



임베디드 IoT 프로그래밍 기말 프로젝트

3조
201720976 장혜린
201720980 조정아

username : pi
password : 7589

목차

- 프로젝트 주제
- 개발 환경 설정
- HW 구성도 및 사진
- 프로그램 설명
- 실행결과
- 참여

프로젝트 주제



온습도 센서와 LED, 푸시 버튼을 이용한 날씨 보기

푸시 버튼을 누르면
LED에 불이 들어오고 온습도 센서가 작동,
센서 출력 값이 웹 페이지에 보여짐

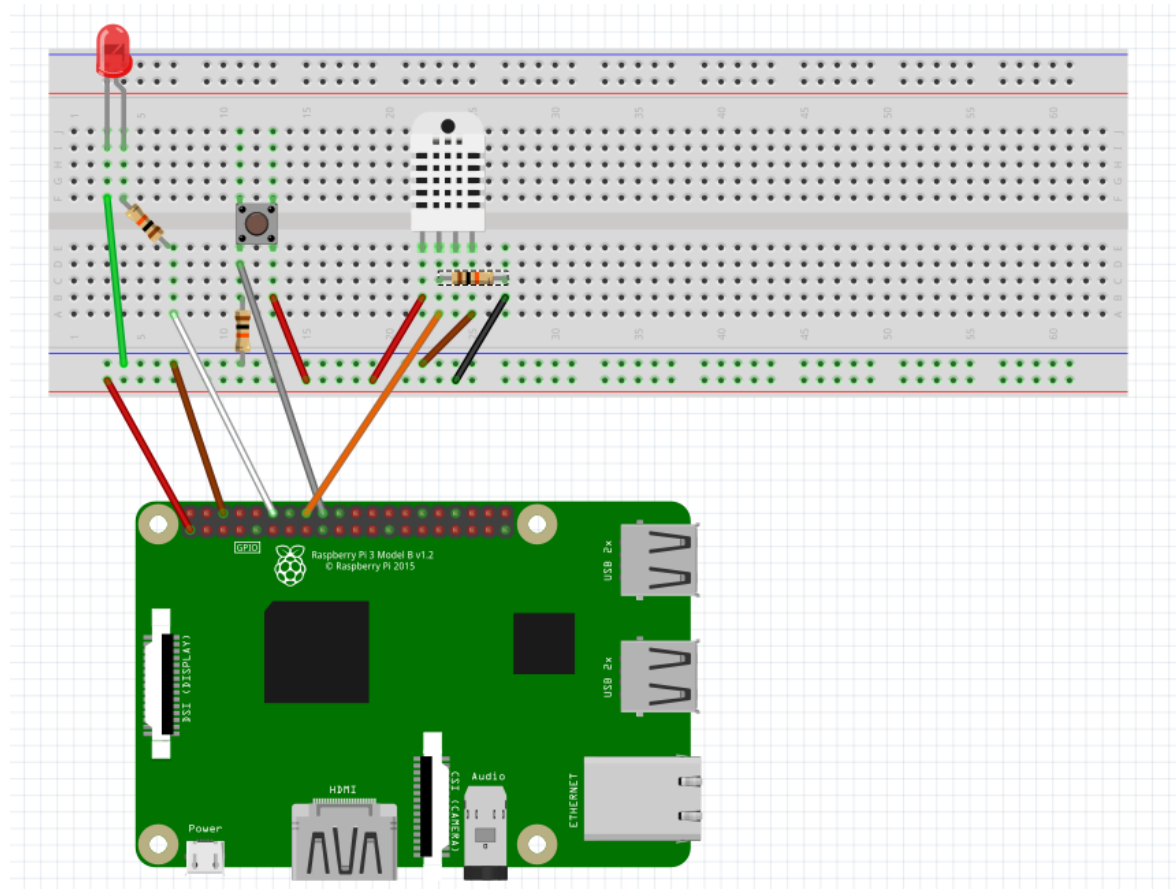
개발 환경 설정

- Flask 설치
 - Python으로 작성된 웹 프레임워크
 - `sudo apt-get install python-pip`
 - `sudo pip install flask`
 - localhost:5000 사용
- 부품
 - 저항 10K Ω 3개
 - 온습도 센서
 - LED센서
 - 푸시 버튼
 - 브레드 보드
 - 점프 케이블

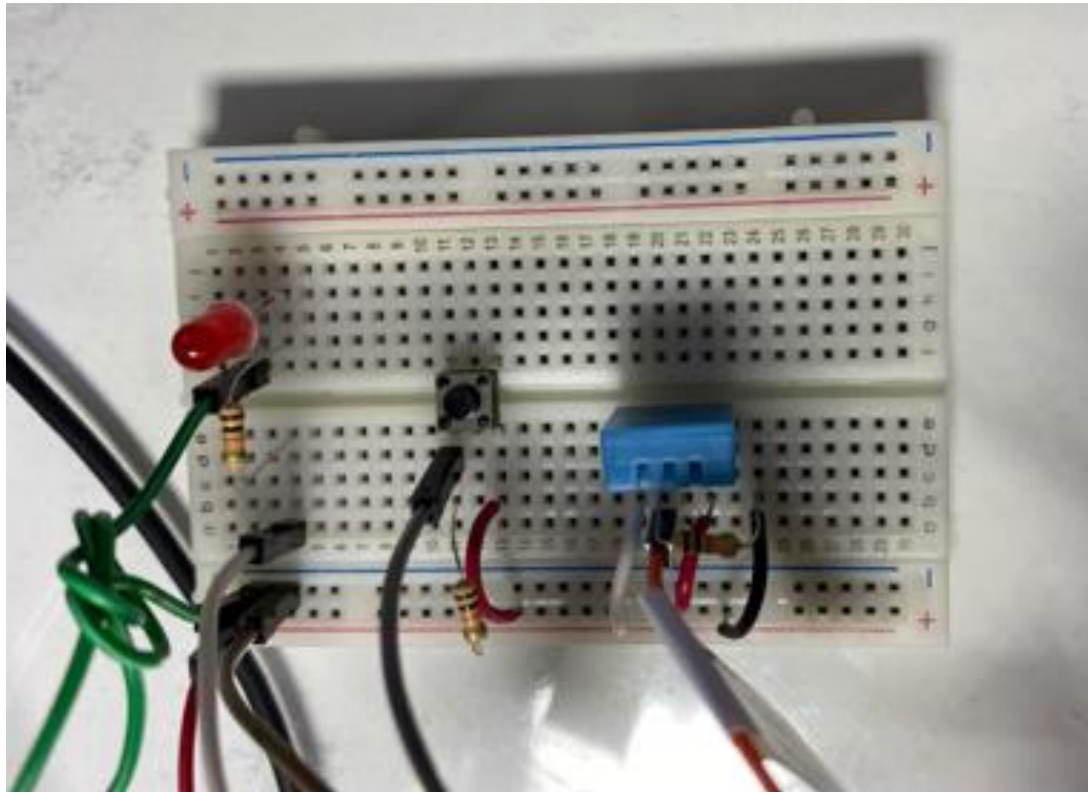


Flask

HW 구성도 및 사진



HW 구성도 및 사진



프로그램 설명 - index2.html(1)

```
newf.py * index2.html *
1 <!DOCTYPE html>
2 <html lang="ko" dir="ltr">
3   <head>
4     <meta charset = "utf-8"> → 한글 설정
5     <title> Local_NOW </title>
6     <style>
7       table{
8         width: 100%
9         border: 1px solid #444444;
10      }
11      th, td{
12        border: 1px solid #444444;
13      }
14      h1 {
15        margin: 20px;
16        padding: 20px 0px;
17        text-align: center;
18        text-transform: uppercase;
19        font-family: "Arial Black", sans-serif;
20        font-size: 60px;
21        font-weight: bold;
22      }
23      .s1{
24        background-color: #0d47a1;
25        color: #ffffff;
26        text-shadow: 4px 4px 0px #bdbdbd;
27      }
28    </style>
29  </head>
30
31  <body onload="showClock()" style ="background-color:#d7d3d1"> → showClock() 호출(자바스크립트)
32    <center>
33
34      <br><h1 class="s1"> ! Press The Button ! </h1> <hr> → 1번째 제목은 Press The Button
35    </center>
36
37    <table border=1 width=700 align="center" cellpadding="3"> → 표를 만드는 태그 <table>
38
39    <tr> → 행은 <td>, 열은 <tr>로 열 안에 행을 작성함
40      <td width="696" height="30" colspan="2" bgcolor="#bbdefb">
41        <font font="italic">
42          <h2 align="center"> ( ( (●'w'●) ) ) Temperature And Humidity </h2> → 첫번째 행에 들어갈 입력
43        </font>
```



newf.py * index2.html ✕

```

42     <h2 align="center"> (( \'w\'•) ) Temperature And Humidity </h2>
43     </font>
44     </td>
45
46 </tr>
47
48 <tr>
49     <td width="40%" hight="15" bgcolor="#e3f2fd" align="center">
50         <font color="navy">
51             <h2> Local DATE </h2> → 2번째 행, 1번째 열에 들어갈 입력
52         </font>
53     </td>
54
55     <td width="60%" align="center" bgcolor="#e3f2fd">
56         <font color="#8a3213">
57             <h2> {{time}} </h2> → 2번째 행, 2번째 열에 들어갈 입력
58         </font>
59     </td>
60
61 </tr>
62
63 <tr>
64     <td width="40%" hight="15" bgcolor="#e3f2fd" align="center">
65         <font color=navy> <h2> Local Time </h2></font> → 3번째 행, 1번째 열에 들어갈 입력
66     </td>
67
68     <td width="60%" align="center" bgcolor="#e3f2fd">
69         <h2> <div id="divClock" class="clock" style ="color:#8a3213"> </div> </h2> → 3번째 행, 2번째 열에 들어갈 입력
70     </td>
71 </tr>
72
73 <tr>
74     <td weight=696 align="center" hight="15" bgcolor="#e3f2fd" colspan="2">
75         <div style = "color:red">
76             <h3 align="center" bgcolor="#e3f2fd"> {{DHT}} </h3> → 4번째 행에 들어갈 입력(DHT라는 변수는
77         </div>                                     newf.py파일에 받아옴)
78     </td>
79
80 </table> → 테이블 표 마감
81
82 <script language="javascript">
83
84     function showClock(){

```


프로그램 설명 - index2.html(3)

```
newf.py * index2.html *
68         <td width="60%" align="center" bgcolor="#e3f2fd">
69             <h2> <div id="divClock" class="clock" style="color:#8a3213"> </div> </h2>
70         </td>
71     </tr>
72
73     <tr>
74         <td weight=696 align="center" hight="15" bgcolor="#e3f2fd" colspan="2">
75             <div style="color:red">
76                 <h3 align="center" bgcolor="#e3f2fd"> {{DHT}} </h3>
77             </div>
78         </td>
79     </tr>
80 </table>
81
82 <script language="javascript"> → 자바스크립트 선언
83
84     function showClock(){
85         var currentDate = new Date();
86         var divClock = document.getElementById("divClock");
87
88         var msg = currentDate.getHours()+"시 "
89         msg += currentDate.getMinutes()+"분 ";
90         msg += currentDate.getSeconds()+"초 ";
91
92         divClock.innerText = msg;
93         setTimeout(showClock,1000); → 콜백함수로 지연시간 1초(1000)뒤 showClock 함수를 실행함 따라서 1초마다
94     }                                     함수가 실행되어 실시간으로 시간 바뀜을 확인할 수 있음
95 </script>
96
97     <center>
98     <hr>
99     <table border=0 width=0>
100         <tr>
101             <td align=center>
102                 <input type="button" value="▶ NEW Load ◀" onclick="window.location.href=window.location.href">
103             </td>
104         </tr>
105     </table>
106
107     </center>
108 </body>
109 </html> → html 코드 닫기
110
```

→ showClock함수 : 현재 시간, 날짜 출력

→ 콜백함수로 지연시간 1초(1000)뒤 showClock 함수를 실행함 따라서 1초마다
함수가 실행되어 실시간으로 시간 바뀜을 확인할 수 있음

→ 페이지 자동 새로고침

→ html 코드 닫기

프로그램 설명 - newf.py(1)

```
newf.py * index2.html *  
1 from flask import Flask, render_template  
2 import RPi.GPIO as GPIO  
3 import Adafruit_DHT  
4 import datetime  
5  
6 app = Flask(__name__) → Flask와 센서, GPIO, 날짜 시간을 사용하기 위한 설정  
7 GPIO.setmode(GPIO.BCM)  
8  
9 sensor = Adafruit_DHT.DHT11 → DHT센서 설정, GPIO핀 23번에 설정  
10 pin = 23  
11 @app.route('/') → Root와 연결(localhost:5000)  
12 def index():  
13     try:  
14         localtime = datetime.datetime.now() → 현재 날짜 출력  
15         strTime = localtime.strftime("%Y. %m. %d")  
16         sensor = Adafruit_DHT.DHT11 → DHT11센서 값을 sensor에 저장  
17         GPIO.setup(24, GPIO.IN)  
18         GPIO.setup(18, GPIO.OUT) → GPIO 24번에 스위치(IN), 18번에 LED(OUT) 연결  
19  
20         value = GPIO.input(24) → value에 24번 핀을 사용하는 스위치 값 저장  
21         print("%s"%value)  
22         humidity, temperature = Adafruit_DHT.read_retry(sensor, pin) → Sensor인 DHT센서와, pin인 23(GPIO)으로 구한 값을  
23                                     humidity와 temperature에 넣음  
24  
25         if value==True: → value가 True(1)일 때는 스위치가 눌러졌을 때를 말함  
26             GPIO.output(18, True) → 스위치가 눌러지면 18번 핀에 연결된 LED가 켜짐  
27             print('현재 온도는 {0:0.1f}*C , 습도는 {1:0.1f}% 입니다!'.format(temperature, humidity))  
28             #dht=format(temperature, humidity)  
29             if humidity is not None and temperature is not None: → Humidity나 temperature의 값이 있을 때(not none)  
30                 dht="[Now Temperature: ' + '{0:0.1f}*C'.format(temperature) + ', Now Humidity: ' + '{0:0.1f}%'.format(humidity) + '"]"  
31             else:  
32                 dht="Failed to get reading. Try again!" → Humidity나 temperature의 값이 없을 때(none)  
33         else:  
34             GPIO.output(18, False) → value가 1이 아닐 때는 스위치가 눌러지지 않았을 때를 말함, value값이 0이므로 LED는 켜지지 않음  
35             dht="If you want to know Tempature and Humidity, PRESS THE BUTTON ONE MORE! "  
36             → 스위치를 다시 누르라고 알림  
37     except:  
38         #print("hi")  
39         dht = "Read error. Try again!" → showClock함수 : 현재 시간, 날짜 출력  
40  
41         DHT = {'time': strTime, 'DHT': dht} → 현재 날짜(strTime)값과 센서 값(dht)을 DHT에 저장  
42  
43         return render_template('index2.html', **DHT) → Index2.html에 DHT값을 리턴하여 index2.html에서 웹 페이지로 출력  
44     GPIO.cleanup()
```

프로그램 설명 - newf.py(2)

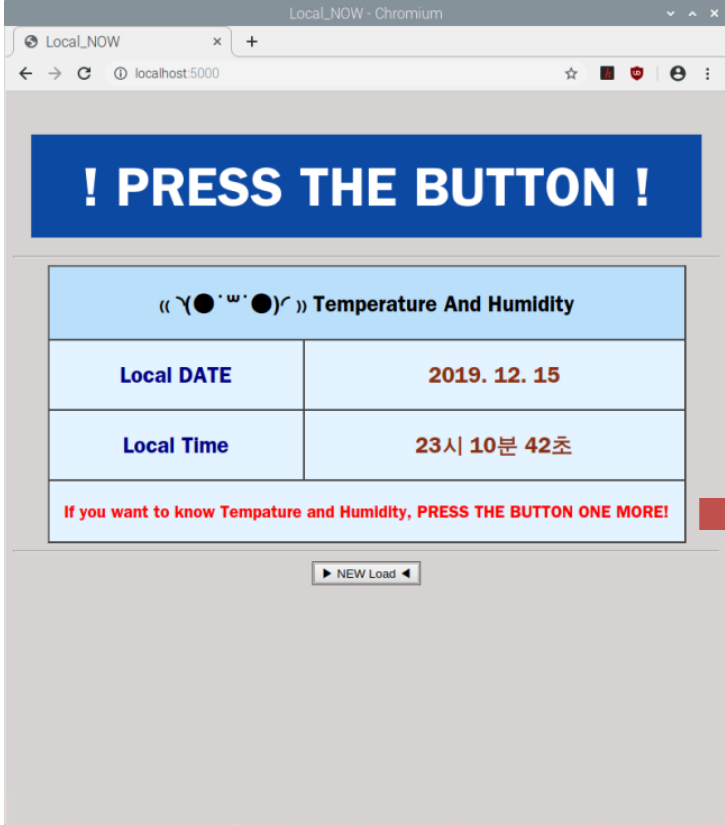
newf.py * index2.html *

```
5
6 app = Flask(__name__)
7 GPIO.setmode(GPIO.BCM)
8
9 sensor = Adafruit_DHT.DHT11
10 pin = 23
11 @app.route('/')
12 def index():
13     try:
14         localtime = datetime.datetime.now()
15         strTime = localtime.strftime("%Y. %m. %d")
16         sensor = Adafruit_DHT.DHT11
17         GPIO.setup(24,GPIO.IN)
18         GPIO.setup(18,GPIO.OUT)
19
20         value = GPIO.input(24)
21         print("%s"%value)
22         humidity, temperature = Adafruit_DHT.read_retry(sensor,pin)
23
24         if value==True:
25             GPIO.output(18,True)
26             print ('현재 온도는 {0:0.1f}*C , 습도는 {1:0.1f}% 입니다!'.format(temperature, humidity))
27             #dht=format(temperature, humidity)
28             if humidity is not None and temperature is not None:
29                 dht="[Now Temperature:'" + "{0:0.1f}*C".format(temperature)+"', Now Humidity:'" + "{0:0.1f}%".format(humidity)+"']"
30             else:
31                 dht="Failed to get reading. Try again!"
32         else:
33             GPIO.output(18,False)
34             dht="If you want to know Tempature and Humidity, PRESS THE BUTTON ONE MORE! "
35
36     except:
37         #print("hi")
38         dht = "Read error. Try again!"
39
40     DHT = {'time':strTime, 'DHT':dht}
41
42     return render_template('index2.html',**DHT)
43 GPIO.cleanup()
44 ##GPIO.setup(18, GPIO.OUT)GPIO.cleanup()
45 if __name__ == '__main__':
46     app.run(host='0.0.0.0', debug=True)
47
```

여기서 print는 명령어 창에서 확인하려고 넣은 코드

GPIO 라이브러리/모듈이 점유한 리소스를 해제하는 기능, 이 함수가 없으면 다음 실행에 오류를 발생시킴

실행 결과



Local_NOW - Chromium

Local_NOW x +

localhost:5000

! PRESS THE BUTTON !

« (●`ω`●) » Temperature And Humidity

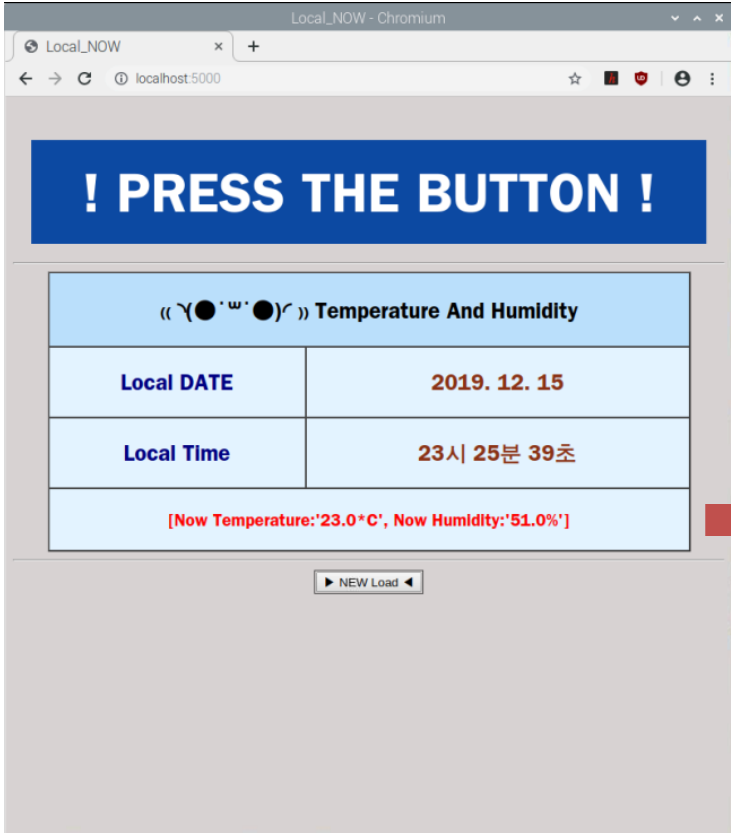
Local DATE	2019. 12. 15
Local Time	23시 10분 42초

If you want to know Tempature and Humidity, PRESS THE BUTTON ONE MORE!

▶ NEW Load ◀

버튼을 안 눌렀을 때

실행 결과



The screenshot shows a web browser window titled "Local_NOW - Chromium" with the address bar displaying "localhost:5000". The page content includes a large blue button with the text "! PRESS THE BUTTON !". Below this is a section titled "Temperature And Humidity" with a decorative header "((ㄹ●'ㅍ'●)ㄹ)". It contains a table with local date and time, and a status bar showing current temperature and humidity. A red arrow points to the status bar with the text "버튼을 눌렀을 때".

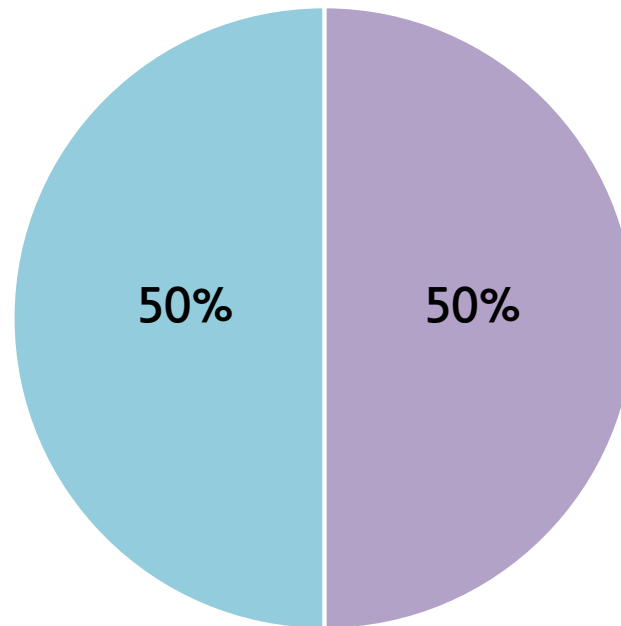
Temperature And Humidity	
Local DATE	2019. 12. 15
Local Time	23시 25분 39초
[Now Temperature:'23.0*C', Now Humidity:'51.0%']	

버튼을 눌렀을 때

참여



참여도



■ 장혜린 ■ 조정아



감사합니다 😊