

REPORT



과 목 명 : 정보통신종합설계

<Assignment #2 Programming using RaspberryPi>

학 과 : 정보통신공학과

201200614 김 영 상

201202682 이 재 환



HANKUK UNIVERSITY OF FOREIGN STUDIES

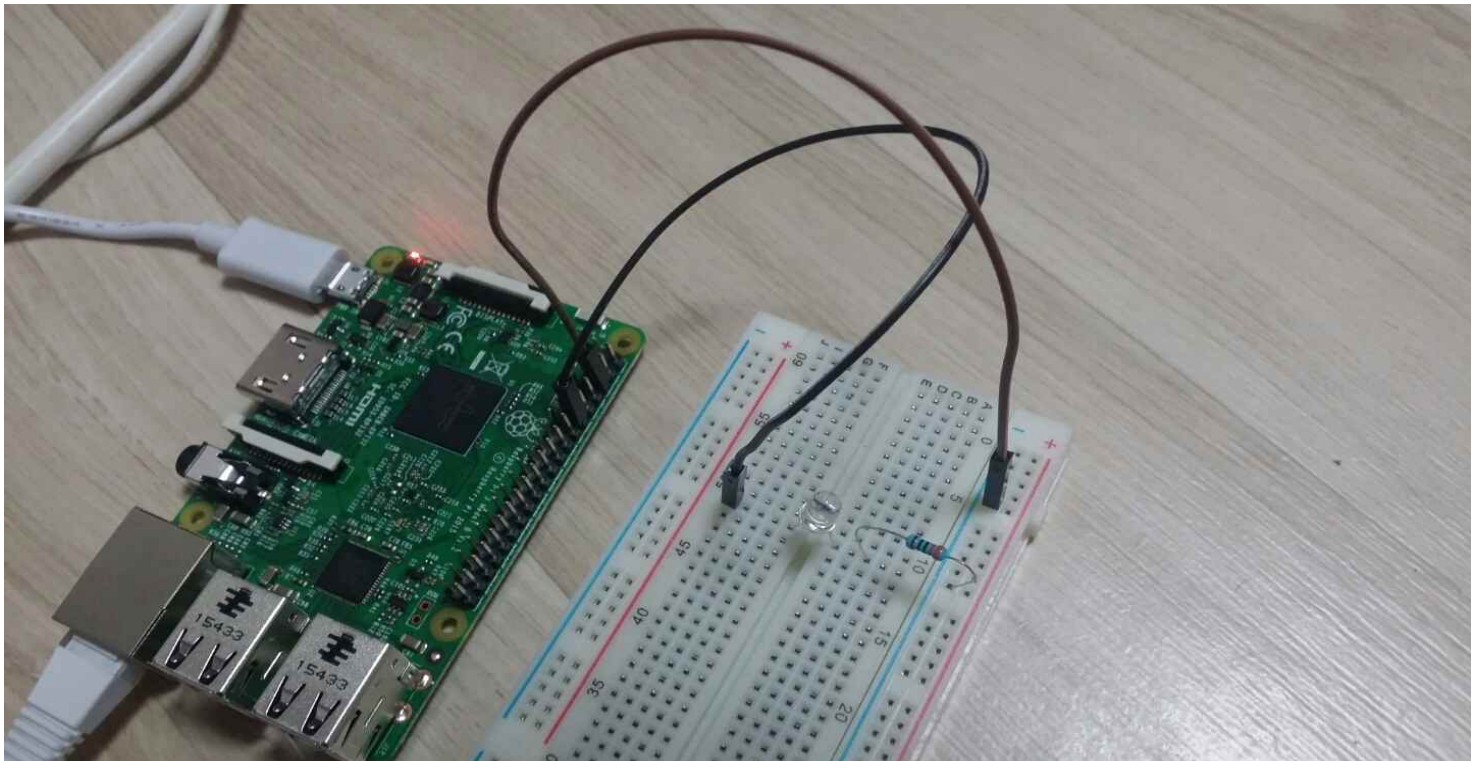
<Example Program>

1. LED Control

1-1. Code

```
pi@raspberrypi: ~  
import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BCM) //GPIO를 사용하기 위해 설정 한다.  
  
LED = 18  
  
GPIO.setup(LED, GPIO.OUT) //led의 초기값을 설정 한다.  
  
try:  
    while(True):  
        GPIO.output(LED, GPIO.HIGH) //led를 점등 한다.  
        time.sleep(1) // delay를 준다.  
        GPIO.output(LED, GPIO.LOW) //led를 소등 한다.  
        time.sleep(1) // delay를 준다.  
except KeyboardInterrupt:  
    GPIO.output(LED, GPIO.LOW) //GPIO종료시 led를 소등 한다.  
    GPIO.cleanup() //프로그램 종료시 GPIO핀 설정을 초기화 해준다.
```

1-2. 회로 사진



1초 간격으로 led가 켜졌다 꺼졌다를 반복하는 설계입니다.

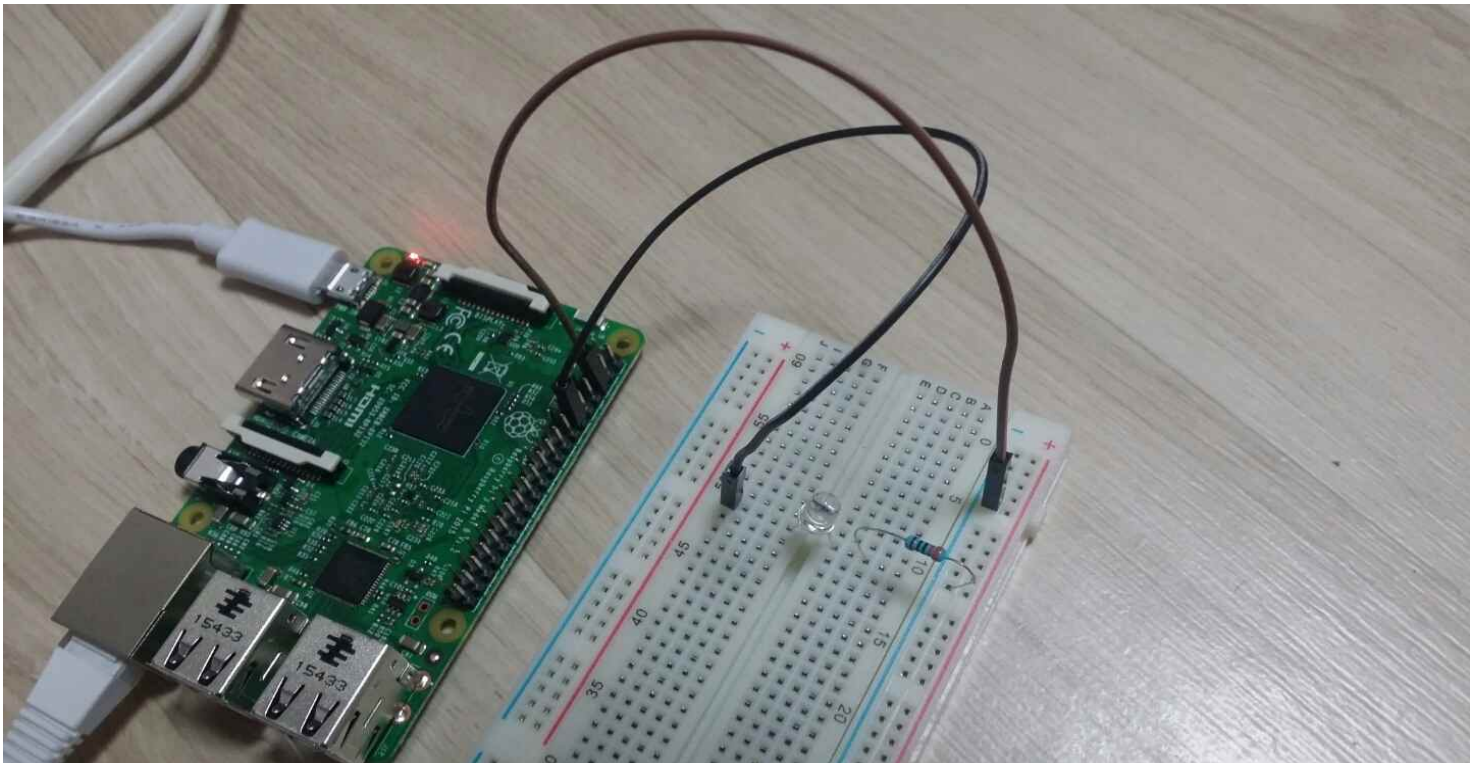
url: <https://youtu.be/xam5N6u02pg>

2. LED 밝기조절 (using PWM)

2-1. Code

```
pi@raspberrypi: ~  
import RPi.GPIO as GPIO  
import time  
  
LED = 18  
GPIO.setmode(GPIO.BCM) //GPIO를 사용하기위해 설정 한다.  
GPIO.setup(LED, GPIO.OUT) //led의 초기값을 설정 한다  
pwm_led = GPIO.PWM(LED,500) //pwm 을 이용하여 led의 밝기를 조절한다.  
pwm_led.start(1) // led의 초기값을 1로 설정 한다.  
delay = 0.01 // delay를 0.01로 준다.  
try:  
    while(1):  
        for i in range(0,101): // 0~100까지 밝기를 올려준다.  
            pwm_led.ChangeDutyCycle(i)  
            time.sleep(delay)  
  
        for i in range(100,-1,-1): // 100~0까지 밝기를 낮춰준다.  
            pwm_led.ChangeDutyCycle(i)  
            time.sleep(delay)  
  
except KeyboardInterrupt:  
    pwm_led.stop() //종료시 led를 소등 한다.  
    GPIO.cleanup() // 프로그램 종료시 GPIO핀 설정을 초기화 한다.
```

2-2. 회로 사진



led의 밝기가 서서히 어두워졌다가 다시 서서히 밝아지기를 반복하는 설계입니다.

주어진 예제 코드와는 약간 다르게 pwm을 사용하여 밝기를 조절하였고

주어진 예제 코드에서는 delay를 0.1초로 하였으나 저희 조에서는 좀 더 빠른 변화값을 확인하기 위하여 delay를 0.01로 설정하였습니다.

url: https://youtu.be/js_vLXeyFOU

3. Arduino와 Raspberry Pi 시리얼 통신하기 (시리얼 통신으로 0~100까지 출력)

3-1-1. 라즈베리파이 코드

```
pi@raspberrypi: ~
import serial

port = '/dev/ttyACM0'//아두이노 포트설정을 해준다.
serialFromArduino = serial.Serial(port,9600) //파이 시리얼을 이용해 아두이노와
//연결을 해준다.

serialFromArduino.flushInput()

while True:
    input_s = serialFromArduino.readline() //아두이노에서 라즈베리파이로
    input = int(input_s) // 읽어들이어 출력한다.
    print input

"ex3.py" 13 lines, 417 characters
```

3-1-2. 아두이노 코드

```
void setup(){
    Serial.begin(9600); // 9600 비트레이트로 시작합니다.
}

void loop(){
    for (int n=0; n < 101; n++){ // 0~100까지의 숫자를 전송합니다.
        Serial.println(n,DEC);
        delay(500);
    }
}
```

3-2. 회로 사진



라즈베리파이와 아두이노 사이의 시리얼 통신을 통해 아두이노에서 0~100까지의 숫자를 시리얼 통신으로 전송하고 라즈베리파이에서 수신하여 모니터에 출력하는 설계입니다.

주어진 아두이노 예제 코드에서는 반복문을 0~1024까지 실행하지만 문제가 0~100까지 출력하기 임으로 반복문에서의 범위를 수정하여 실행하였습니다.

url: <https://youtu.be/OqckWiJP5Hk>

<Assignment>

4. Arduino와 Raspberry Pi의 시리얼 통신을 통한 온도센서로 Servo motor제어

4-1-1. 라즈베리파이 코드

```
pi@raspberrypi: ~  
import serial  
import RPi.GPIO as GPIO  
import time  
LED =18  
GPIO.setmode(GPIO.BCM) // GPIO를 사용하기 위해 설정 한다.  
GPIO.setup(LED, GPIO.OUT) // led의 초기값을 설정 한다.  
port_1 = '/dev/ttyACM0' //서보모터를 연결한 아두이노를 라즈베리파이에 연결 한다.  
port_2 = '/dev/ttyACM1' //온도센서를 연결한 아두이노를 라즈베리파이에 연결 한다.  
  
serialFromArduino1 = serial.Serial(port_1,9600) //pyserial을 이용하여 아두이노를  
serialFromArduino2 = serial.Serial(port_2,9600) //라즈베리파이에 연결 한다.  
  
serialFromArduino1.flushOutput()  
serialFromArduino2.flushInput()  
  
try:  
  
    while(True):  
        input_s = serialFromArduino2.readline() // 아두이노에서 라즈베리파이로  
        input_in = (input_s[:5]) // 온도를 읽어들인다.  
        input_out = float(input_in)  
        if input_out > 27:  
            GPIO.output(LED, GPIO.HIGH) //온도가 27도 이상이 되면  
            //Led를 점등하고 서보모터가 연결된  
            //아두이노로 '0'을 전송 한다.  
            output_s = serialFromArduino1.write('0'.encode())  
        else:  
            GPIO.output(LED, GPIO.LOW) //온도가 27도 이하면  
            //Led를 소등하고 서보모터가 연결된  
            //아두이노로 '1'을 전송 한다.  
            output_s = serialFromArduino1.write('1'.encode())  
        print 'temp is = ',input_out //온도를 출력한다.  
  
except KeyboardInterrupt :  
    GPIO.output(LED, GPIO.LOW) //GPIO종료시 led를 소등 한다.  
    GPIO.cleanup() //프로그램 종료시 GPIO핀 설정을 초기화 해준다.
```


4-1-2. 아두이노 코드 (온도센서)

```

    sketch_jun05a
const unsigned int TEMP_SENSOR_PIN = 0;
#include <Servo.h>
Servo myservo; // 서보모터를 선언해줍니다.

void setup(){
    Serial.begin(9600); // 9600보드레이트로 시작합니다.
}

void loop(){
    int voltage = analogRead(TEMP_SENSOR_PIN); // 온도센서에서 값을 받아옵니다.
    float temperature = (voltage * 5.0 / 1024) * 100 - 50; // 받은 값을 섭씨 온도로 변환해줍니다.

    Serial.println(temperature); // 변환된 온도를 시리얼 전송합니다.
    delay(1000); // 1초간 대기합니다.
}

```

4-1-3. 아두이노 코드 (Servo motor)

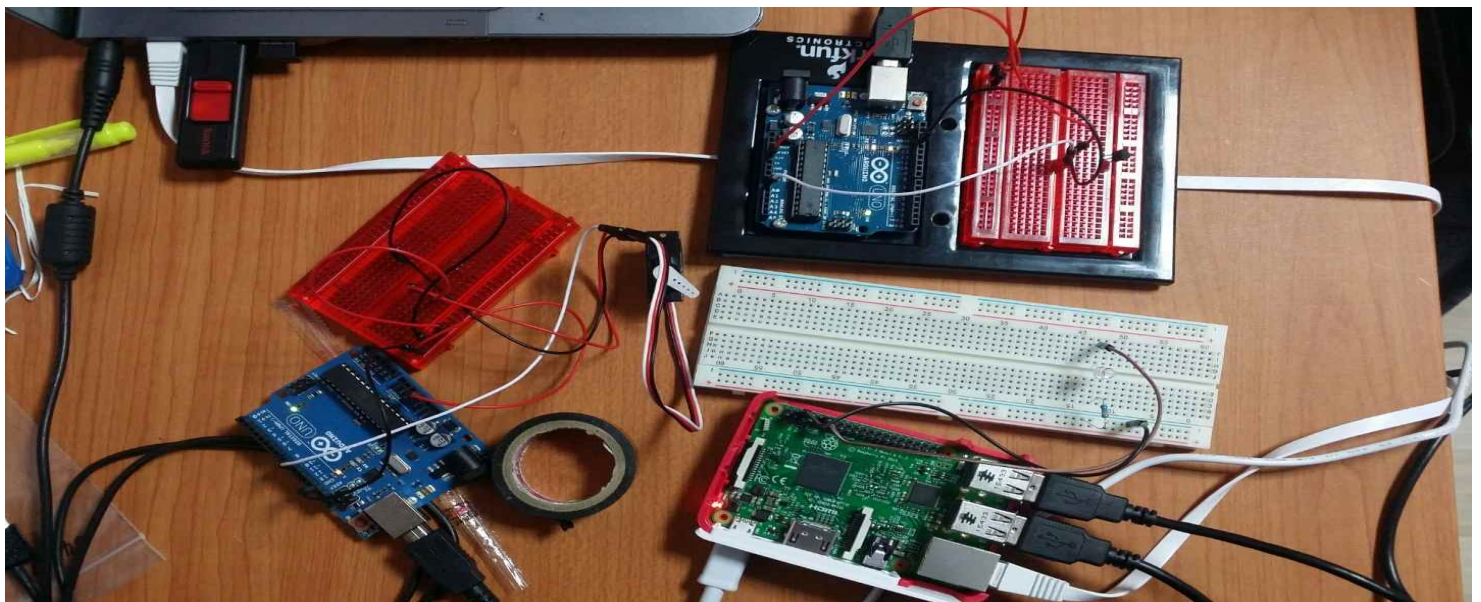
```
sketch_jun05a
#include <Servo.h>
Servo myservo; // 서보모터를 선언해줍니다.

void setup(){
  Serial.begin(9600); // 9600보드레이트로 시작합니다.
  myservo.attach(9); // 서보모터를 9번핀에 연결합니다.
  myservo.write(30); // 서보모터의 각도를 초기화 합니다.
}

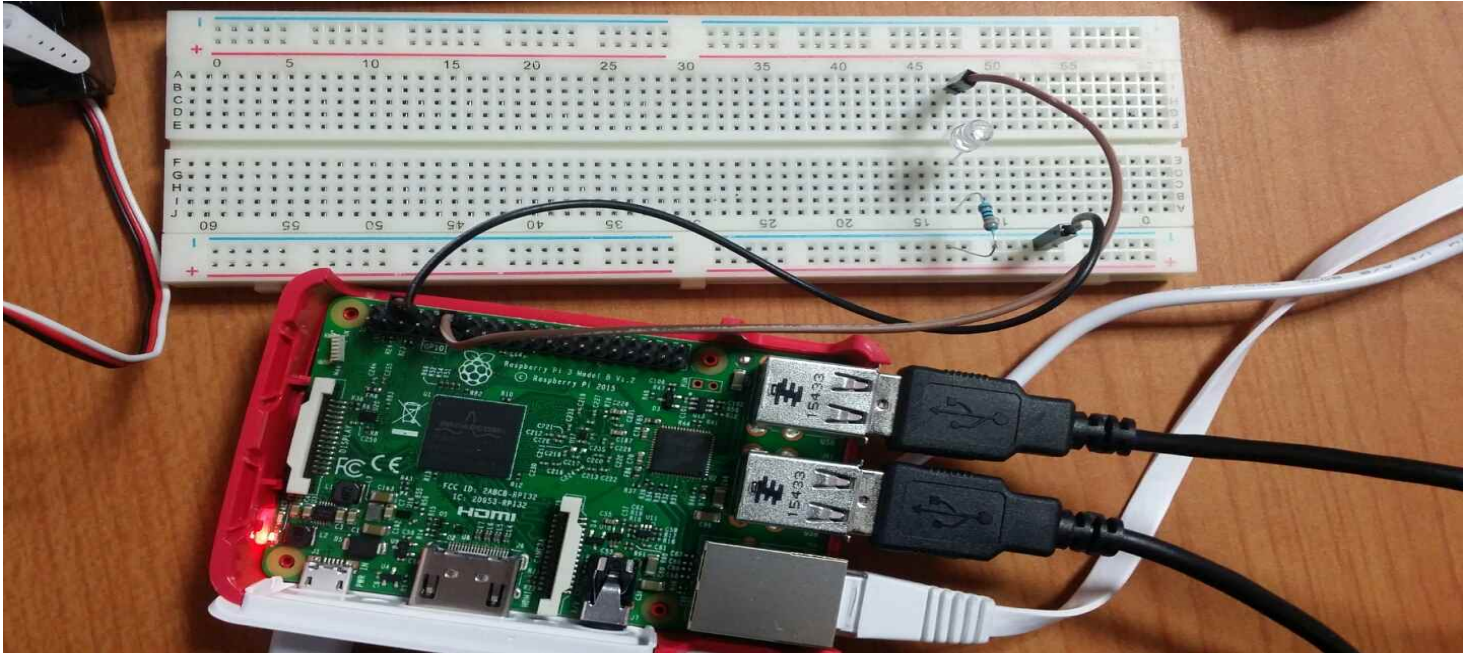
void loop(){
  while (Serial.available()){ // 시리얼 입력이 가능할 때 실행합니다.
    char tmp = (char)Serial.read(); // 시리얼 통신으로 입력값을 받아옵니다.
    if (tmp == '0'){ // 입력값이 0이면(온도가 27도보다 높을때) 실행합니다.
      myservo.write(60); // 서보모터의 각도를 60도로 변경합니다.
      delay(1000); // 1초간 대기합니다.
    }

    if (tmp == '1'){ // 입력값이 1이면(온도가 27도보다 낮을때) 실행합니다.
      myservo.write(30); // 서보모터의 각도를 30도로 변경합니다.
      delay(1000); // 1초간 대기합니다.
    }
  }
}
```

4-2-1. 회로 사진 (전체 회로)



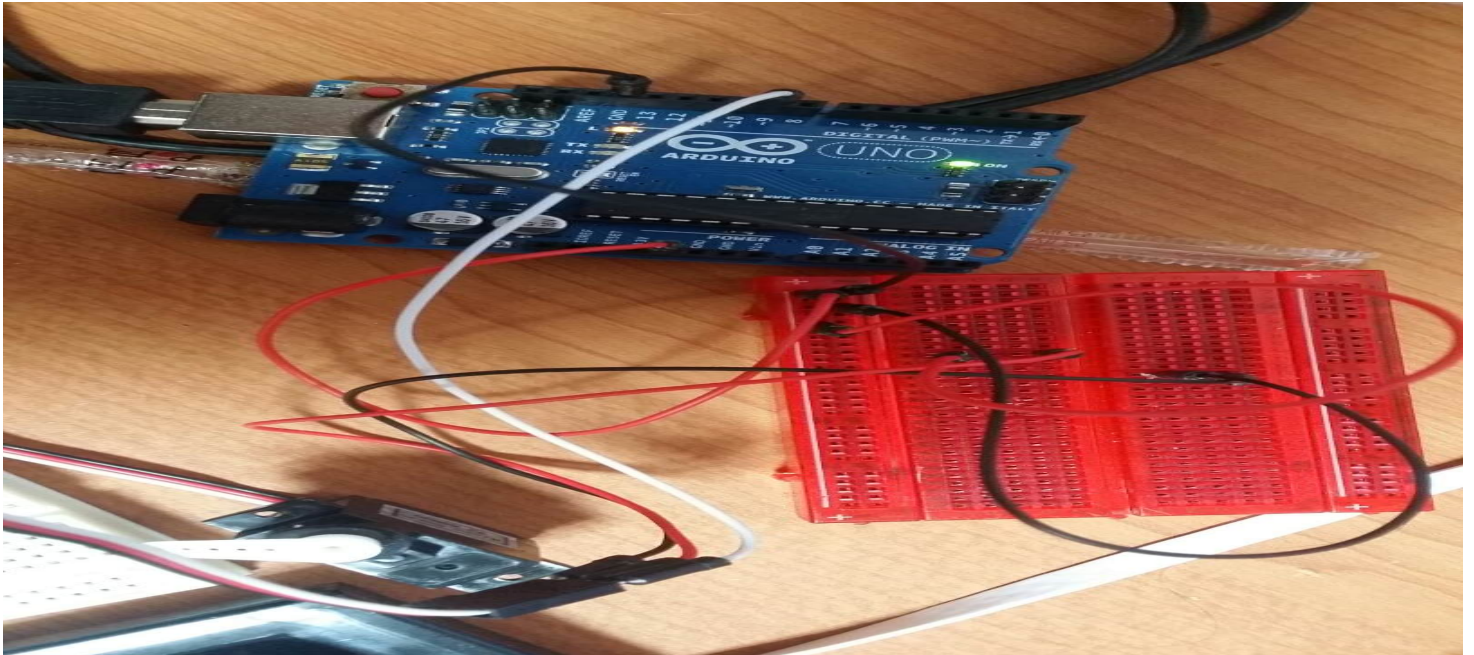
4-2-2. 회로 사진 (라즈베리 파이)



4-2-3. 회로 사진 (아두이노 - 온도센서)



4-2-4. 회로 사진 (아두이노 - servo motor)



1번 아두이노는 부착된 온도센서에서 측정한 값을 온도 공식에 대입하여 환산합니다.

환산된 값은 시리얼 통신을 통해 라즈베리 파이로 전송되며,

라즈베리 파이는 1번 아두이노로부터 받은 값이 27도가 보다 높으면

온도센서를 사용자가 잡은 것으로 인식하여 라즈베리 파이에 연결 되어 있는 LED를 켜고

시리얼 통신을 통해 2번 아두이노에 0이라는 시그널을 전송합니다.

2번 아두이노는 라즈베리 파이로부터 0이라는 시그널을 받으면 서보모터의 각도를 60도로 변경합니다.

반대로 온도센서로부터 전송된 값이 27도 보다 낮으면

온도센서를 사용자가 잡지 않은 것으로 인식하여 라즈베리 파이에 연결 되어 있는 LED를 끄고

시리얼 통신을 통해 2번 아두이노에 1이라는 시그널을 전송합니다.

2번 아두이노는 라즈베리 파이로부터 1이라는 시그널을 받으면 서보모터의 각도를 30도로 변경합니다.

실제로 실험한 결과 사용자가 온도센서를 잡으면 온도센서에서 측정한 온도가 서서히 올라감을 라즈베리 파이 콘솔창을 통해 확인 할 수 있고, 온도가 27도가 넘어가면 라즈베리 파이에 연결된 LED가 켜짐과 동시에 서보모터의 각도가 60도로 변경되는 것을 확인 할 수 있었습니다.

또한 온도센서에서 손을 떼면 온도센서에서 측정되는 온도가 서서히 낮아짐을 라즈베리 파이 콘솔창을 통해 확인 할 수 있으며 27도 아래로 내려가는 순간 라즈베리 파이에 연결된 LED가 꺼지고 동시에 서보모터의 각도가 30도로 변경됨을 확인할 수 있었습니다.

url: <https://youtu.be/7bZbVdkz-Cw>