

라즈베리파이4를 이용한 간단한 I/O 제어 보고서

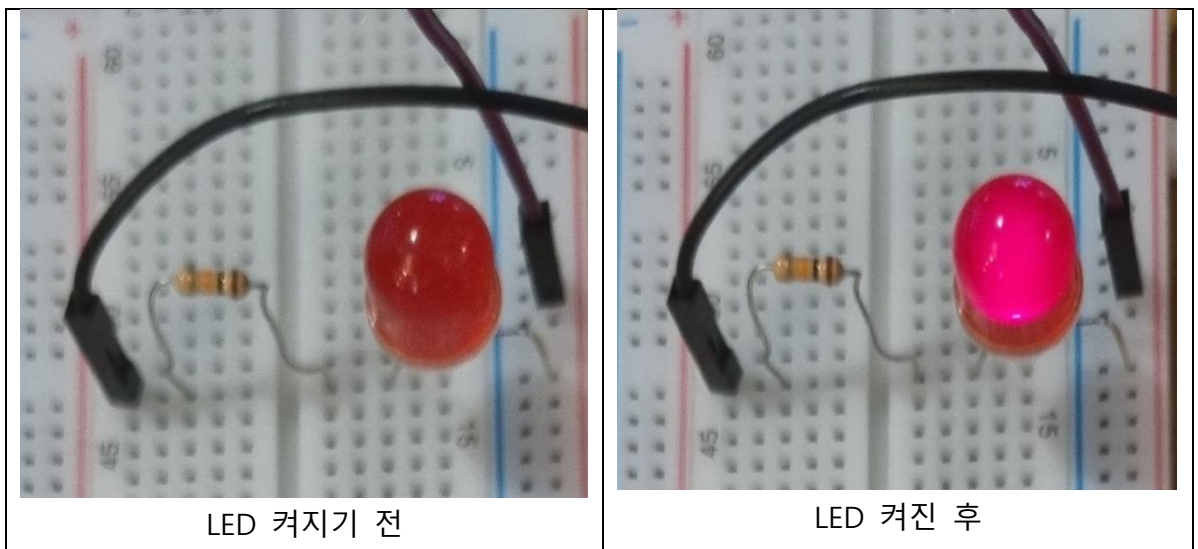
송용찬, 신명준, 염정현

임베디드 시스템 9조

작동 영상은 [ecampus](#)에 같이 첨부해서 업로드 했습니다.

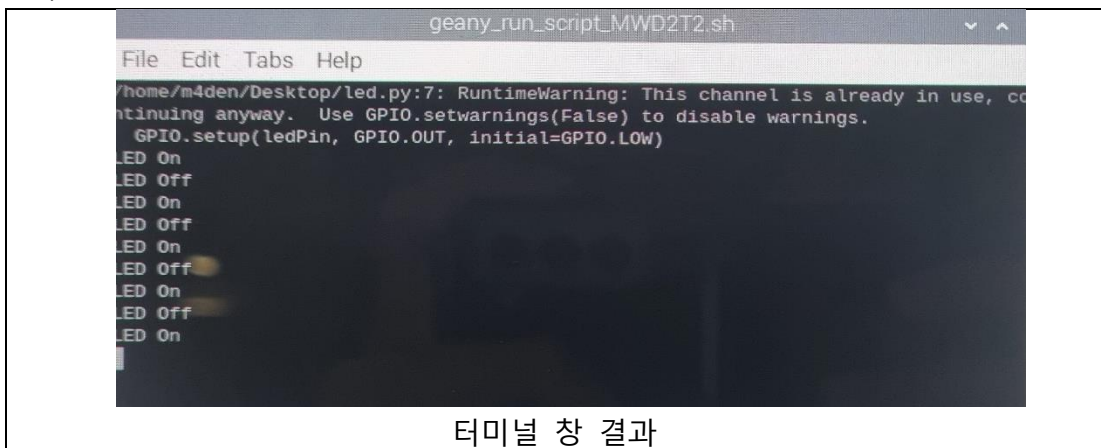
1. LED 를 부착하고 1초마다 LED 가 켜지고 꺼지게 하는 프로그램

1-1) 회로 사진



회로 설명: LED를 18번 핀과 저항, GND와 연결했습니다.

1-2) 터미널 창 결과

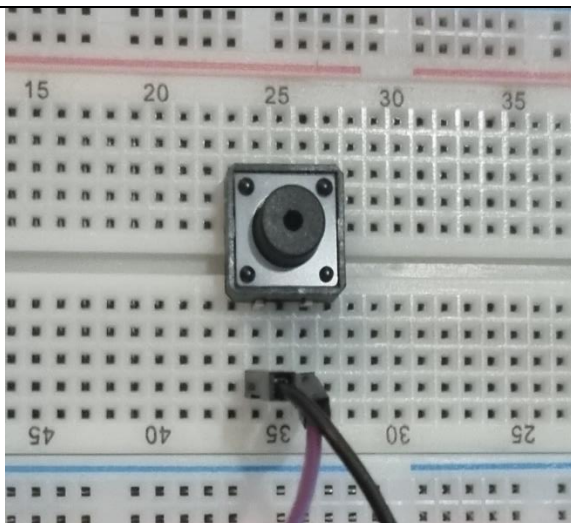


1-3) 코드

```
import RPi.GPIO as GPIO
import time
ledPin = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(ledPin, GPIO.OUT, initial=GPIO.LOW)
while True:
    GPIO.output(ledPin, GPIO.HIGH)
    print("LED On")
    time.sleep(1)
    GPIO.output(ledPin, GPIO.LOW)
    print("LED Off")
    time.sleep(1)
```

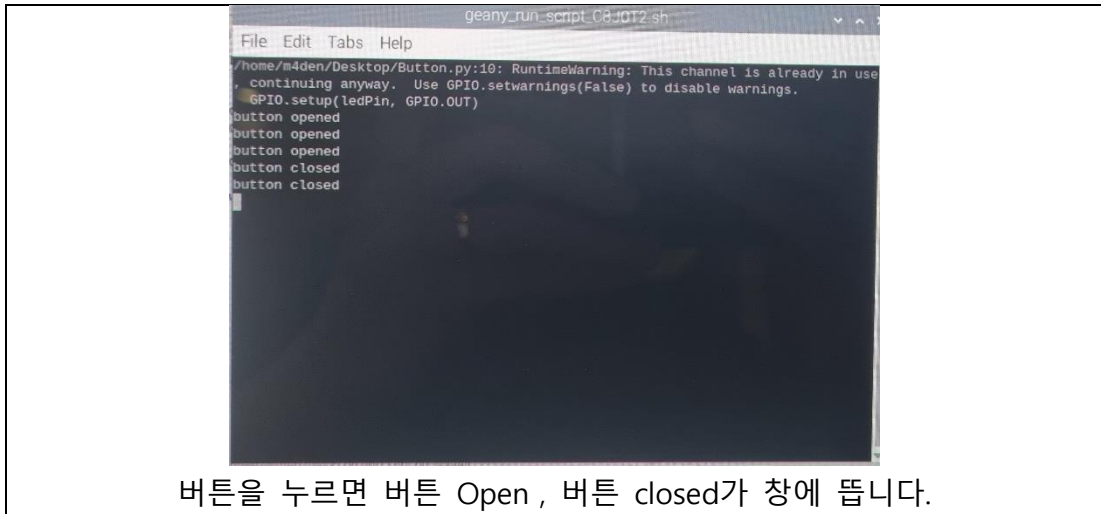
-
2. 버튼 스위치를 부착하고 버튼을 누르면 LED 가 켜지고 버튼을 떼면 LED 가 꺼지도록 한다. (혹은 터미널 창에 "버튼 눌림", "버튼 풀림" 을 프린트 하도록 해도 무방)

2-1) 회로 사진



버튼을 GND와 15번 핀에 각각 연결했습니다.

2-2) 터미널 창 결과

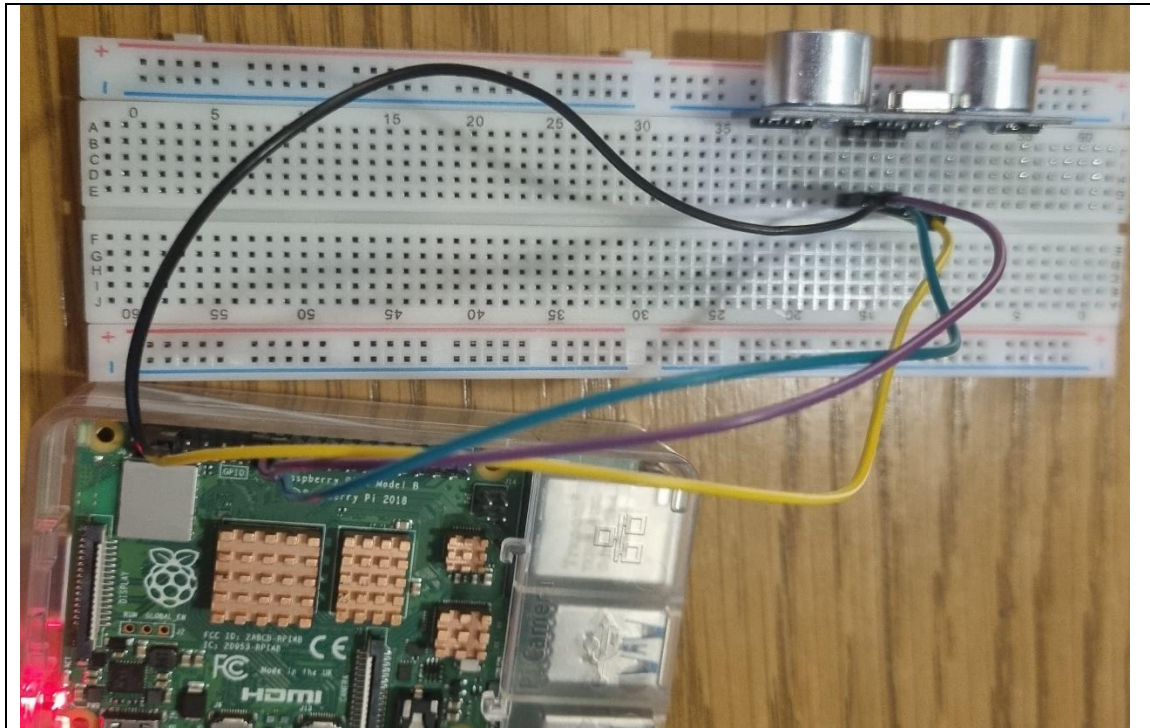


2-3) 코드

```
import RPi.GPIO as GPIO
import time
button_pin = 15
GPIO.setmode(GPIO.BCM)
GPIO.setup(button_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
while True:
    buttonState = GPIO.input(button_pin)
    if buttonState == False:
        print("button closed")
    else:
        print("button opened")
    time.sleep(1)
```

3. 거리 센서를 부착하고 센서에 장애물을 가져다 대면 매 1초마다 거리를 측정해서 프린트 하는 프로그램 작성, "10cm 에 장애물이 있습니다. "

3-1) 회로 사진



3-2) 터미널 창 결과

영상으로 첨부

3-3) 코드

```
코드)
import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

TRIGGER = 24
ECHO = 23

GPIO.setup(TRIGGER, GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

start = time.time()
```

```
try:
    while True:
        GPIO.output(TRIGGER,GPIO.LOW)
        time.sleep(1)
        GPIO.output(TRIGGER,GPIO.HIGH)
        time.sleep(0.00002)
        GPIO.output(TRIGGER,GPIO.LOW)

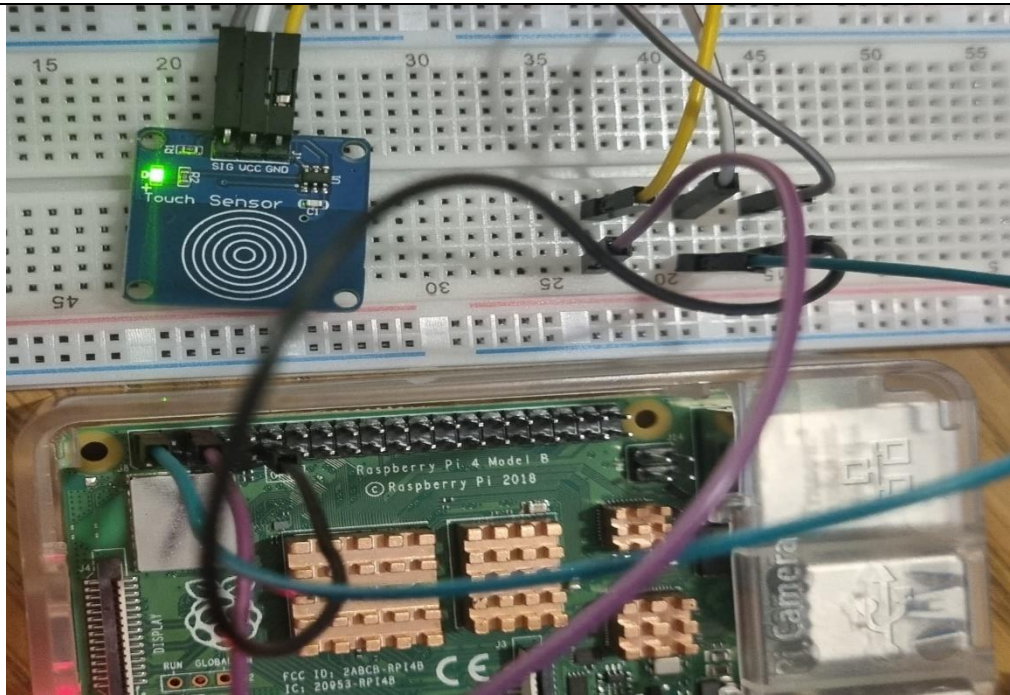
        while GPIO.input(ECHO)==GPIO.LOW:
            start = time.time()
        while GPIO.input(ECHO)==GPIO.HIGH:
            stop = time.time()

        period=stop-start
        dist1=round(period*1000000/58,2)
        dist2=round(period*17241,2)
        distance=(dist1+dist2)/2
        print("Distance : %.1f cm" %distance)

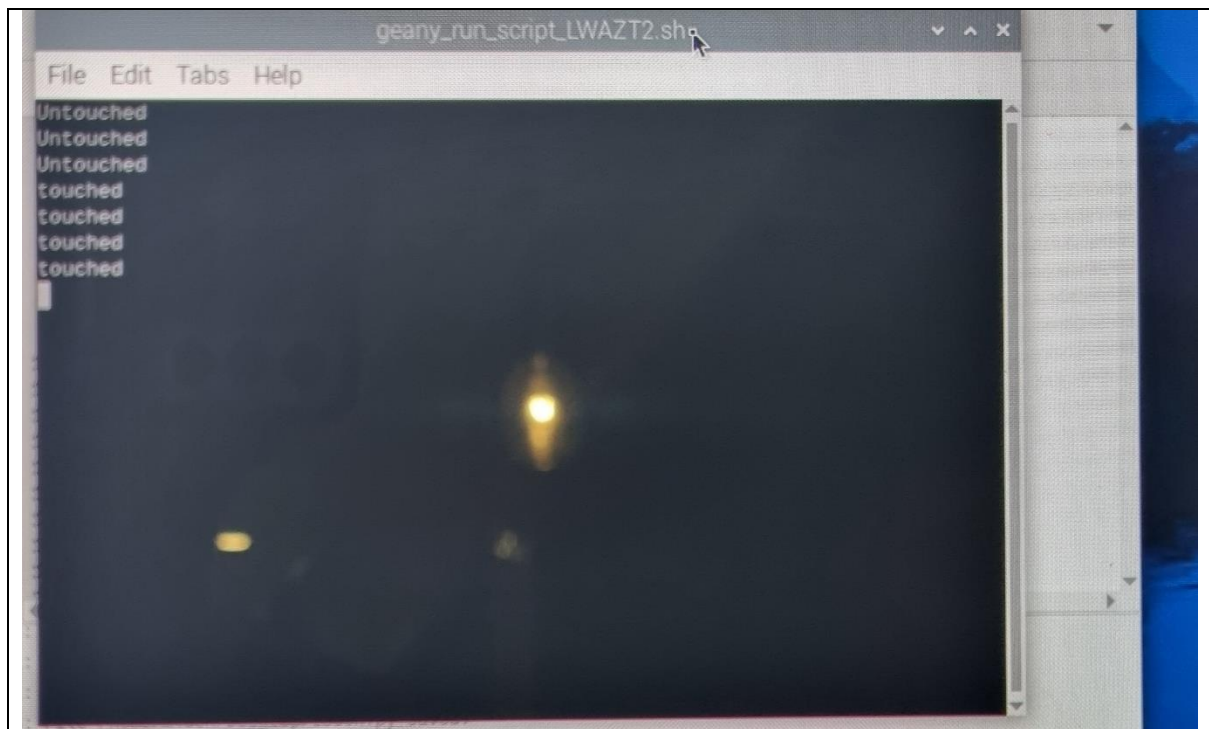
except KeyboardInterrupt:
    print("Complete")
    GPIO.cleanup()
```

4. 터치 센서를 부착하고 센서를 터치하면 "터치 되었습니다", "터치 되지 않았습니다" 와 같이 출력될 수 있도록 프로그램 작성

4-1) 회로 사진



4-2) 터미널 창 결과



4-3) 코드

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

```
touchPin = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(touchPin,GPIO.IN)

try:
    while True:
        touchState = GPIO.input(touchPin)
        if touchState == GPIO.HIGH:
            print("touched")
        else:
            print("Untouched")

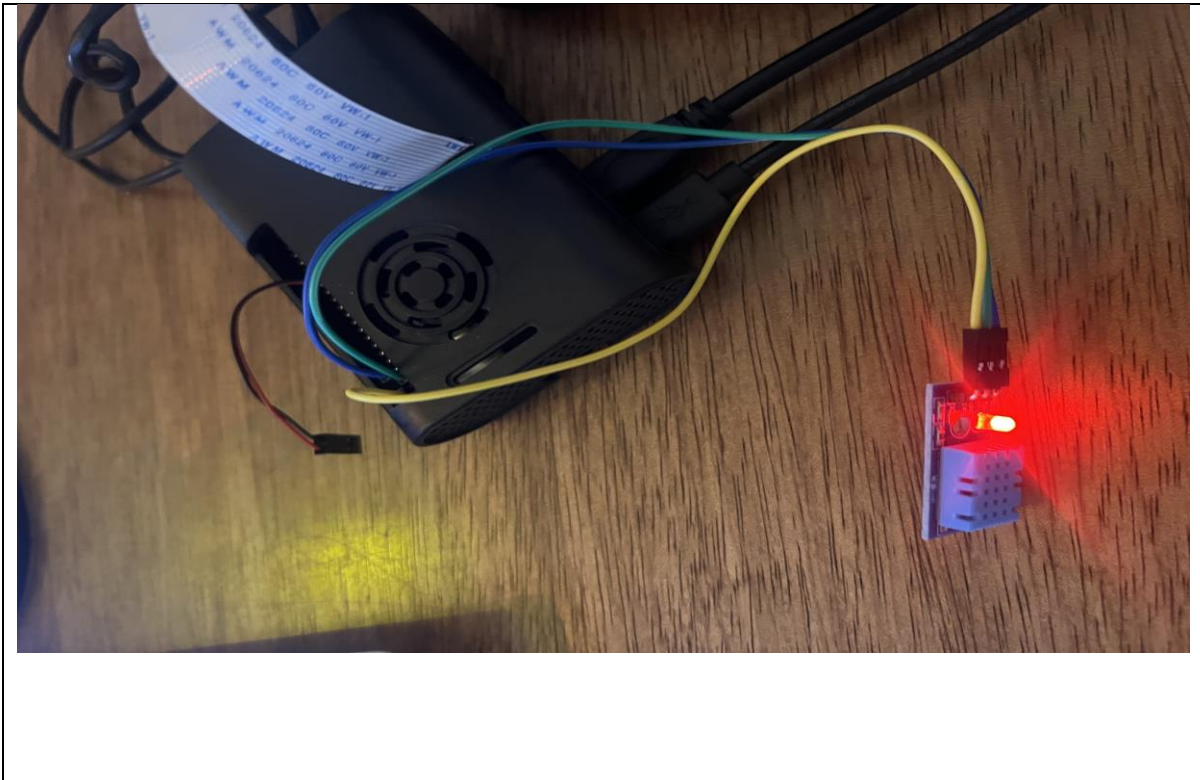
        time.sleep(1)

except KeyboardInterrupt:
    print("Quit")

finally:
    GPIO.cleanup()
```

5. 온습도 센서를 부착하고 매 1초마다 "온도 : 23.5도, 습도 : 45 %" 와 같이 출력되도록 프로그램 작성

5-1) 회로 사진



5-2) 터미널창 결과

영상으로 첨부

5-3) 코드

```
import adafruit_dht
import board
import time

dht_device = adafruit_dht.DHT11(board.D2)

try:
    while True:
        try:

            humidity = dht_device.humidity
            temperature = dht_device.temperature
```



```
        if humidity is not None and temperature is not None:
            print("Temperature: {:.1f} C".format(temperature))
            print("Humidity: {:.1f} %".format(humidity))
        else:
            print("Failed to retrieve data from sensor")

    except RuntimeError as error:

        print("Error reading sensor data: ", error)

    except Exception as e:

        dht_device.exit()
        print("Error: ", e)
        break

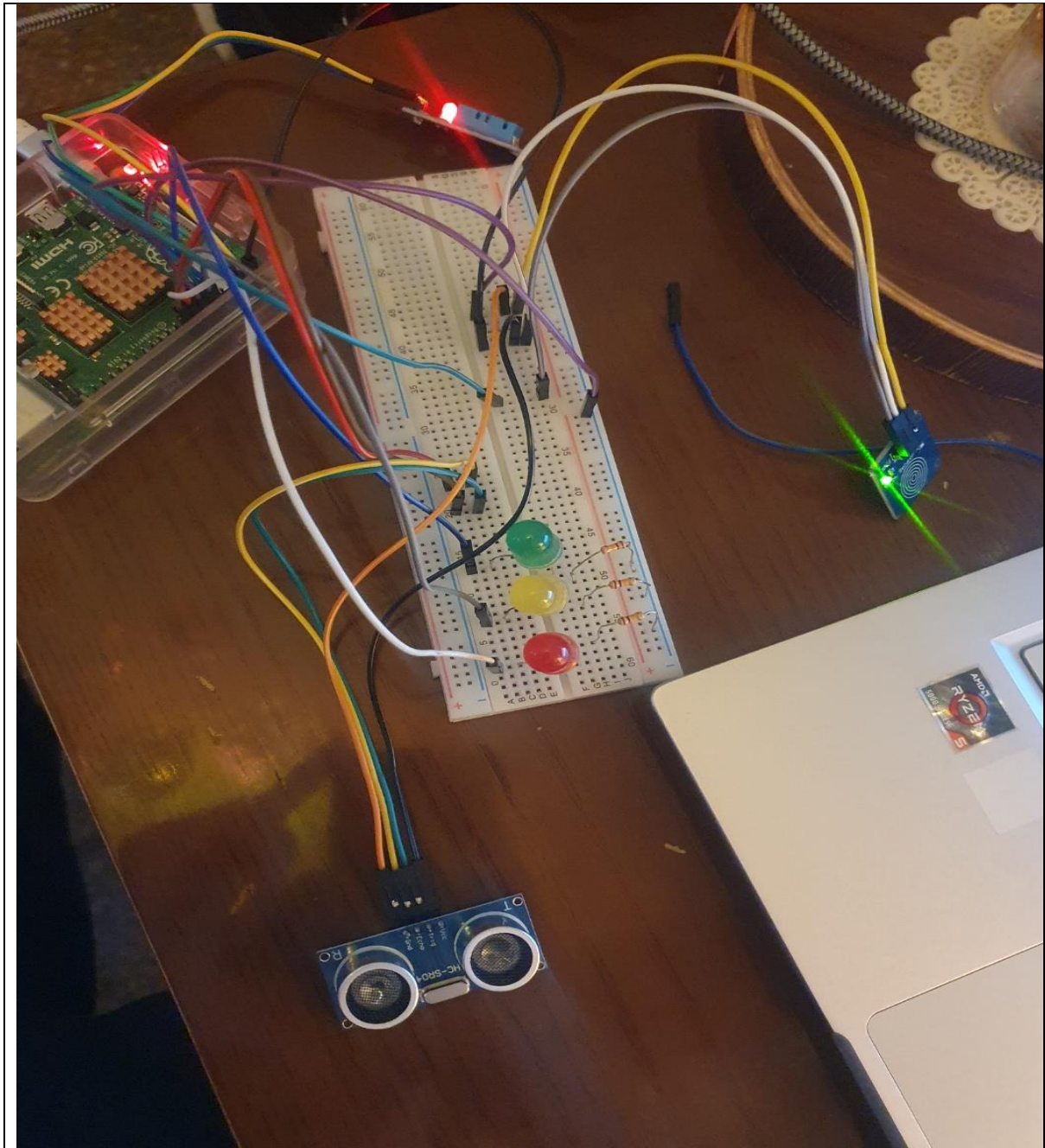
    time.sleep(1)

except KeyboardInterrupt:
    print("Program terminated by user.")

finally:
    dht_device.exit()
```

6. 종합 라즈베리파이 시스템 설계

6-1) 회로 사진



6-2) 터미널 창 결과

영상으로 첨부

6-3) 코드

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

```
import adafruit_dht
import board

dht_device = adafruit_dht.DHT11(board.D2)
yled = 18
gled = 27
rled = 22
touchPin = 17
TRIGGER = 24
ECHO = 23

currentState = False
previousTouchState = GPIO.LOW

temperature_printed = False

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(yled, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(gled, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(rled, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(TRIGGER, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.setup(touchPin, GPIO.IN)

def get_distance():
    GPIO.output(TRIGGER, GPIO.LOW)
    time.sleep(0.01)
    GPIO.output(TRIGGER, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(TRIGGER, GPIO.LOW)

    start = time.time()
    stop = time.time()
```

```

while GPIO.input(ECHO) == GPIO.LOW:
    start = time.time()
while GPIO.input(ECHO) == GPIO.HIGH:
    stop = time.time()

elapsed_time = stop - start
distance = (elapsed_time * 34300) / 2
return round(distance, 2)

def read_sensor_data():
    try:
        humidity = dht_device.humidity
        temperature = dht_device.temperature
        return humidity, temperature
    except RuntimeError as e:
        print(f"RuntimeError! ")
        time.sleep(2)
        return read_sensor_data()
    except Exception as e:
        print(f"Error ! ")
        return None, None

try:
    while True:
        touchState = GPIO.input(touchPin)

        if touchState == GPIO.HIGH and previousTouchState == GPIO.LOW:
            if not currentState:
                currentState = True
                print("System operating")
                if not temperature_printed:
                    humidity, temperature = read_sensor_data()
                    if humidity is not None and temperature is not None:
                        print("Temperature: {:.1f} C".format(temperature))
                        print("Humidity: {:.1f} %".format(humidity))

```

```

        temperature_printed = True
    else:
        print(" sensor error ")
    else:
        currentState = False
        print("System finished.")
        GPIO.output(yled, GPIO.LOW)
        GPIO.output(gled, GPIO.LOW)
        GPIO.output(rled, GPIO.LOW)
        break

if currentState:
    distance = get_distance()
    print(f"Distance: {distance} cm")

    if distance > 20:
        GPIO.output(gled, GPIO.HIGH)
        GPIO.output(yled, GPIO.LOW)
        GPIO.output(rled, GPIO.LOW)
        print("LED Green ON - Distance > 20 cm")
    elif 10 < distance <= 20:
        GPIO.output(gled, GPIO.LOW)
        GPIO.output(yled, GPIO.HIGH)
        GPIO.output(rled, GPIO.LOW)
        print("LED Yellow ON - 10 < Distance <= 20 cm")
    elif distance <= 10:
        GPIO.output(gled, GPIO.LOW)
        GPIO.output(yled, GPIO.LOW)
        GPIO.output(rled, GPIO.HIGH)
        print("LED Red ON - Distance <= 10 cm, TOO CLOSE")

previousTouchState = touchState
time.sleep(1)

```

except KeyboardInterrupt:

```
print("Program error ")
```

```
finally:
```

```
    GPIO.cleanup()
```

```
    print("program exit .")
```