임베디드시스템프로그래밍2 기말 프로젝트

#17

HOME CONTROLLER

201912436 정유정

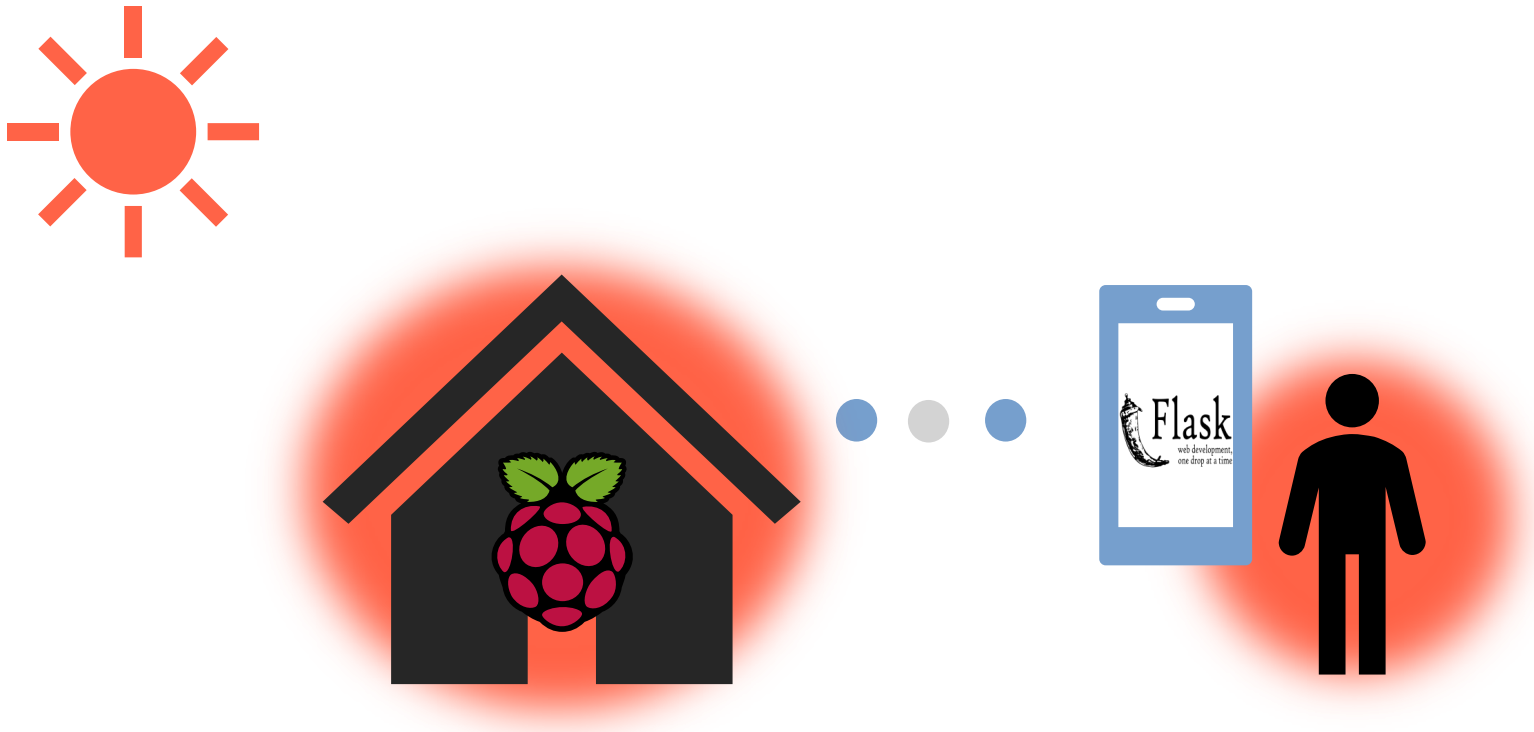
INDEX

1. HOME CONTROLLER 소개

IDEA

1. HOME CONTROLLER 소개

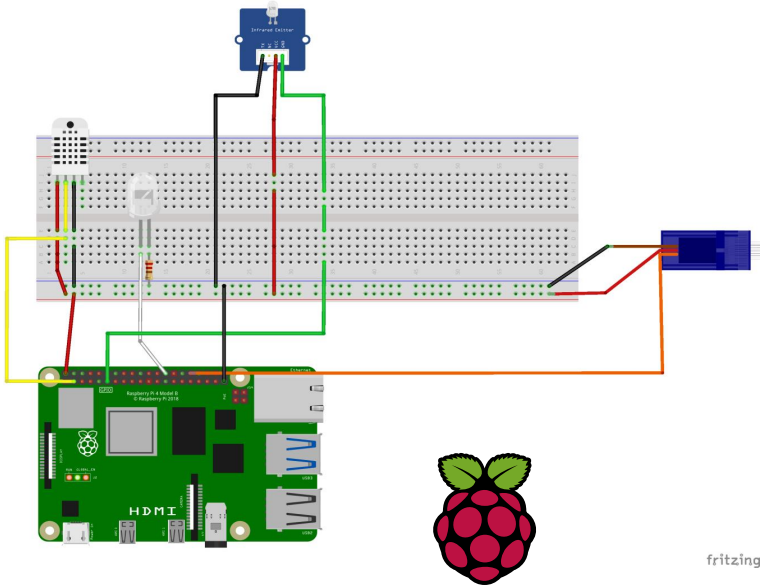
IDEA



1. HOME CONTROLLER 소개

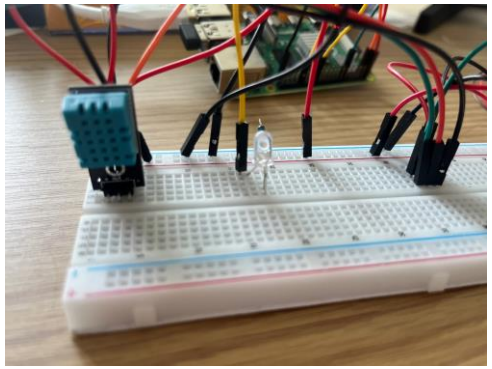
구현

Flask
web development,
one drop at a time



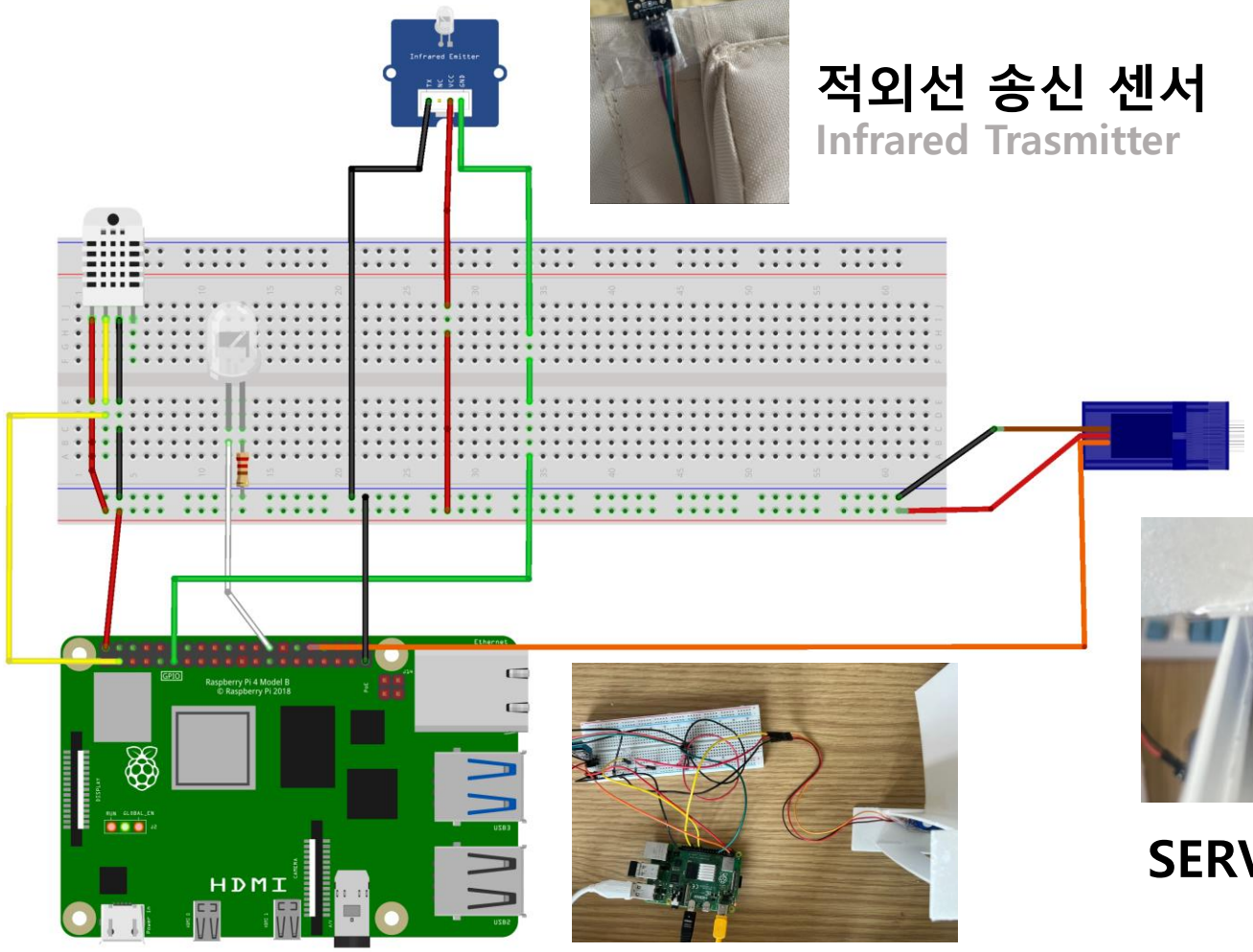
2. 회로

적외선 송신 센서
Infrared Trasmitter



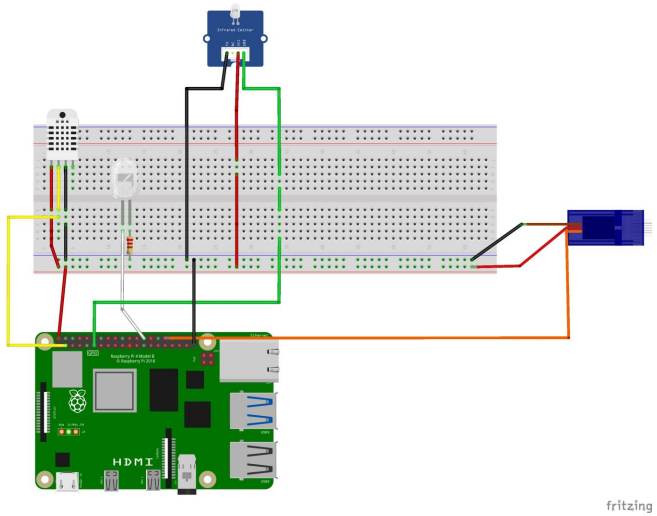
온습도 센서
DHT-11

LED



SERVO MOTOR

2. 회로



	5V	
	GND 0V	
	humSensorPin	라이브러리 사용
	LED_PIN	GPIO.OUT
	IR_PIN	GPIO.OUT
	SERVO_PIN	GPIO.OUT

remote.py

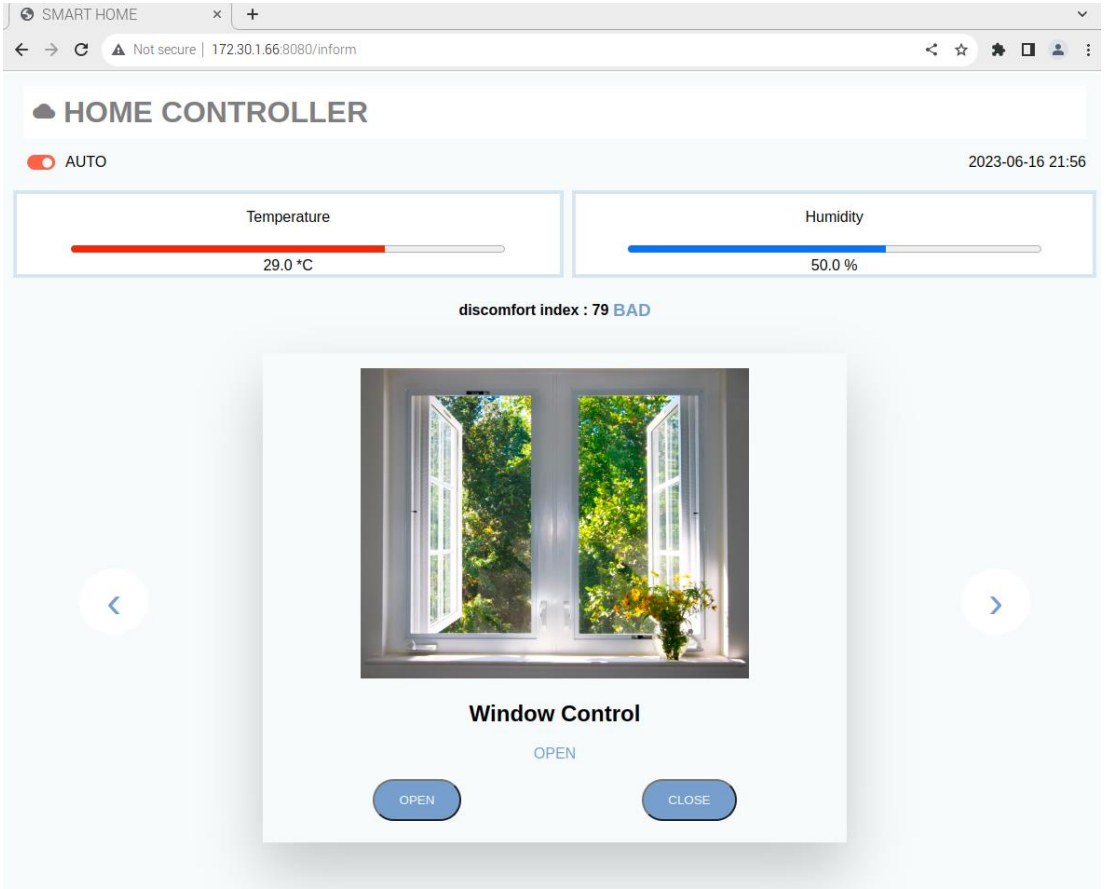
```python
from flask import Flask,render_template
from os import system as cmd
from irControl import control
from time import sleep

import Adafruit_DHT
import datetime
import RPi.GPIO as GPIO

humSensor = Adafruit_DHT.DHT11
humSensorPin = 2      #BCM, phy = 3
LED_PIN = 26          #phy
irReceiver = 12       #phy
IR_PIN = 11       #phy
SERVO_PIN= 32         #phy

hum, tem = Adafruit_DHT.read_retry(humSensor, humSensorPin) #read data and store
app = Flask(__name__)

GPIO.setmode(GPIO.BOARD)
GPIO.setup(LED_PIN, GPIO.OUT, initial=GPIO.LOW) #LED OFF init
GPIO.setup(SERVO_PIN, GPIO.OUT) #SERVO PIN init
GPIO.setup(irReceiver, GPIO.IN) #ir sensor PIN init
GPIO.setup(IR_PIN, GPIO.OUT)

servo = GPIO.PWM(SERVO_PIN, 50) #PWM mode, 50Hz
servo.start(0) #start servo, if duty=0, not run
servo_max_duty = 12
servo_min_duty = 3
```

온습도 센서 사용 라이브러리

핀 번호

핀 설정 & 초기화

3. 기능 및 주요코드



기능1 에어컨 ON/OFF

기능2 창문 OPEN/CLOSE

기능3 온습도 측정 & 자동 모드

3. 기능 및 주요코드

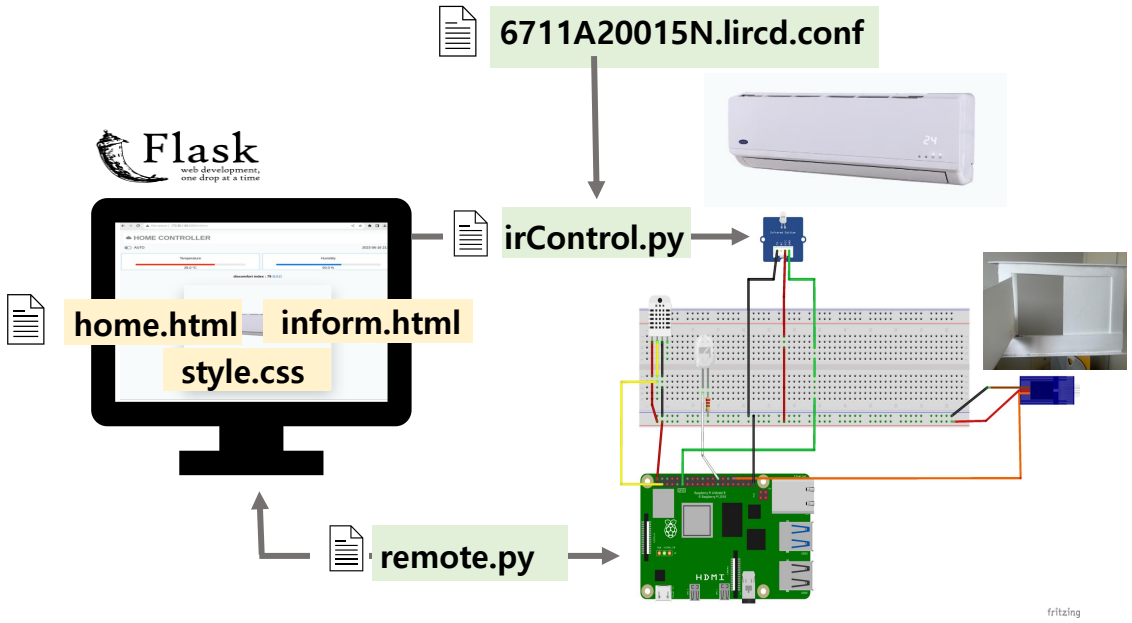
시스템 & 파일 구조

서버& 센서 동작 웹페이지

- /home/pi
  - 6711A20015N.lircd.conf
  - Project
    - irControl.py
    - remote.py
    - static
      - style.css
    - templates
      - home.html
      - inform.html

Flask 웹 서버 실행 &
이벤트에 맞는 센서 동작

<html> 전체 페이지 구성 + <script> 이벤트 발생 시 flask 동작



3. 기능 및 주요 코드

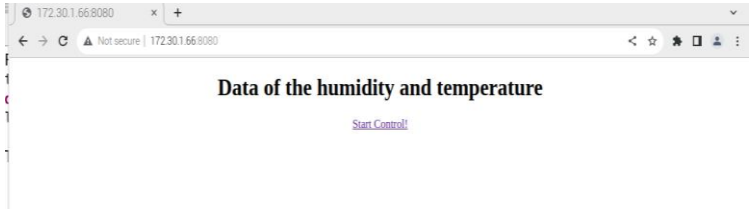
Flask : 웹서버 구현

remote.py

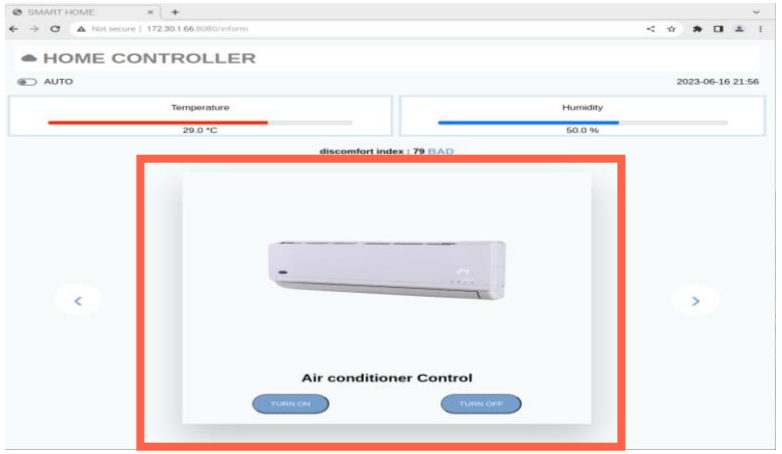
```python
from flask import Flask,render_template
from os import system as cmd
from irControl import control
from time import sleep
```

```python
app = Flask(__name__)
```

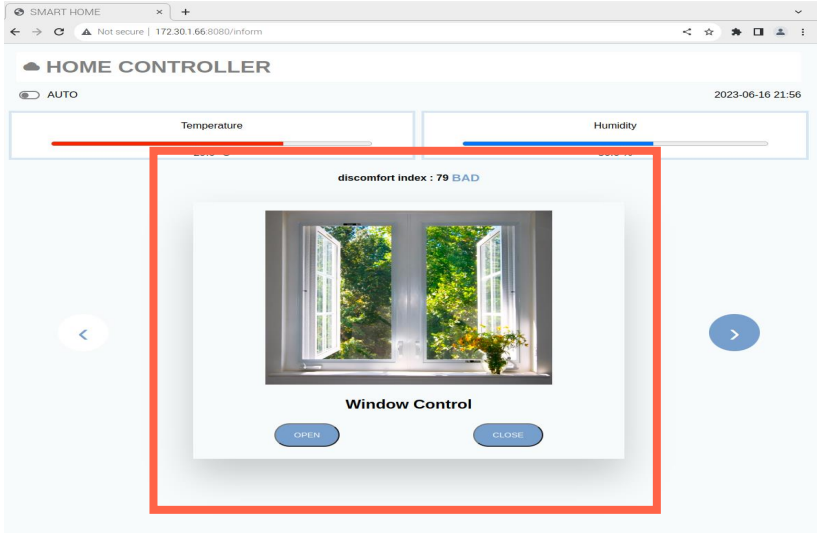
```python
@app.route('/')
def home():
    return render_template('home.html')
```




```python
@app.route('/aircon/on')
def aircon_on():
    try:
            GPIO.output(LED_PIN, GPIO.HIGH)
            control('POWER_ON')
            return render_template('inform.html')
    except KeyboardInterrupt:
        pass
    GPIO.cleanup()

@app.route('/aircon/off')
def aircon_off():
    try:
            GPIO.output(LED_PIN, GPIO.LOW)
            control('POWER_OFF')
            return render_template('inform.html')
    except KeyboardInterrupt:
        pass
    GPIO.cleanup()
```



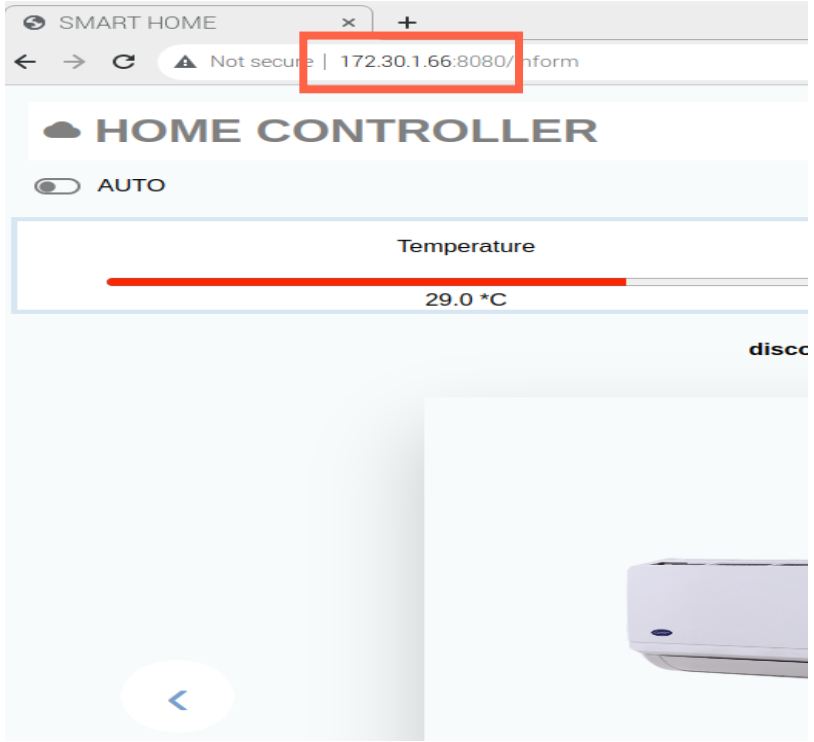

```python
@app.route('/window/open')
def window_open():
    try:
        set_servo_degree(70)
        return render_template('inform.html')
        sleep(1.3)
    except KeyboardInterrupt:
        GPIO.cleanup(SERVO_PIN)
        pass

@app.route('/window/close')
def window_close():
    try:
        set_servo_degree(180)
        return render_template('inform.html')
        sleep(1.3)
    except KeyboardInterrupt:
        GPIO.cleanup(SERVO_PIN)
        pass

if __name__=="__main__":
    app.run(host
            ="0.0.0.0", port = "8080")

GPIO.cleanup()
```



3. 기능 및 주요 코드

Flask : 웹서버 구현 – 포트 포워딩(port forwarding)

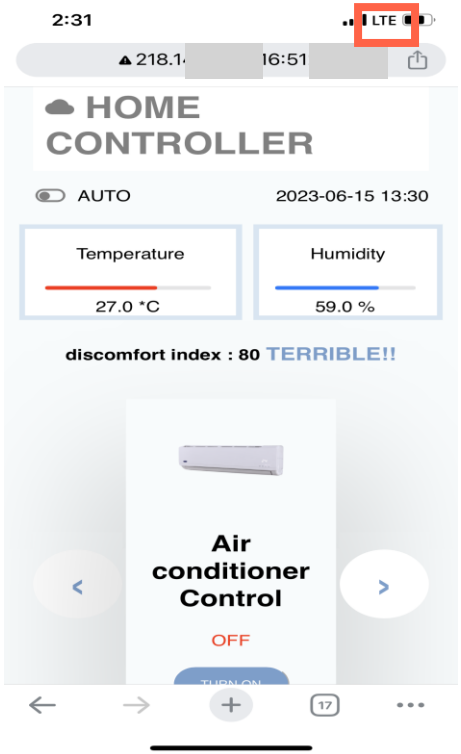
내부IP : 8080



연결

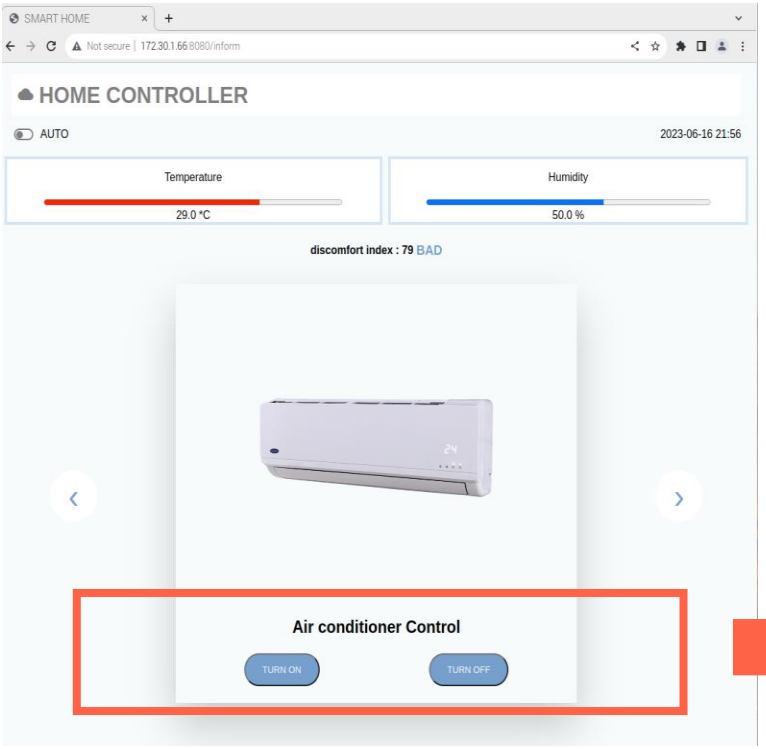
ip 주소 확인

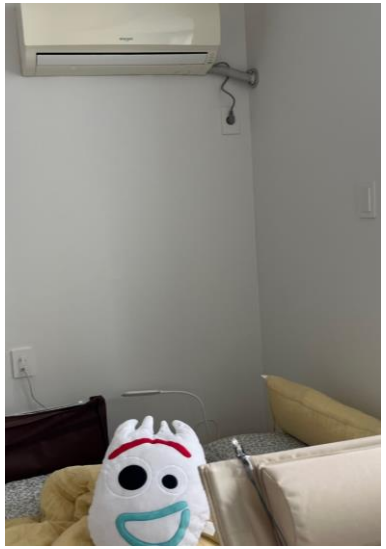
외부IP : 외부 포트번호

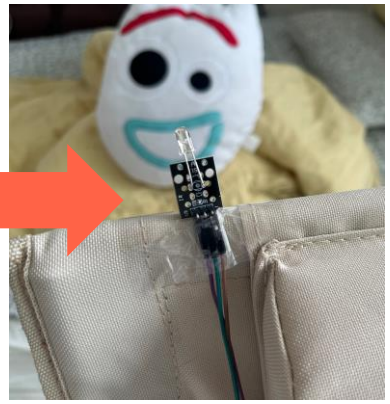


3. 기능 및 주요 코드

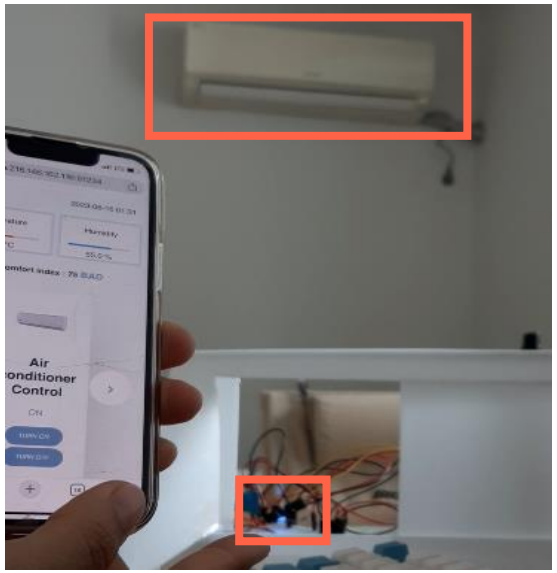
기능1 : 에어컨 ON/OFF



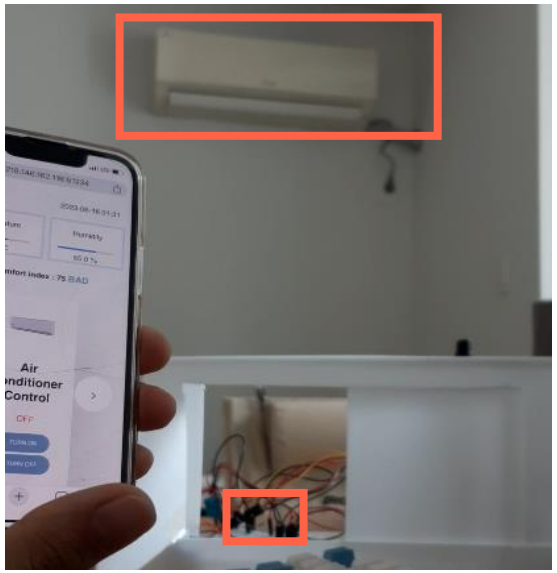




적외선 센서 – 신호 보내기



에어컨 ON
LED ON



에어컨 OFF
LED OFF

3. 기능 및 주요 코드

기능1 : 에어컨 ON/OFF 코드

inform.html

① 버튼 클릭

```html
<div class="card" id="airCard">
    <img src="https://external-content.duckduckgo.com/iu/?u=htt
    <div class="container">
        <h2> Air conditioner Control </h2>
        <p id="isOn">{{isOn}}</p>
        <div class="Btn">
            <button onclick="aircon_on()"> TURN ON</button>
            <button onclick="aircon_off()"> TURN OFF</button>
        </div>
    </div>
</div>
```

② 함수 호출 후 웹페이지 에어컨 상태 변경

```html
function aircon_on(){
    fetch("/aircon/on")
    .then(response=>response.text())
    .then(data=>{
        document.querySelector("#isOn").textContent="ON";
        document.querySelector("#isOn").style.color="#769FCD";
    });
}

function aircon_off(){
    fetch("/aircon/off")
    .then(response=>response.text())
    .then(data=>{
        document.querySelector("#isOn").textContent="OFF";
        document.querySelector("#isOn").style.color="#FF2400";
    });
}
```

remote.py

③ 웹페이지 & 함수 호출

```python
from flask import Flask,render_template
from os import system as cmd
from irControl import control
from time import sleep
```

```python
@app.route('/aircon/on')
def aircon_on():
    try:
            GPIO.output(LED_PIN, GPIO.HIGH)
            control('POWER_ON')
            return render_template('inform.html')
    except KeyboardInterrupt:
        pass
    GPIO.cleanup()

@app.route('/aircon/off')
def aircon_off():
    try:
            GPIO.output(LED_PIN, GPIO.LOW)
            control('POWER_OFF')
            return render_template('inform.html')
    except KeyboardInterrupt:
        pass
    GPIO.cleanup()
```

```python
@app.route('/window/open')
def window_open():
    try:
        set_servo_degree(70)
        return render_template('inform.html')
        sleep(1.3)
    except KeyboardInterrupt:
        GPIO.cleanup(SERVO_PIN)
        pass

@app.route('/window/close')
def window_close():
    try:
        set_servo_degree(180)
        return render_template('inform.html')
        sleep(1.3)
    except KeyboardInterrupt:
        GPIO.cleanup(SERVO_PIN)
        pass
```

irControl.py

④ 센서 동작

emote.py | irControl.py

```python
from os import system as cmd

def control(keyName):
    cmd("irsend SEND_ONCE LGE_6711A20015N " + __decode(keyName))

def __decode(keyName):
    POWER_ON="UN-JEON/JEONG-JI_18" #POWER ON setting 18 degree

    POWER_OFF="UN-JEON/JEONG-JI_OFF" #"0xC0051" #UN-JEON/JEONG-JI_OFF

    print("air conditioner " + keyName)
    return eval(keyName)
```

3. 기능 및 주요 코드

기능1 : 에어컨 ON/OFF 코드

irControl.py keyName 디코드 & 에어컨 전원 ON/OFF 센서 송신 명령어 cmd에 전달

④ 센서 동작

emote.py ✖ irControl.py ✖

```python
from os import system as cmd

def control(keyName):
    cmd("irsend SEND_ONCE LGE_6711A20015N " + __decode(keyName))
```

사용한 리모컨 모델명

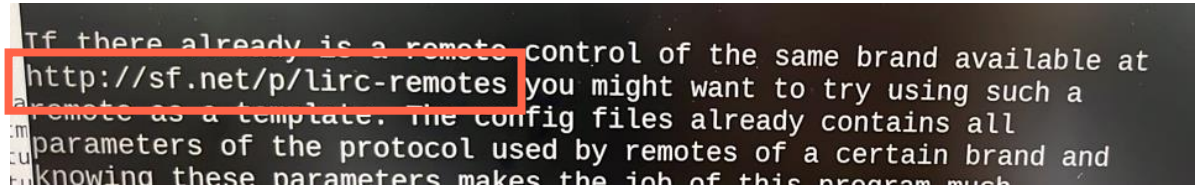
적외선 신호 송신

remote.py에서 받아온 변수

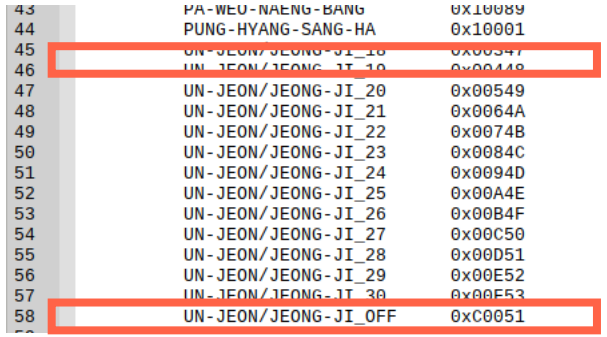
```python
def __decode(keyName):
    POWER_ON="UN-JEON/JEONG-JI_18" #POWER ON setting 18 degree

    POWER_OFF="UN-JEON/JEONG-JI_OFF" #"0xC0051" #UN-JEON/JEONG-JI_OFF

    print("air conditioner " + keyName)
    return eval(keyName)
```

*Linux Infrared Remote Control

6711A20015N.lircd.conf LG 에어컨 리모컨 lirc 정보 & key code



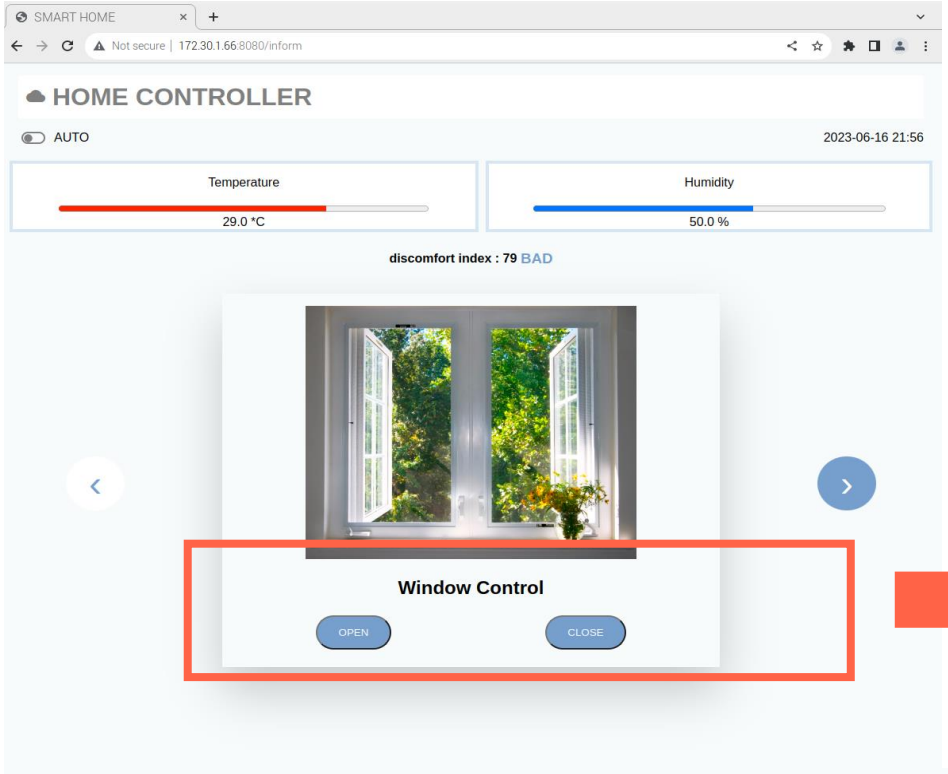
```
# this config file was automatically generated
# using lirc-0.8.1-CVS(default) on Sun Jul 23 17:16:07 2006
#
# contributed by Christoph Bartelmus
#
# brand:                    LG
# model no. of remote control:  6711A20015N
# devices being controlled by this remote: Six Sigma air conditioner

begin remote

  name  LGE_6711A20015N
  bits           20
  flags SPACE_ENC|CONST_LENGTH
  eps            30
  aeps          100

  header       8576  4224
  one           640  1664
  zero          640   640
  ptrail        640
  pre_data_bits   8
  pre_data       0x88
#actually this remote does not send any repeat signals!
  gap          1000000
  toggle_bit      0


      begin codes
          UN-JEON-SEON-TAEK_2      0x09801
          UN-JEON-SEON-TAEK_1      0x09845
          JEOL-JEON                0xC0095
          GEO-JIM-YE-YAK_1         0x903C8
          GEO-JIM-YE-YAK_2         0x90788
          GEO-JIM-YE-YAK_3         0x90B48
```

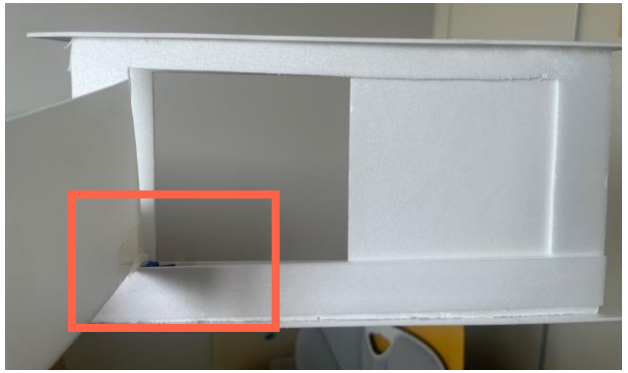


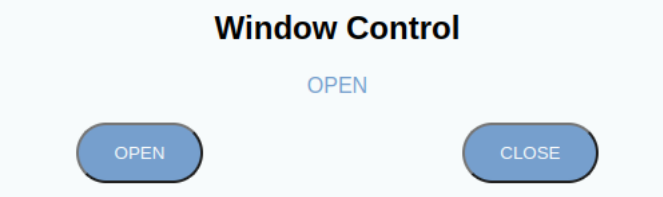
```
PA-WEO-NAENG-BANG        0x10089
PUNG-HYANG-SANG-HA       0x10001
UN-JEON/JEONG-JI_18      0x00547
UN-JEON/JEONG-JI_19      0x00448
UN-JEON/JEONG-JI_20      0x00549
UN-JEON/JEONG-JI_21      0x0064A
UN-JEON/JEONG-JI_22      0x0074B
UN-JEON/JEONG-JI_23      0x0084C
UN-JEON/JEONG-JI_24      0x0094D
UN-JEON/JEONG-JI_25      0x00A4E
UN-JEON/JEONG-JI_26      0x00B4F
UN-JEON/JEONG-JI_27      0x00C50
UN-JEON/JEONG-JI_28      0x00D51
UN-JEON/JEONG-JI_29      0x00E52
UN-JEON/JEONG-JI_30      0x00E53
UN-JEON/JEONG-JI_OFF     0xC0051
```

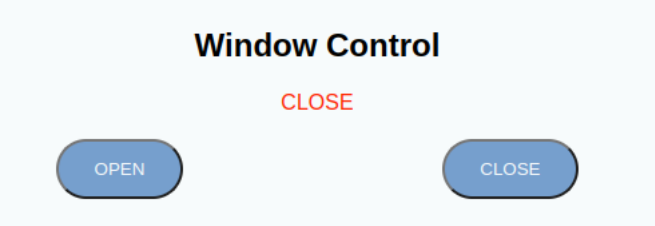
3. 기능 및 주요 코드

기능2 : 창문 OPEN/CLOSE









3. 기능 및 주요 코드

기능2 : 창문 OPEN/CLOSE 코드

inform.html

① 버튼 클릭

```html
<div class="card" id="winCard">
    <img src="https://external-content.duckduckgo.com/iu/?u=htt
    <div class="container">
        <h2> Window Control </h2>
        <p id="isOpen">{{isOpen}}</p>
        <div class="Btn">
            <button onclick="window_open()">OPEN</button>
            <button onclick="window_close()">CLOSE</button>
        </div>
    </div>
</div>
```

② 함수 호출 후 웹페이지 창문 상태 변경

```javascript
function window_open(){
    fetch("/window/open")
    .then(response=>response.text())
    .then(data=>{
        document.querySelector("#isOpen").textContent="OPEN";
        document.querySelector("#isOpen").style.color="#769FCD";
    });
}

function window_close(){
    fetch("/window/close")
    .then(response=>response.text())
    .then(data=>{
        document.querySelector("#isOpen").textContent="CLOSE";
        document.querySelector("#isOpen").style.color="#FF2400";
    });
}
```

remote.py

③ 웹페이지 & 함수 호출

```python
from flask import Flask,render_template
from os import system as cmd
from irControl import control
from time import sleep
```

```python
servo = GPIO.PWM(SERVO_PIN, 50) #PWM mode, 50Hz
servo.start(0) #start servo, if duty=0, not run
servo_max_duty = 12
servo_min_duty = 3
```

analog -> digital

```python
def set_servo_degree(degree):
    if degree > 180:
            degree = 180    #max=180
    elif degree < 0:
            degree = 0  #min=0

    duty = servo_min_duty + (degree*(servo_max_duty-servo_min_duty)/180.0)
    GPIO.setup(SERVO_PIN, GPIO.OUT)
    servo.ChangeDutyCycle(duty)
    sleep(0.7)
```

duty비 -> 각도

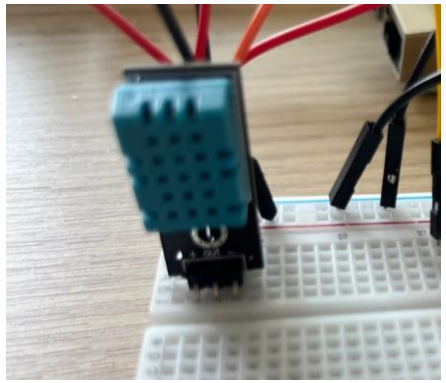
```python
@app.route('/window/open')
def window_open():
    try:
        set_servo_degree(70)
        return render_template('inform.html')
        sleep(1.3)
    except KeyboardInterrupt:
        GPIO.cleanup(SERVO_PIN)
        pass

@app.route('/window/close')
def window_close():
    try:
        set_servo_degree(180)
        return render_template('inform.html')
        sleep(1.3)
    except KeyboardInterrupt:
        GPIO.cleanup(SERVO_PIN)
        pass

if __name__=="__main__":
    app.run(host
            ="0.0.0.0", port = "8080")

GPIO.cleanup()
```

④ 센서 동작

3. 기능 및 주요 코드

기능3 : 온습도 측정 & 자동모드

불쾌 지수 & 예상 기분 표시



3. 기능 및 주요 코드

기능3 : 온습도 측정 & 자동모드 코드

remote.py

① 온도 습도 측정 후 변수에 저장

```python
from flask import Flask,render_template
from os import system as cmd
from irControl import control
from time import sleep

import Adafruit_DHT
import datetime
import RPi.GPIO as GPIO

humSensor = Adafruit_DHT.DHT11
humSensorPin = 2     #BCM, phy = 3
LED_PIN = 26         #phy
irReceiver = 12      #phy
IR_PIN = 11      #phy
SERVO_PIN= 32        #phy

hum, tem = Adafruit_DHT.read_retry(humSensor, humSensorPin) #read data and store
app = Flask(__name__)
```

Adafruit에서 제공하는 DHT 센서 library 사용

hum = 습도(humidity)
tem = 온도(temperature)

② 메인 페이지 데이터 표시

```python
@app.route('/inform')
def inform():
    now = datetime.datetime.now()
    timeString = now.strftime("%Y-%m-%d %H:%M")
    badFeel=(1.8*tem - 0.55*(1-hum)*(1.8*tem-26) + 32)/10.0
    feelString=str(int(badFeel))

    if badFeel>=68 and badFeel<=75 :
        feel='NOT GOOD'
    elif badFeel>75 and badFeel<=80:
        feel='BAD'
    elif badFeel>80:
        feel='TERRIBLE!!'
    else:
        feel='GOOD:)'

    sensorData = {
        'hum' : hum,
        'tem': tem,
        'feelIndex':feelString,
        'feel': feel,
        'time' : timeString
    }

    if hum is not None and tem is not None:
        return render_template('inform.html', **sensorData)
    else:
        return "<h1> Failed to get reading!!!! </h1>"
```

불쾌지수 공식 , 미국 기후학자 Thom 제안

불쾌 지수에 따른 기분 4단계

웹페이지로 넘길 데이터 저장

습도, 온도, 불쾌 지수, 기분, 시간

3. 기능 및 주요 코드

기능3 : 온습도 측정 & 자동모드 코드

inform.html

```html
<div class="fieldset">
<label>
    <input role="switch" type="checkbox" id="autoCheck"/>
    <span id="isAuto" style="color:black;">AUTO</span>
</label>
<p style="display:block;"> {{time}} </p>
</div>
<div class="information">
    <div class="temp">
        <p> Temperature </p>
        <progress id="temProgress" value={{tem}} min="-10" max="40" style="accent-color:#FF2400; width:80%"></progress>
        <br> {{tem}} *C </br>
    </div>
    <div class="hum">
        <p> Humidity </p>
        <progress id="humProgress" value={{hum}} min="0" max="80" style="width:80%"></progress>
        <br> {{hum}} % </br>
    </div>
</div>
<div class="information">
    <h4 style="margin-right:0.3rem;"> discomfort index : {{ feelIndex }}</h4>
    <h3 style="color:#769FCD" id="feel">{{ feel }}</h3>
</div>
```

자동 모드 선택

④ 자동 모드

```cpp
const checkbox = document.querySelector("#autoCheck");
checkbox.addEventListener("change", function() {
    const feel = document.querySelector("#feel").textContent;
    console.log(feel);
    if (checkbox.checked) {
            if (feel === "TERRIBLE!!") {
                aircon_on();
                window_close();
            } else if (feel === "GOOD:)") {
                aircon_off();
                window_open();
            }
    }else {
            document.querySelector("#isAuto").style.color = "#cccccc";
    }
});
```

③ 데이터 표시 &

값에 따른 progress Bar 표시

HOME CONTROLLER
자동 모드 시연 영상

Q&A