설계과제1

[웹 서버 상에서 MQTT를 이용하여 NodeMCU 제어하기]

21900413 심정훈, 22100113 김성빈

목차

1. 서론
   1. 프로젝트 목표
   2. 사용된 장비들
   3. 프로그램 알고리즘
2. 본론
   1. Sensor 값 처리
   2. 값 display 형식
   3. 데이터 전달 형식
   4. Flask 웹서버
   5. MQTT 통신
   6. 추가기능
3. 한계점
4. 결론
5. 팀워크
6. 참조

서론

* 1. 프로젝트 목표

라즈베리 파이를 사용하여 웹 서버를 동작시키고 여기서 발생하는 이벤트(커맨드)들을 MQTT를 활용하여 NodeMCU에 전달하고 이를 통하여 actuator를 제어하고 NodeMCU는 sensor를 통해 측정된 값을 10초마다 라즈베리 파이로 전송하는 프로그램을 구현 했다. 이를 확장하여 MQTT 통신을 지원하는 다양한 Actuator를 웹 상에서 그 상태를 확인하거나 컨트롤 하는 것이 가능하다

* 1. 사용된 장비들:

Sensor:

DHT22

CDS

Actuator:

LED

RELAY (USB LED)

OLED DISPLAY

* 1. 프로그램 알고리즘

NodeMCU

1. CDS는 1초마다 측정, DHT22는 10초마다 측정
2. 라즈베리파이로 10초마다 MQTT 측정값 전송
3. MQTT로 오는 명령어에 따른 Actuator 제어

라즈베리 파이 (웹 서버)

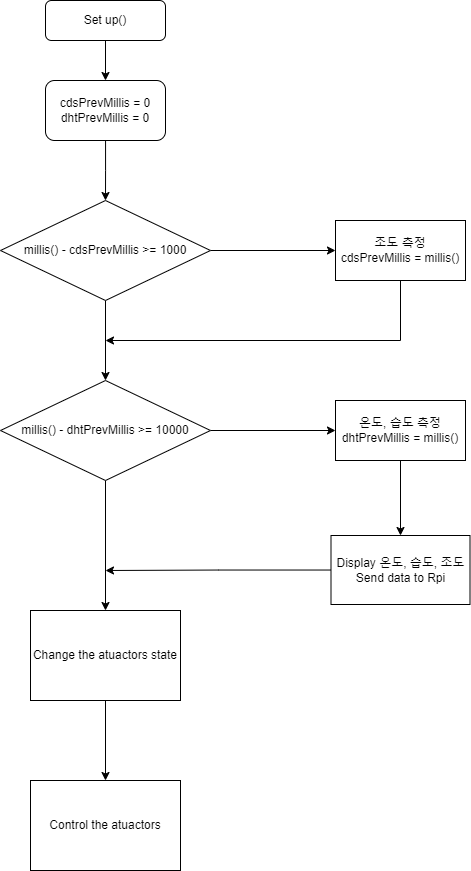
1. 명령어가 적혀있는 버튼 클릭 시 그에 대응되는 json데이터를 만들어 NodeMCU로 전송
2. MQTT를 통해 들어오는 data가 있다면 그 값을 화면에 표시 (10초마다 혹은 get 명령어 시)

명령어 종류

led, led\_on, led\_off

usb, usb\_on, usb\_off

get light intensity, temperature, humidity

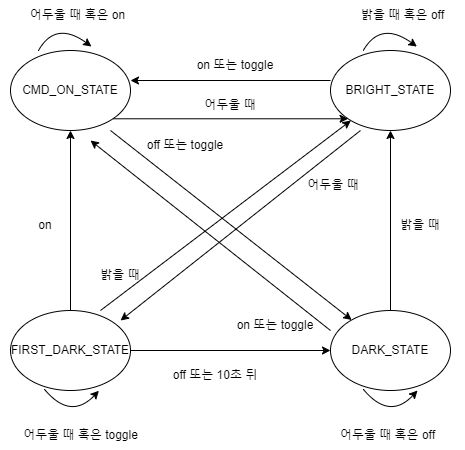
\* led, usb는 각 actuator에 대한 toggle을 의미 \* 

<NodeMCU Flow chart>

Actuator state

LED: HIGH, LOW

RELAY: 밑의 4가지 state를 가진다.



<제어/조도 데이터에 따른 USB LED FSM>

OLED\_DISPLAY: 10초마다 온도, 습도, 조도를 출력

본론

1. Sensor 값 처리

NodeMCU에서 센서들을 통해 온도, 습도, 조도 값을 읽어온다. 온도와 습도는 DHTesp[1] 외부 라이브러리를 사용하였다. DHT 타입은 22로 통일하여 사용하였다. 조도 값은 CDS 센서를 사용하여 analogRead()를 통하여 값을 읽어왔다.

1. 값 display 처리

그리고 10초 간격으로 측정한 값들을 OLED\_DISPLAY와 MQTT로 내보낸다. 10초 간격을 재기 위해 non-blocking 형식으로 코드를 작성하여 제어에 피해를 끼치지 않도록 작성하였다.

1. 데이터 전달 형식

측정 후 NodeMCU는 MQTT를 사용하여 값들을 통신 한다. 이때 주고 받는 데이터들의 개수와 종류가 다양하기에 JSON과 같은 형식을 주어 읽고 쓰기 편하게 구현하였다. 이를 위해 외부 라이브러리 Arduino\_JSON[2]을 사용하였다. 아래는, NodeMCU에서 보내는 데이터 총 3가지, {온도, 습도, 조도}를 한번에 보내는 형식이다.

{

“temp”: temperature value,

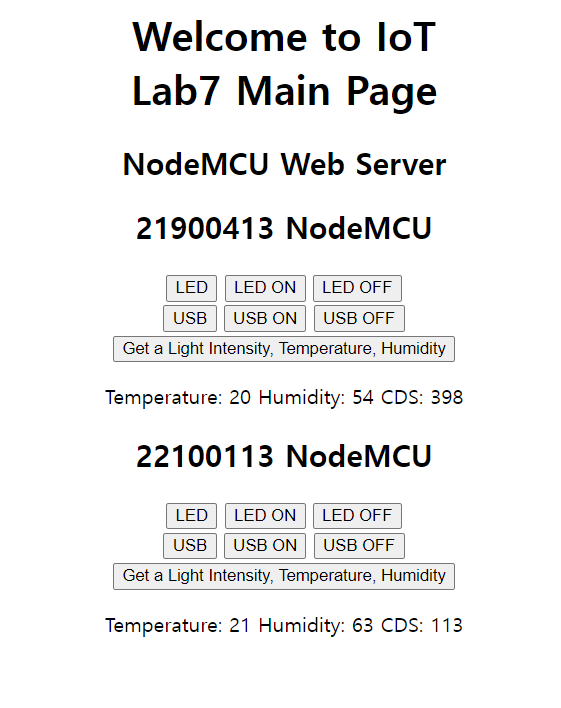
“hum”: humidity value,

“cds”: light\_intensity value

}

1. Flask 웹 서버

파이썬의 Flask 라이브러리를 사용하여 Web server를 구현하였으며 화면의 구성은 아래와 같다.



<Flask 라이브러리를 사용한 Web server 구동 화면>

1. MQTT 통신

2대의 NodeMCU로부터 MQTT 통신을 통해 주고받은 데이터를 화면에 보여준다. 이를 위해 2개의 서로 다른 topic을 MQTT를 통해 subscribe한 상태로 대기한다.

LED, USB LED는 각 actuator의 toggle을 의미한다.

Get Light intensity, temperature, humidity는 NodeMCU로 부터 데이터 측정을 요청하고 그 값을 화면에 보여준다.

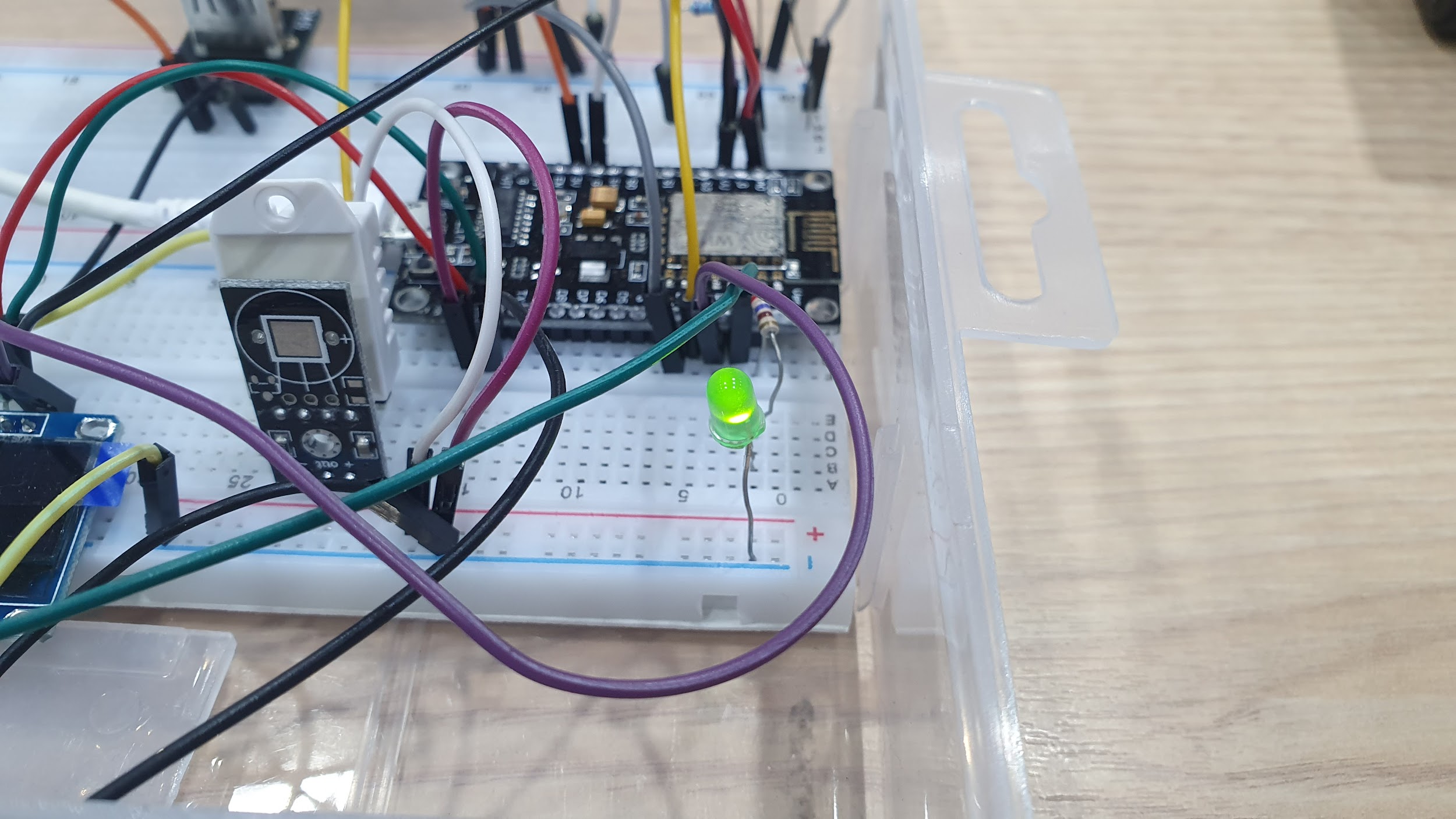
MQTT 통신은 발생한 이벤트를 key로하여 value는 1로 설정한 json 데이터로 publish한다.

{

“led\_on”: 1

}

이러한 값이 들어오면 NodeMCU에서 JSON 데이터를 파싱하여 actuator를 제어한다.



<MQTT를 통해 NodeMCU가 LED를 led\_on한 장면>

1. 추가기능

설계과제로 주어지지 않은, 추가 기능으로는 NodeMCU에서 센서 값을 보내는 시간 간격에 맞춰 html 페이지도 같은 간격으로 새로고침을 진행한다. 아래 코드[4]를 header에 추가하여 구현하였다.

<head>

…

<meta http-equiv="refresh" content="10">

…

</head>

한계점

NodeMCU에서 USB LED를 환경에 따라 상태를 조도 값에 따라 제어하기 위해 loop를 새로 돌때마다 analogRead() 함수를 사용한다. 하지만, 이럴 경우 NodeMCU가 와이파이에서 연결이 끊어지는 현상이 일어난다는 자료[3]가 많았고, 직접 했을 때도 같은 현상이 발생했다. 이를 극복하기 위해 조도 값 측정을 1초 마다 non-blocking 모드로 작동하도록 하였다.

결론

라즈베리 파이와 NodeMCU를 사용해 IoT 서버를 구축하는 과제를 진행하였다. 라즈베리 파이는 웹 서버를 동작시켜 사용자 지정 이벤트(커맨드)를 NodeMCU에게 전달하여 actuator들을 제어한다. NodeMCU는 스스로 sensor들을 통해 측정된 값이나 라즈베리 파이에게서 받은 커맨드에 따라 측정한 값을 MQTT로 라즈베리 파이로 전송한다. 이러한 상호작용이 가능한 IoT 서버를 구현하였다.

팀워크

라즈베리 파이 코드와 NodeMCU 코드 부분을 중점으로 나누어서 작업했다. (김성빈) NodeMCU를 이용해 JSON 데이터를 받아 다양한 actuator 제어와, 10초마다 publish하는 작업을 맡음. (심정훈) Flask 웹 서버와 html 코드를 주로 담당하며 JSON 형식을 이용해 MQTT통신 데이터로 사용자에게 html 값을 보여주는 작업을 함. 추가로 같이 NodeMCU의 한계점을 해결하기 위해 작업함. 보고서는 함께 작성을 진행하였고, (심정훈) Function flowchart, FSM diagram을 추가로 맡았고, (김성빈) 제어 작동 사진과 readme 작성으로 분배하였다.

많은 작업을 시간을 맞춰 만나서 작업하였기에 일의 분배는 비슷했다고 판단이 된다.

참조

[1] DHT sensor library for ESPx, beegee\_tokyo, ver.1.19, http://desire.giesecke.tk/index.php/2018/01/30/esp32-dht11/

[2] Arduino\_JSON, Arduino, ver.0.2.0, https://github.com/arduino-libraries/Arduino\_JSON

[3] NodeMCU running arduino - wifiDrops after reading analogRead, ESP8266 Community Forum, https://www.esp8266.com/viewtopic.php?p=22540

[4] HTML <meta> http-equiv Attribute, w3 Schools, https://www.w3schools.com/tags/att\_meta\_http\_equiv.asp