍼선 × 2

라즈베리 파이	LED
GPIO 4	+ 저항
GND	-



▪ LED 실습

(2) 파이썬 코드 작성하기

(3) 코드 실행하기

LED가 10회 깜빡이는 것을
확인할 수 있다.

```
#-*-coding:utf-8-*-  
#모듈 불러오기  
import RPi.GPIO as GPIO  
import time  
  
#GPIO 핀 번호 모드를 BCM으로 설정  
GPIO.setmode(GPIO.BCM)  
  
#LED 핀번호 설정  
led_pin = 4  
  
#LED 핀 출력으로 설정화  
GPIO.setup(led_pin, GPIO.OUT)  
  
#10번 반복하여 LED on/off  
for i in range(10):  
    GPIO.output(led_pin, 1)  
    time.sleep(1)  
    GPIO.output(led_pin, 0)  
    time.sleep(1)  
  
#GPIO 설정 초기  
GPIO.cleanup()
```

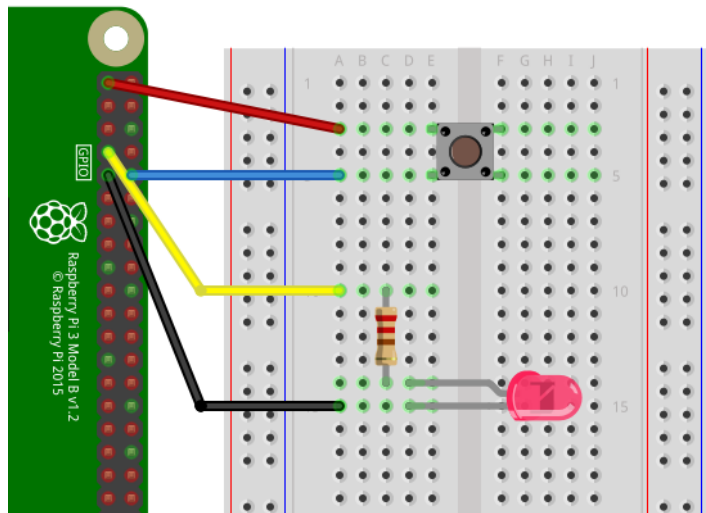
led.py



■ 푸시 버튼 입력 감지하기

푸시 버튼을 추가 연결하여 버튼을 눌렀을 때 모니터에 “Button push!” 메시지를 출력한다.

(1) 브레드보드 연결하기



준비물 : Button × 1, 점퍼선 × 2

라즈베리 파이	Button
VCC(3.3V)	버튼 연결
GPIO 15	버튼 연결



■ Button 실습

(2) 파이썬 코드 작성하기

(3) 코드 실행하기

Button을 눌렀을 때 출력창에
“Button pushed!”가 출력되는 것을
확인한다.

button.py

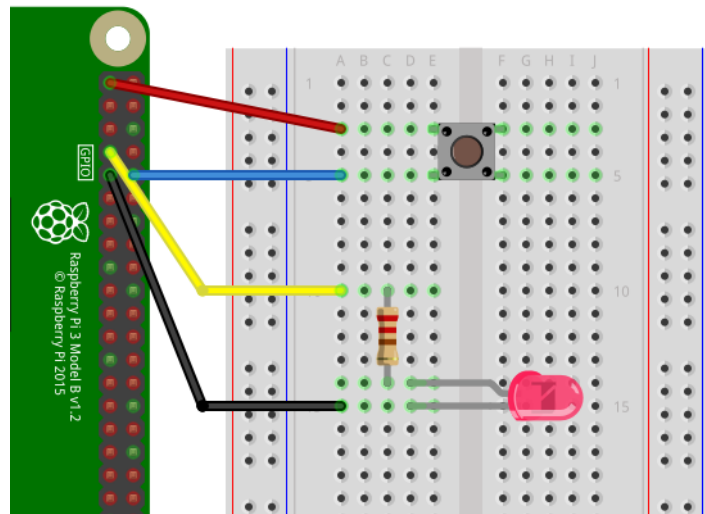
```
#-*-coding:utf-8-*-  
#모듈 불러오기  
import RPi.GPIO as GPIO  
import time  
  
#GPIO 핀 번호 모드를 BCM으로 설정  
GPIO.setmode(GPIO.BCM)  
  
#Button 핀번호 설정  
button = 15  
  
#버튼을 input으로 설정  
GPIO.setup(button, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)  
while 1:  
    if GPIO.input(button) == GPIO.HIGH:  
        print("Button pushed!")  
        time.sleep(0.1)
```



■ 푸시 버튼 입력으로 LED 켜고 끄기 실습

푸시버튼을 연결하여 Event 알림 방식으로 입력을 받아서 LED의 ON/OFF를 제어한다.

(1) 브레드 보드 연결하기



준비물 : LED × 1, 저항(220Ω) × 1, 점퍼선 × 4, Button × 1

라즈베리 파이	LED	Button
GPIO 4	+ 저항	
GND	-	
VCC(3.3V)		버튼 연결
GPIO 15		버튼 연결



■ 푸시 버튼 입력으로 LED 켜고 끄기 실습

(2) 파이썬 코드 작성하고, 버튼을 눌렀을 때 LED가 켜지는 것을 확인한다.

```
#-*-coding:utf-8-*-
```

```
#모듈 불러오기
```

```
import RPi.GPIO as GPIO
```

```
import time
```

```
#불필요한 warning 제거
```

```
GPIO.setwarnings(False)
```

```
#GPIO 핀모드 설정
```

```
GPIO.setmode(GPIO.BCM)
```

```
#사용할 GPIO핀 번호 설정
```

```
button = 15
```

```
led = 4
```

```
#버튼을 input으로 설정
```

```
GPIO.setup(button, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
```

```
#LED를 out으로 설정
```

```
GPIO.setup(led, GPIO.OUT)
```

```
light_on = False
```

```
def button_callback(channel):
```

```
    global light_on
```

```
    if light_on == False:
```

```
        GPIO.output(led, 1)
```

```
        print("LED ON!")
```

```
    else:
```

```
        GPIO.output(led, 0)
```

```
        print("LED OFF!")
```

```
    light_on = not light_on
```

```
GPIO.add_event_detect(button, GPIO.RISING,  
callback=button_callback, bouncetime=300)
```

```
while 1:
```

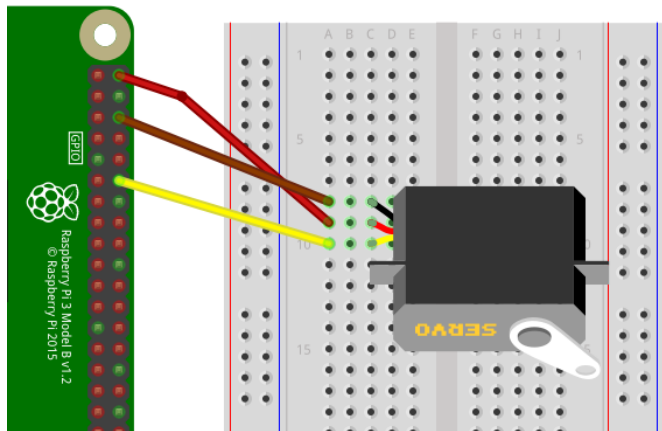
```
    time.sleep(0.1)
```



■ PWM으로 서버모터 움직이기

서버모터가 원하는 각도만큼 움직이게 한다.

(1) 브레드 보드 연결하기



준비물 : Servo × 1, 점퍼선 × 3

라즈베리 파이	Servo
VCC(5V)	빨간색
GPIO 18	오렌지
GND	갈색



■ PWM으로 서보모터 움직이기

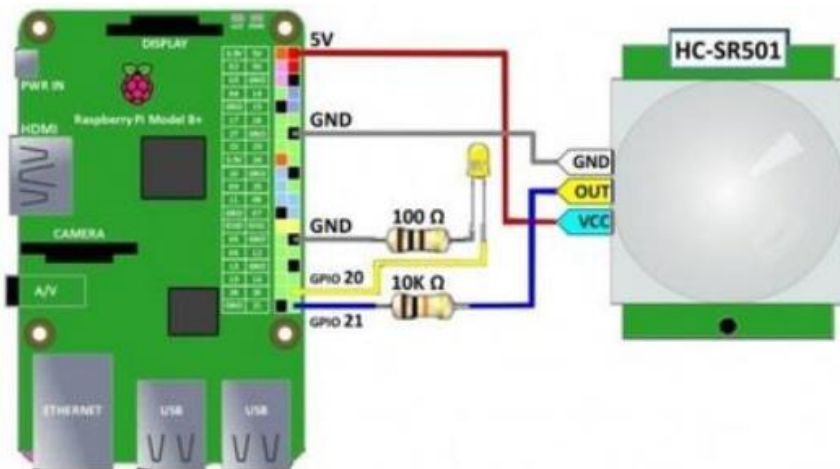
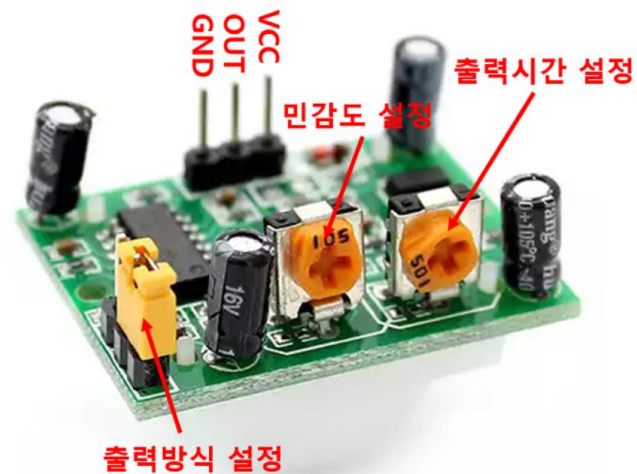
(2) 파이썬 코드 작성하고, 실행했을 때 서보 모터가 듀티를 변경하여 움직이는 것을 확인한다.

```
#-*-coding:utf-8-*-  
#모듈 불러오기  
import RPi.GPIO as GPIO  
import time  
  
#불필요한 warning 제거  
GPIO.setwarnings(False)  
  
#GPIO 핀모드 설정  
GPIO.setmode(GPIO.BCM)  
  
#사용할 GPIO핀 번호 설정  
SERVO_PIN = 18  
  
#서보 핀의 출력 설정  
GPIO.setup(SERVO_PIN, GPIO.OUT)  
  
#PWM 듀티비 0으로 시작  
servo.start(0)  
  
try:  
    while True:  
        servo.ChangeDutyCycle(7.5) #90도  
        time.sleep(1)  
        servo.ChangeDutyCycle(12.5) #180도  
        time.sleep(1)  
        servo.ChangeDutyCycle(2.5) #0도  
        time.sleep(1)  
except KeyboardInterrupt:  
    servo.stop()  
    GPIO.cleanup()
```



- 적외선 인체감지 센서(PIR)를 이용하여 LED 켜고 끄기 실습

사람의 움직임을 감지하여 적외선 변화가 있을 때 노란색 LED를 켜고, 감지되지 않을 때 빨간색 LED를 켜다.





■ 적외선 인체감지 센서(PIR)를 이용하여 LED 켜고 끄기 실습

(1) 브레드보드 연결하기

준비물 : PIR센서 × 1, 저항(220Ω) × 2, 점퍼선 × 5, LED × 2

라즈베리 파이	PIR센서	LED
VCC(5V)	VCC	
GPIO 4	OUT	
GPIO 20		LED(Red) +
GPIO 21		LED(Yellow) +
GND	GND	



■ 적외선 인체감지 센서(PIR)를 이용하여 LED 켜고 끄기 실습

(2) 파이썬 코드 작성하고, 실행했을 때 인체의 움직임을 감지하여 LED가 켜지는 것을 확인한다.

```
# -*- coding: utf-8 -*-
# 모듈 불러오기
import RPi.GPIO as GPIO
import time

# 불필요한 warning 제거
GPIO.setwarnings(False)

# GPIO 핀모드 설정
GPIO.setmode(GPIO.BCM)

# 사용할 GPIO핀 번호 설정
led_R = 20
led_Y = 21
sensor = 4

# 핀 입출력 설정
GPIO.setup(led_R, GPIO.OUT)
GPIO.setup(led_Y, GPIO.OUT)
GPIO.setup(sensor, GPIO.IN)

print("PIR Ready ....")
time.sleep(5) # 센서 준비 시간

try:
    while True:
        if GPIO.input(sensor) == 1:
            GPIO.output(led_Y, 1)
            GPIO.output(led_R, 0)
            print("Motion detected!")
            time.sleep(0.2)

        if GPIO.input(sensor) == 0:
            GPIO.output(led_Y, 0)
            GPIO.output(led_R, 1)
            print("Motion undetected!")
            time.sleep(0.2)

except KeyboardInterrupt:
    print("Stopped by user")
    GPIO.cleanup()
```

Raspberry Pi를 활용한 IoT 프로젝트

Thank You