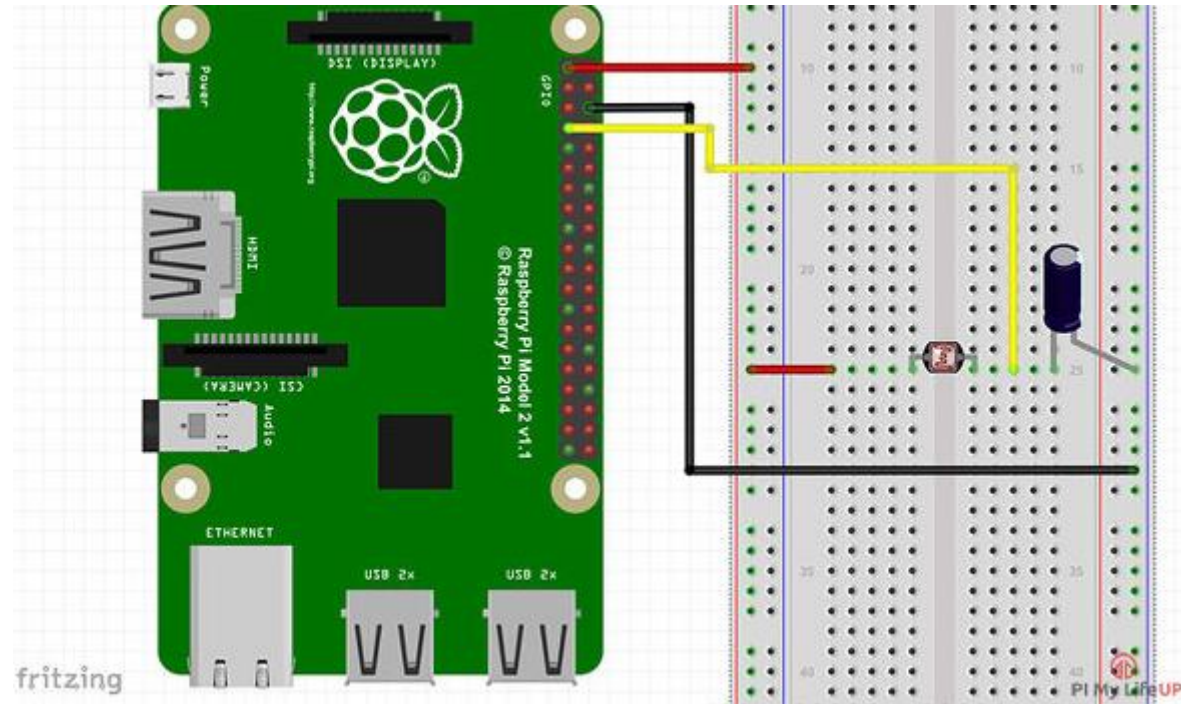
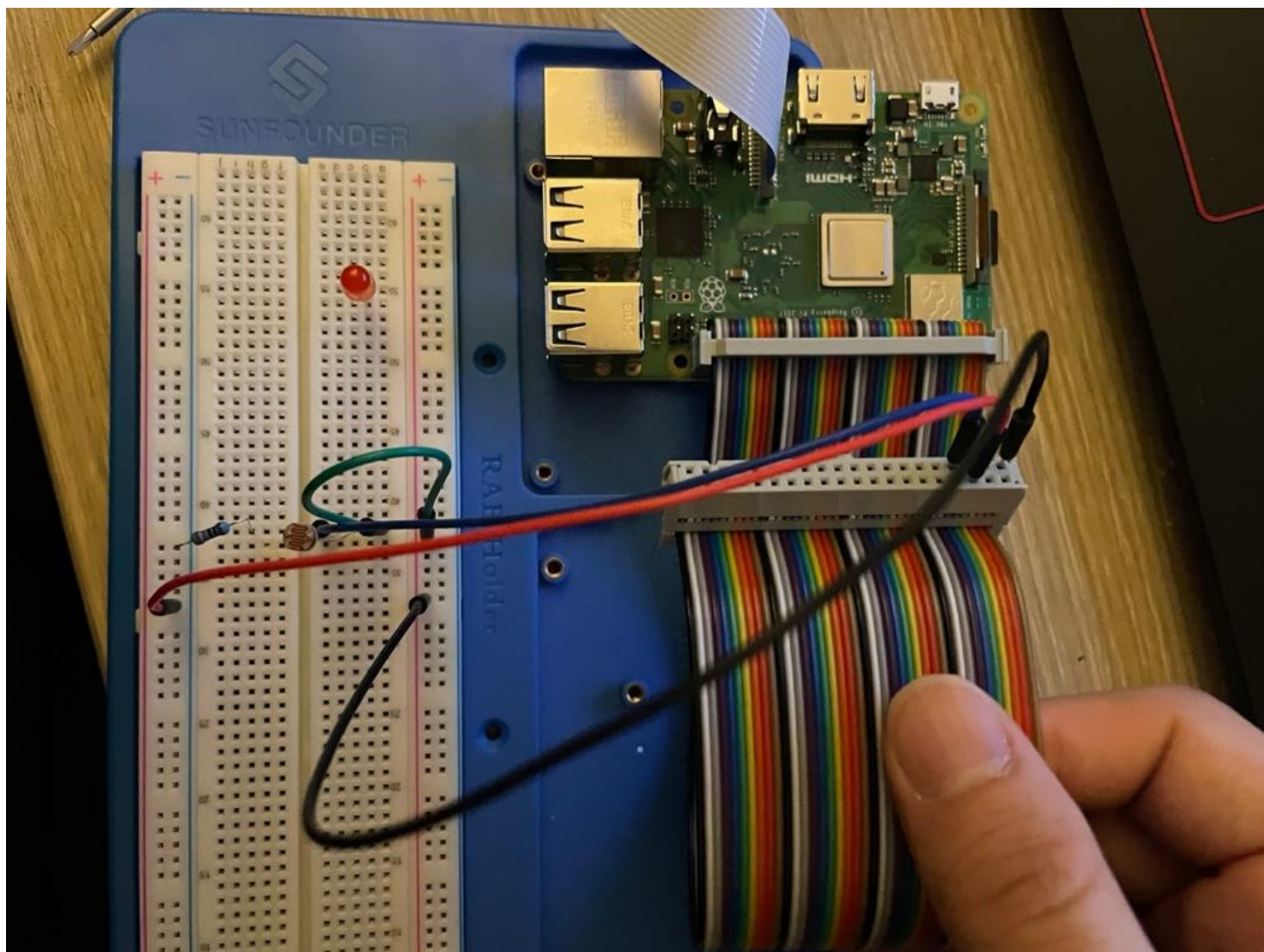


빛 감지, 가습기, 카메라

# 빛 감지





```
1  #!/usr/local/bin/python
2
3  import RPi.GPIO as GPIO
4  import time
5
6  __author__ = 'Gus (Adapted from Adafruit)'
7  __license__ = "GPL"
8  __maintainer__ = "pimylifeup.com"
9
10 GPIO.setmode(GPIO.BOARD)
11
12 #define the pin that goes to the circuit
13 pin_to_circuit = 7
14
15 def rc_time (pin_to_circuit):
16     count = 0
17
18     #Output on the pin for
19     GPIO.setup(pin_to_circuit, GPIO.OUT)
20     GPIO.output(pin_to_circuit, GPIO.LOW)
21     time.sleep(0.1)
22
23     #Change the pin back to input
24     GPIO.setup(pin_to_circuit, GPIO.IN)
25
```

```
25
26     #Count until the pin goes high
27     while (GPIO.input(pin_to_circuit) == GPIO.LOW):
28         count += 1
29
30     return count
31
32 #Catch when script is interrupted, cleanup correctly
33 try:
34     # Main loop
35     while True:
36         print(rc_time(pin_to_circuit))
37 except KeyboardInterrupt:
38     pass
39 finally:
40     GPIO.cleanup()
```

---

# 카메라



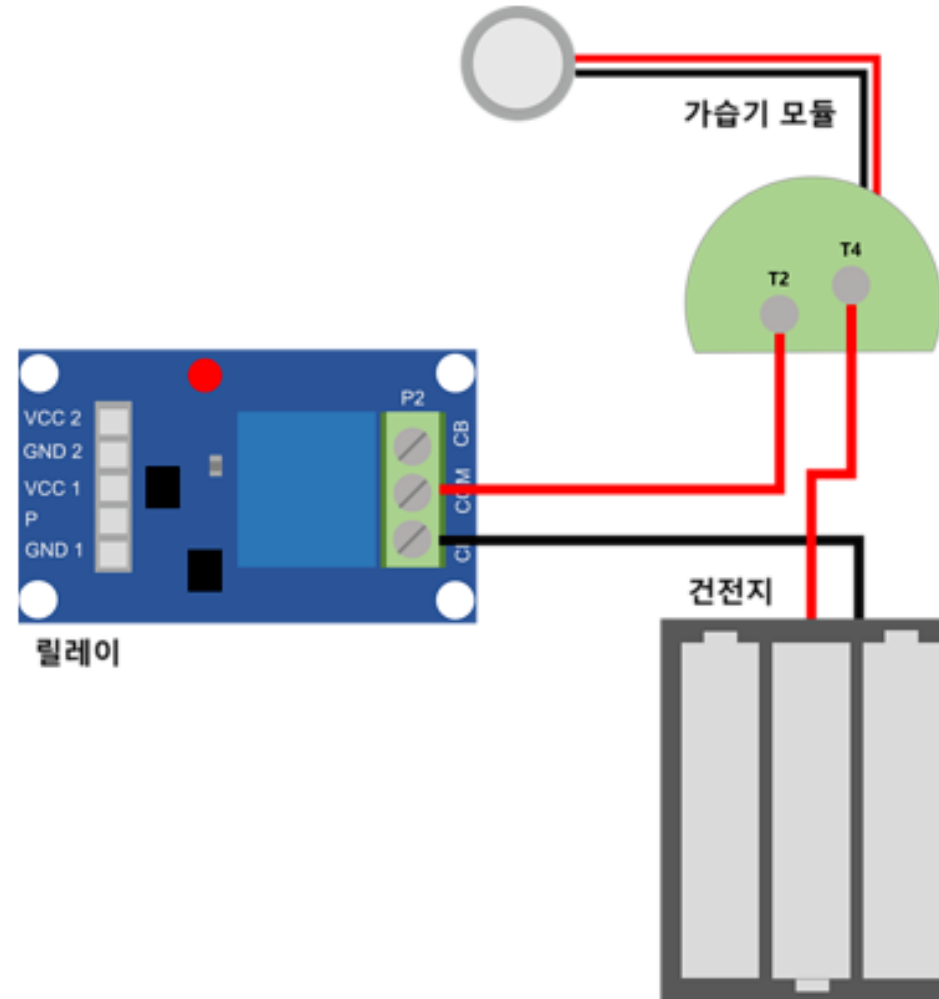
1. **PiCamera**의 **capture** 메서드를 이용해 사진을 찍을 수 있다. **capture** 매개변수에 저장할 파일명을 입력한다. 다음을 실행하면 5초 있다가 사진을 찍어 저장한다.  
파이 카메라 조도가 설정되는 시간을 기다려주기 위해 사진을 찍기 전 최소 2초 정도 여유를 두는 것이 좋다.

```
from picamera import PiCamera
from time import sleep

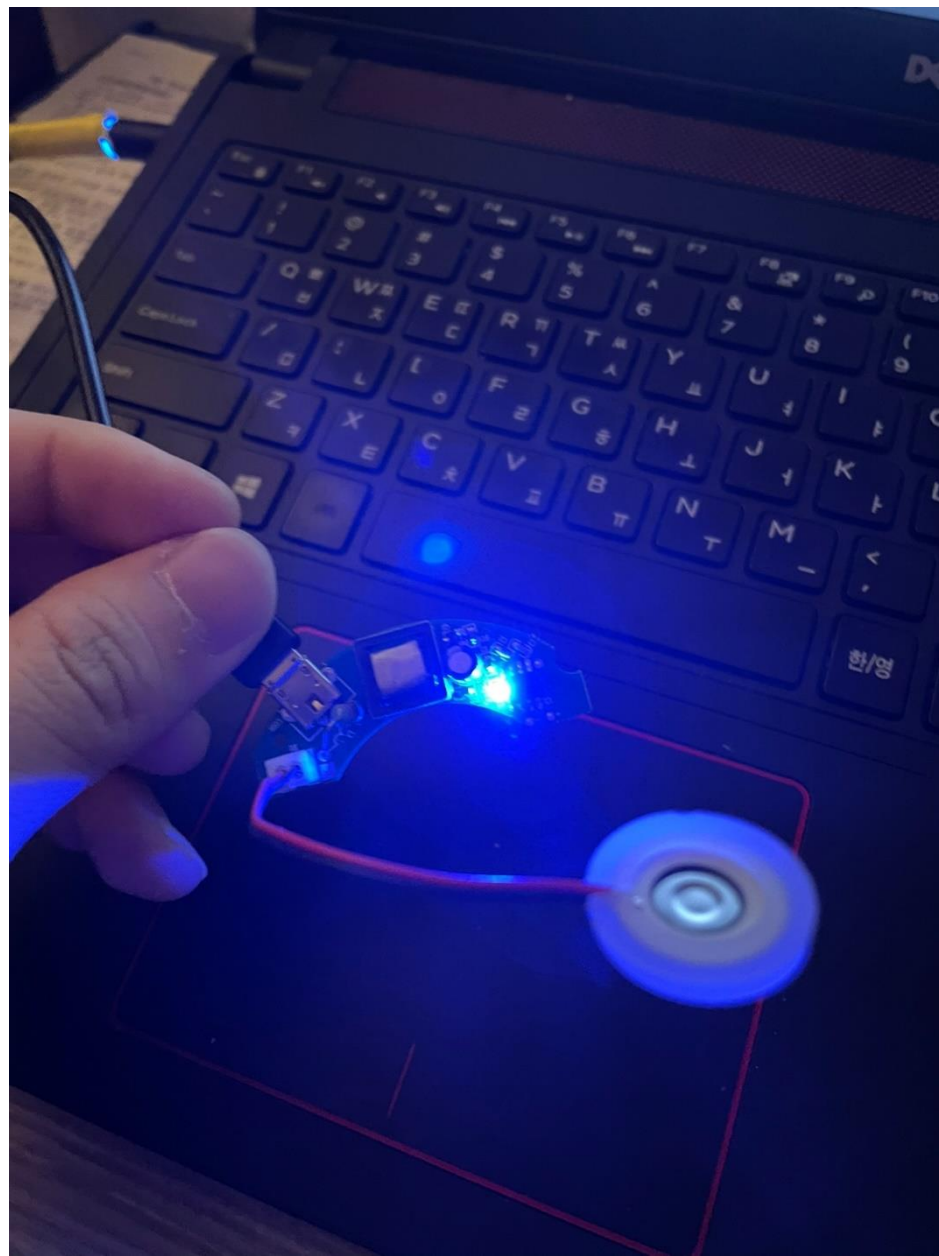
camera = PiCamera()

camera.start_preview()
sleep(5)
camera.capture('/home/pi/image.jpg')
camera.stop_preview()
```

# 가습기 모듈









```
#include <LiquidCrystal.h>
#include <DHT11.h>
#define relay 9      //relay와 연결된 9번 핀
#define dht11Pin 6   //dht11과 연결된 6번 핀

LiquidCrystal lcd(12,11,5,4,3,2); //RS 핀, E핀, 데이터 핀 4개
String lcdString = "";

DHT11 dht11(dht11Pin);  //dht11 라이브러리 사용을 위한 객체 선언

void setup()
{
  pinMode(relay, OUTPUT);

  lcd.begin(16,2);      //LCD 크기 지정, 2줄 16칸
  lcd.clear();          //화면 지우기

  Serial.begin(9600);
}
```

```
void loop()
{
  int err;  //에러구분 변수
  float humi; //습도 저장 변수
  float tem; //온도 저장 변수, 필요한 분은 사용하세요.

  if((err=dht11.read(humi, tem))==0) //습도값을 에러없이 읽어 왔다면
  {
    onOff(humi);      //가습기 on/off 함수 호출
    lcdHumidity(humi); //현재 습도 lcd 출력 함수 호출
  }
  else
  {
    //읽어오는 도중 문제가 발생할 경우 에러번호와 에러 시리얼 모니터에 출력
    Serial.println();
    Serial.print("Error No :");
    Serial.print(err);
    Serial.println();
  }
  delay(DHT11_RETRY_DELAY); //다시 읽어올때 까지 멈춤
}
```

//가습기 on/off 함수 호출

```
void onOff(float h) {
```

```
    if (h < 50 ) { //습도가 50%보다 작으면
```

```
        digitalWrite(relay, HIGH); //가습기 ON
```

```
    }else {
```

```
        digitalWrite(relay,LOW);
```

```
    }
```

```
}
```

//현재 습도 LCD 출력 함수

```
void lcdHumidity(float hu) {
```

```
    lcd.setCursor(0,0); //커서 0,0에 배치
```

```
    lcd.print("current humidity"); //""안에 글자 출력
```

```
    lcd.setCursor(1,3); //커서 위치 1,3으로 변경 후
```

```
    char buffer[10]; //float를 string으로 변환하기 위한 임시 char 배열 선언
```

```
    dtostrf(hu, 4, 1, buffer); //float -> char로 변환
```

```
    String humidity = " : "; //두번째 줄에 출력할 구문 만들기 (: 습도 %)
```

```
    humidity += (String)buffer; //char -> string
```

```
    humidity += "%";
```

```
    Serial.print(humidity);
```

```
    lcd.print(humidity); //현재 습도 출력
```

```
}
```