

Module 33: IoT Server 설계 – Rpi3 and nodeMCU

ECE30003 IoT시스템설계

조 윤 석

전산전자공학부

설계 프로젝트 1

❑ 목적

- NodeMCU와 raspberry pi를 이용하여 나만의 IoT 서버를 설계하고 구축하기

❑ 내용

- Web server는 raspberry pi에서 실행되며, Python3 Flask package 활용
- Sensor node는 NodeMCU 기반의 sensors/actuators 들로 구성됨(지금까지 수업 시간 및 숙제에 사용된 device들)
- Web server와 sensor node 간에는 MQTT protocol로 통신

❑ Key components and wiring

- NodeMCU, LED, DHT22, CDS, Relay/USBLED, 0.96" OLED display, Raspberry pi
- NodeMCU의 기본 핀 할당은 HW#1의 자료 참조

❑ Recommended Libraries or packages to be used

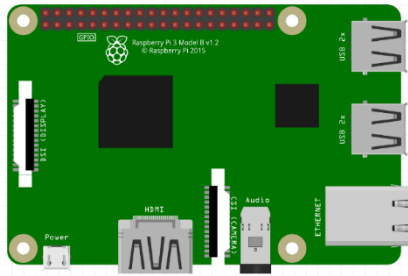
- NodeMCU: DHTesp for DHT22, adafruit_SSD1306 and adafruit_gfx for OLED display, **EspMQTTClient for MQTT**
- NodeMCU의 경우 **Non-blocking code**로 작성해야 함 (불가피한 경우 제외하고 delay() 함수 사용 금지)
- Rpi: mosquito MQTT broker and client, flask-mqtt, render_template

❑ 추가 기능

- 개인적으로 가능한 부분 추가하고, 관련하여 보고서에 내용 기술
 - 예: 서버 홈 페이지를 10초마다 갱신하여 html 페이지의 새로운 데이터 갱신하기 등

최종 구현 시나리오 및 목표

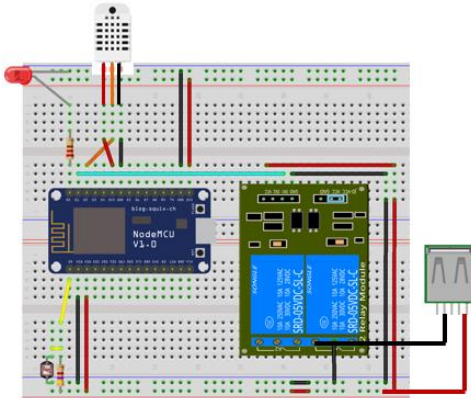
<Flask 서버, MQTT Broker>



센서 값 확인

Publish/Client 명령
Publish/Client 센서 값 전달

nodeMCU1



<Led, Relay/ USBLED, DHT22, CDS, 0.96" OLED display>

Specifications

❑ NodeMCU에서의 기본 기능

- LED/USBLED 수동 제어 (on/off/toggle) 가능
- USBLED 제어
 - **USBLED 수동제어(ON)는 조도값에 따른 제어보다 우선함**
 - LED/USBLED가 OFF 상태이면 우선제어 상태가 해제된 상태임
 - **OFF 버튼이 눌러지면 강제로 OFF됨. 이후 이벤트가 발생하면 이벤트에 따라 동작**
 - 어두워지는 event(밝은 상태→어두운 상태)가 발생하면 10초 동안 USBLED 불 켜지기. 계속 어두운 상태이거나, USBLED turn on 상태에서 밝아진 상태에서 어두워지면 USBLED 상태에 영향을 주지 않음
 - 이벤트 동작상태에서는 toggle 버튼은 동작하지 않음
- 온도/습도/조도 데이터
 - 온도/습도/조도 데이터는 10초마다 측정하고 publish
 - nodeMCU에서 측정되는 Temperature, Humidity, 조도센서 등의 값은 browser에서 버튼 클릭 후 해당 값 표시 (기본 기능임. 추가로 기능을 향상시키는 것 가능)
- Non-blocking code로 작성해야 함 (불가피한 경우 제외하고 delay() 함수 사용 금지)

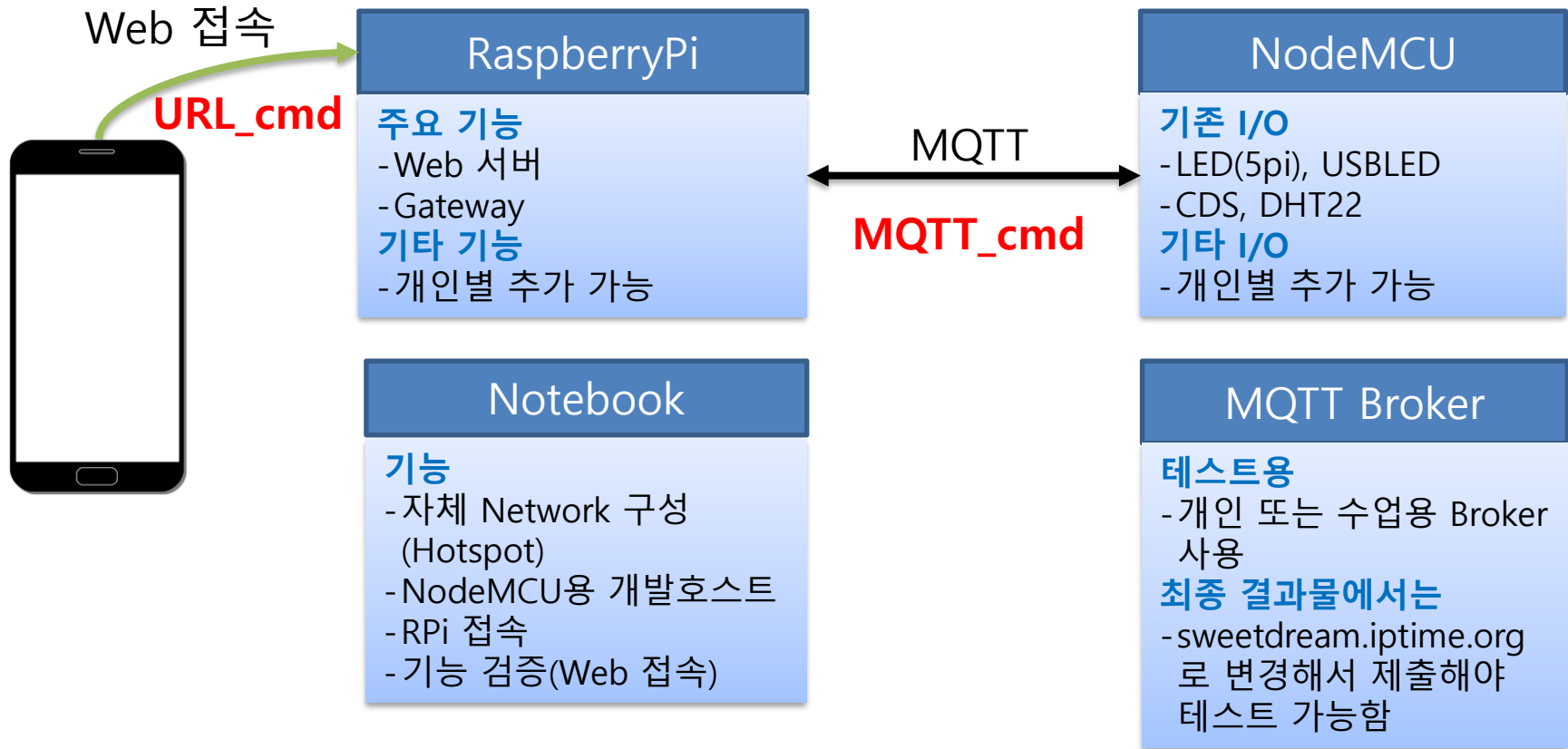
❑ Web server에서의 기본 기능

- LED/USBLED 제어를 위한 버튼 (on/off/toggle)
- nodeMCU에서 측정되는 Temperature, Humidity, 조도센서 등의 값은 browser에서 버튼 클릭 후 해당 값 표시 (기본 기능임. 추가로 기능을 향상시키는 것 가능)

❑ 추가 기능

- 개인적으로 가능한 부분 추가하고, 관련하여 보고서에 내용 기술
 - 예: 서버 홈 페이지를 10초마다 갱신하여 html 페이지의 새로운 데이터 갱신하기 등

최종 실습과제 시스템 구성도



(Note) URL_cmd와 MQTT_cmd는 다른 슬라이드에 기술되어 있음

List of URL Command (URL_cmd)

❑ Light Control

`http://{자신의 RPi IP주소}/iot/ 학번/{command}`

- `{led, ledon, ledoff}` – LED {toggle, on, off}
- `{usbled, usbledon, usbledoff}` – USBled {toggle, on, off}

❑ Sensor Node

`http://{자신의 RPi IP주소}/iot/ 학번/{command}`

- `cds` - Get light intensity
- `dht22` - Get temperature and humidity data from DHT22 in **JSON**
 - (참고) [Decoding and Encoding JSON Arduino | Random Nerd Tutorials](#)
- ~~`dht22_t`~~, ~~`dht22_h`~~

❑ Error page 처리

- 정상적인 IP 주소가 입력되었을 경우 (자신의 RPi3 IP 주소 및 포트번호) 모두 정상적인 웹페이지로 동작하도록 유도
- Page not found 형태의 결과 출력되지 않도록 하기

List of MQTT Topics (MQTT_cmd)

- ❑ MQTT topics for NodeMCU
 - Topics to be subscribed for LED and USBLED devices
 - Topic: **iot/{학번}**
 - Message: {led, ledon, ledoff, usbled, usbledon, usbledoff}
 - Topics to publish DHT22/CDS sensor data
 - Topics for publishing
 - **iot/{학번}/{dht22, cds}**
 - Message: JSON form
- ❑ MQTT topics for other devices (rpi, browser and etc)
 - Topics to control LED and USBLED devices
 - Topic: **iot/{학번}**
 - Message: {led, ledon, ledoff, usbled, usbledon, usbledoff}
 - Topics to get sensor data
 - Topics: **iot/{학번}/{dht22, cds}**
 - Message: JSON form

Wireframe

http://192.168.137.110:5000

Refresh

HEADER

온도	습도	조도
LED	LED ON	LED OFF
USBLED	USBLED ON	USBLED OFF

설계과제 #1 제출 내용

□ 내용

- 보고서: 관련된 기본내용을 서론,본론,결론 형태의 논리적 흐름을 가지고 구성. 핵심적 기능이 잘 파악될 수 있도록 state diagram or ASM chart 등을 활용할 것
- 소스코드: 소스코드 문서화도 되어 있어야 함
- 실행방법에 대한 안내가 있어야 함: readme.txt 또는 howtorun.txt 등의 파일에 제시
- 제출물: 보고서, 소스코드 및 실행안내 문서

□ 제출

- LMS를 통해 제출
- 기한: 11월 4일(토) 오후 10시