데이터 입력/출력

양방향 통신(IOT)

MQTT

라즈베리파이

제어 컨트롤러

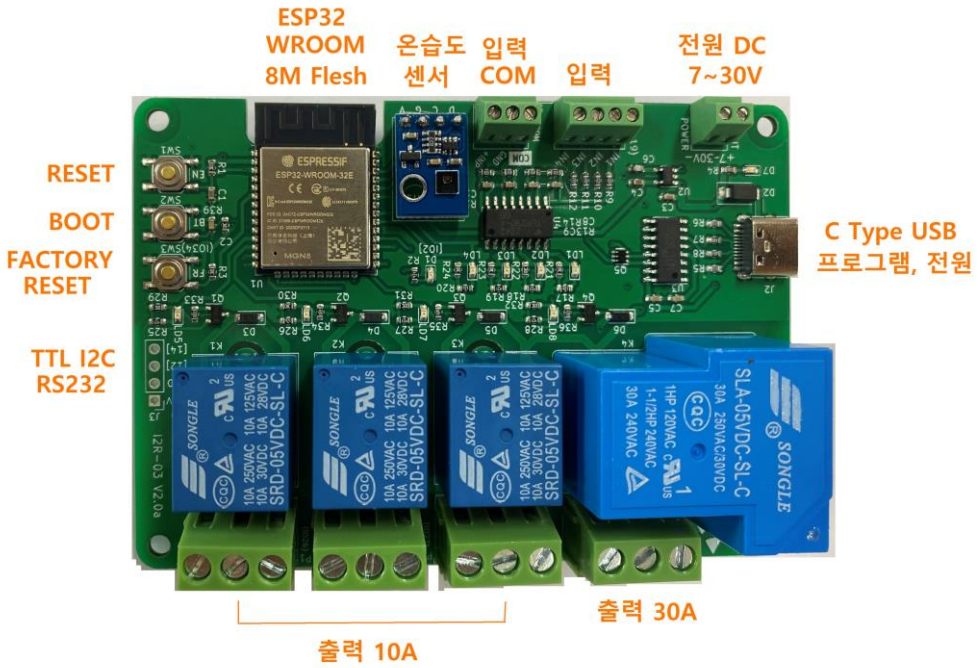
Node-RED

데이터저장

데이터저장

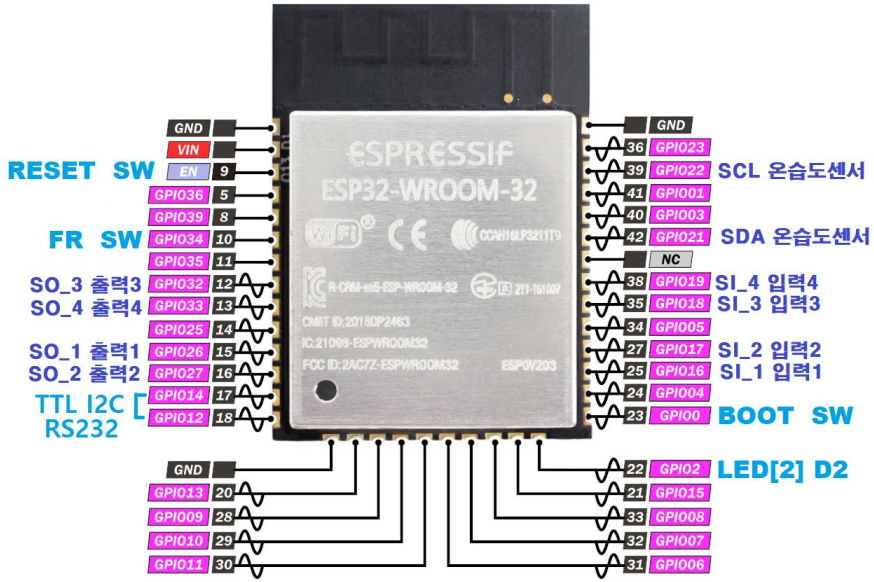
MySQL

IoT Board



온습도 센서

ESP32



ESP32는 Wi-Fi와 Bluetooth 기능을 통합한 마이크로컨트롤러(MCU)이다.

전체 코드 (default)

https://github.com/kdi6033/i2r-03/blob/main/0%20Source-Program-IoT/board-i2r-03/board-i2r-03.ino

아두이노 IDE 설치

공식 웹사이트 : https://www.arduino.cc/en/software

Arduino IDE 2.3.3

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the **Arduino IDE 2.0 documentation**.

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on **GitHub**.

Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

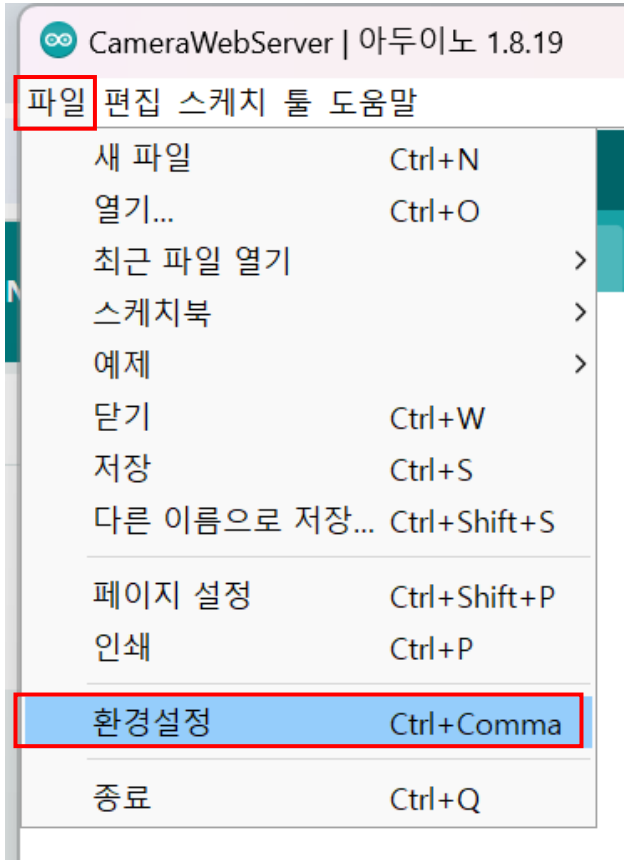
Refer to the **Arduino IDE 1.x documentation** for installation instructions.

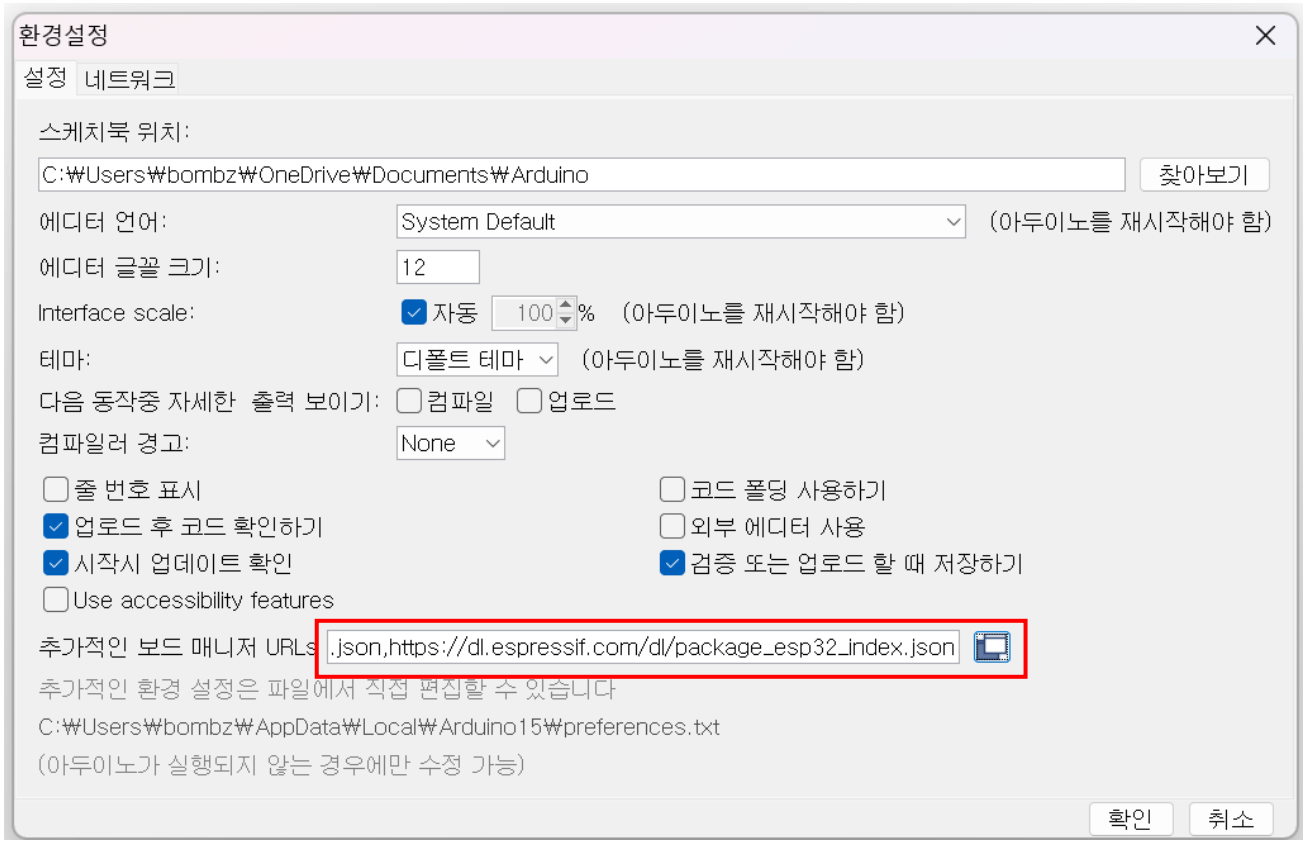
SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

- 아두이노 IDE를 열고 파일 > 환경 설정 메뉴로 이동합니다.
- 추가 보드 매니저 URL에 다음 주소를 추가합니다.

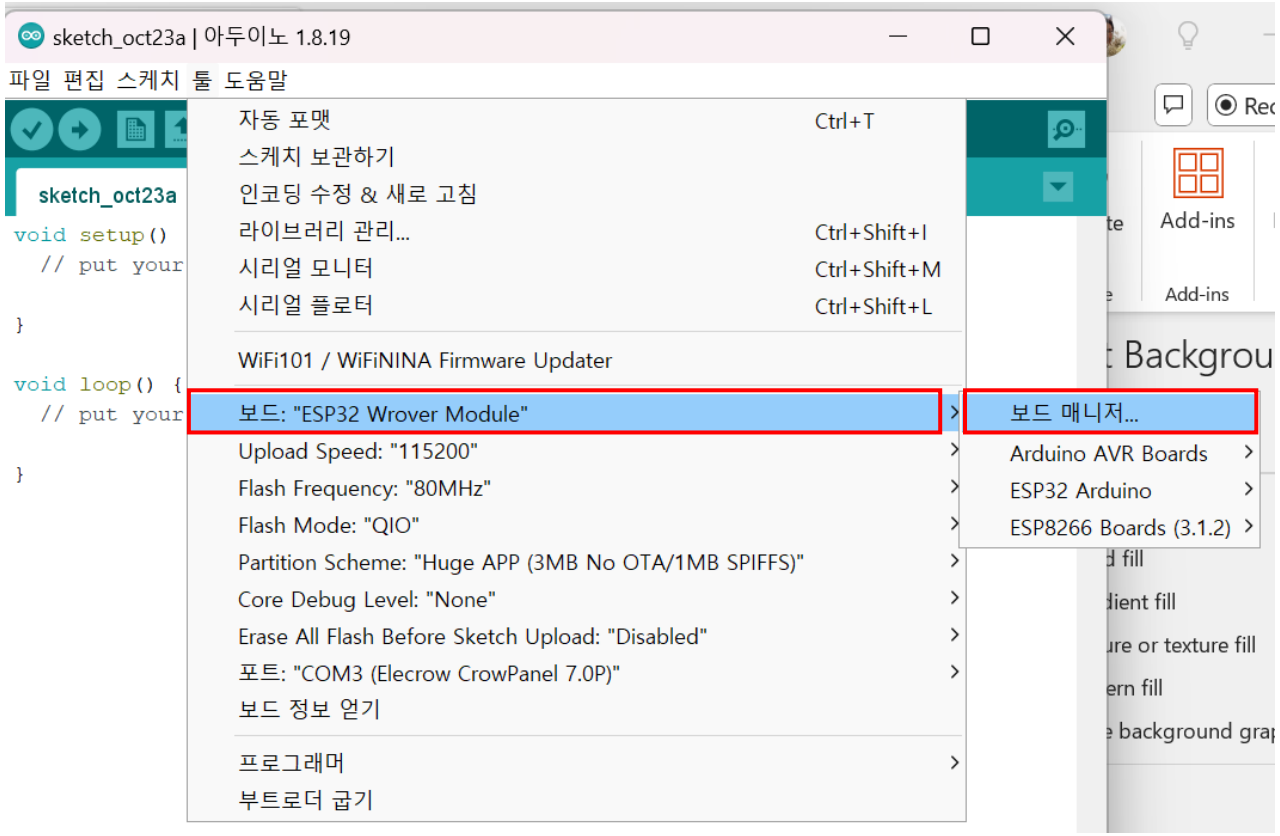
  https://dl.espressif.com/dl/package_esp32_index.json

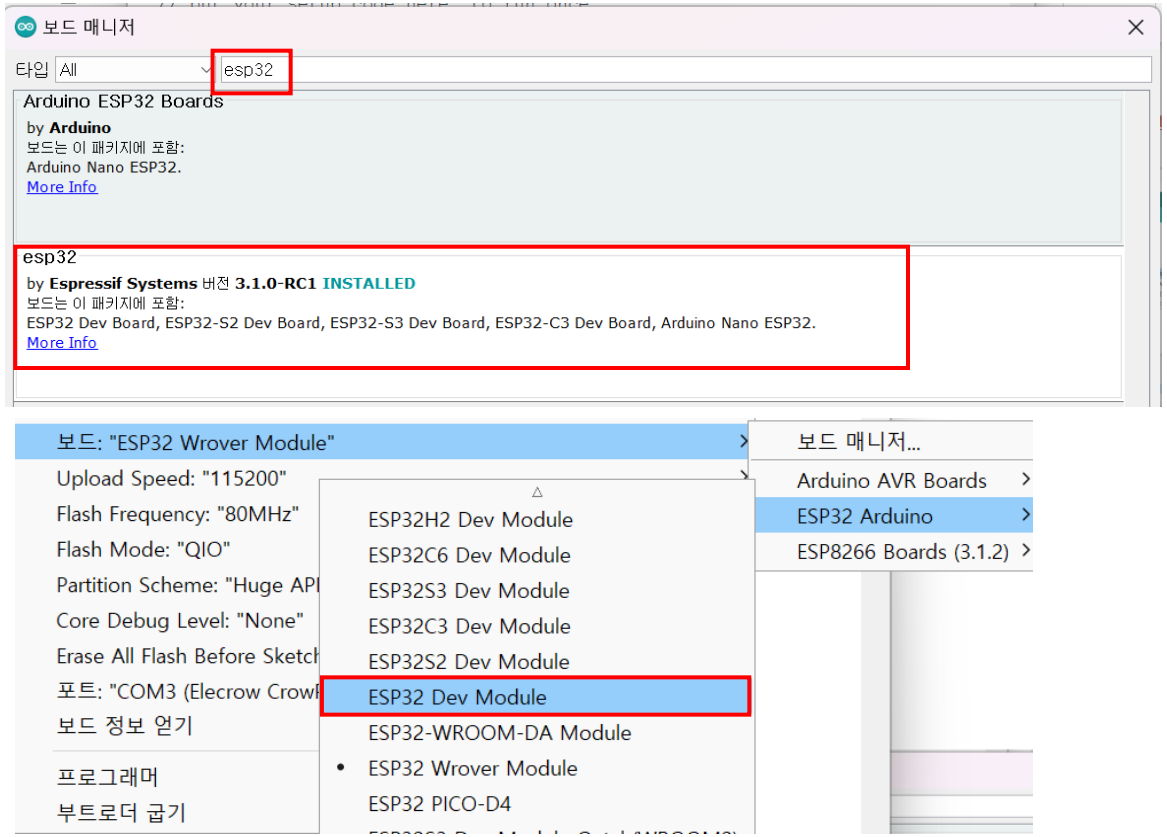




https://dl.espressif.com/dl/package_esp32_index.json

- 설정을 저장한 후, 도구 > 보드 > 보드 매니저에서 "ESP32"를 검색하고 설치합니다.
- 설치 후, 도구 > 보드에서 ESP32 개발 보드를 선택합니다 (예: ESP32 Dev Module).





IoT 보드 상태 정보 출력 (콘솔)

https://github.com/BumjuAhn/2024_IOT_lecture/blob/main/lecture_08_week/01_in_out/01_in_out.ino

```ino
#include <Wire.h>
#include <Adafruit_AHTX0.h>

Adafruit_AHTX0 aht;

// Output pin numbers
const int outputPins[4] = {26, 27, 32, 33};
// Input pin numbers
const int inputPins[4] = {16, 17, 18, 19};

void setup() {
  // Initialize Serial for debugging
  Serial.begin(115200);

  // Check if AHT sensor is connected
  if (! aht.begin()) {
    Serial.println("Could not find AHT sensor!");
    while (1) delay(10);
  }

  // Set each output pin as an output
  for (int i = 0; i < 4; i++) {
    pinMode(outputPins[i], OUTPUT);
  }

  // Set each input pin as an input
  for (int i = 0; i < 4; i++) {
    pinMode(inputPins[i], INPUT);
  }
}

void loop() {
  // Cycle through each output pin
  for (int i = 0; i < 4; i++) {
    // Turn the current output pin on
    digitalWrite(outputPins[i], HIGH);

    // Wait for a second
    delay(1000);

    // Turn the current output pin off
    digitalWrite(outputPins[i], LOW);
  }

  // Cycle through each input pin
  for (int i = 0; i < 4; i++) {
    // Read the state of the current input pin
    int pinState = digitalRead(inputPins[i]);

    // Print the state of the current input pin
    Serial.print("Pin ");
    Serial.print(inputPins[i]);
    Serial.print(": ");
    Serial.println(pinState);
  }

  // Read temperature and humidity from AHT sensor
  sensors_event_t humidity, temp;
  aht.getEvent(&humidity, &temp);

  // Print temperature and humidity
  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degrees C");

  Serial.print("Humidity: ");
  Serial.print(humidity.relative_humidity);
  Serial.println("% rH");

  delay(2000); // Wait for 2 seconds before reading again
}
```

라이브러리 포함 및 객체 생성

핀번호 설정

통신속도 설정

온습도센서체크

출력핀 선언

입력핀 선언

출력핀 ON/OFF

입력핀 데이터 출력

온습도 데이터 출력

프로그램 업로드

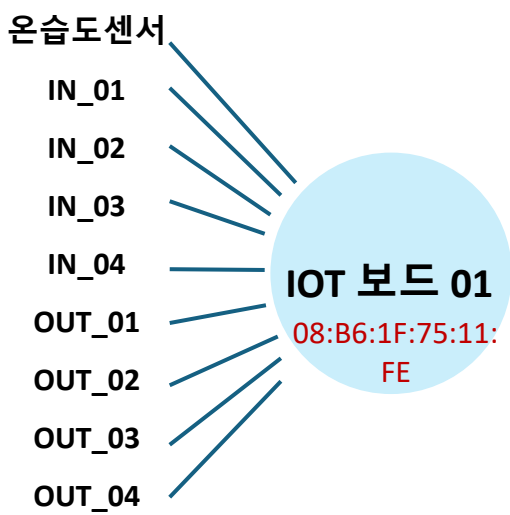
sketch_oct23a | 아두이노 1.8.19

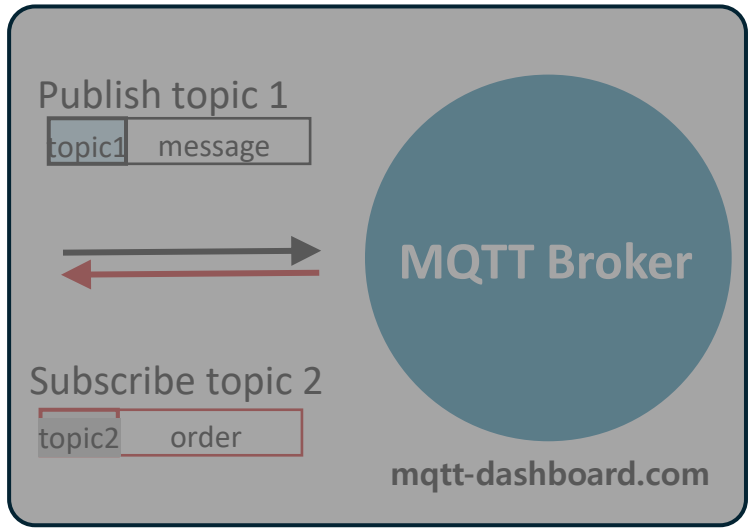
파일 편집 스케치 툴 도움말

sketch_oct23a §

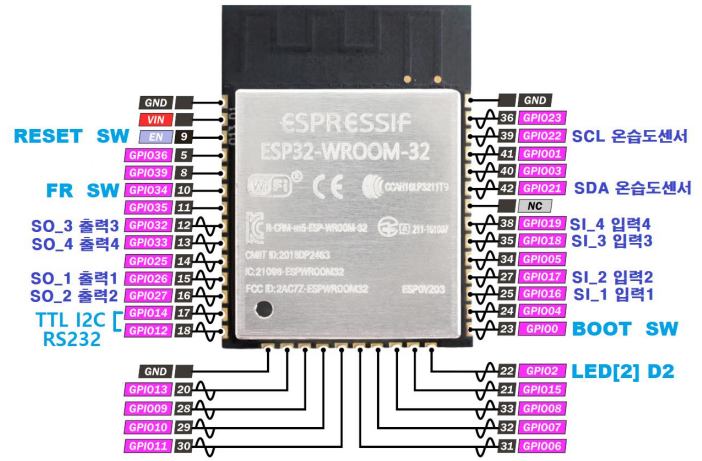
```
#include <Wire.h>
#include <Adafruit_AHTX0.h>

Adafruit_AHTX0 aht;

// Output pin numbers
const int outputPins[4] = {26, 27, 32, 33};
// Input pin numbers
const int inputPins[4] = {16, 17, 18, 19};

void setup() {
  // Initialize Serial for debugging
  Serial.begin(115200);

  // Check if AHT sensor is connected
  if (! aht.begin()) {
    Serial.println("Could not find AHT sensor!");
    while (1) delay(10);
  }

  // Set each output pin as an output
  for (int i = 0; i < 4; i++) {
    pinMode(outputPins[i], OUTPUT);
  }

  // Set each input pin as an input
  for (int i = 0; i < 4; i++) {
    pinMode(inputPins[i], INPUT);
  }
}

void loop() {
  // Cycle through each output pin
  for (int i = 0; i < 4; i++) {
    // Turn the current output pin on
    digitalWrite(outputPins[i], HIGH);

    // Wait for a second
```

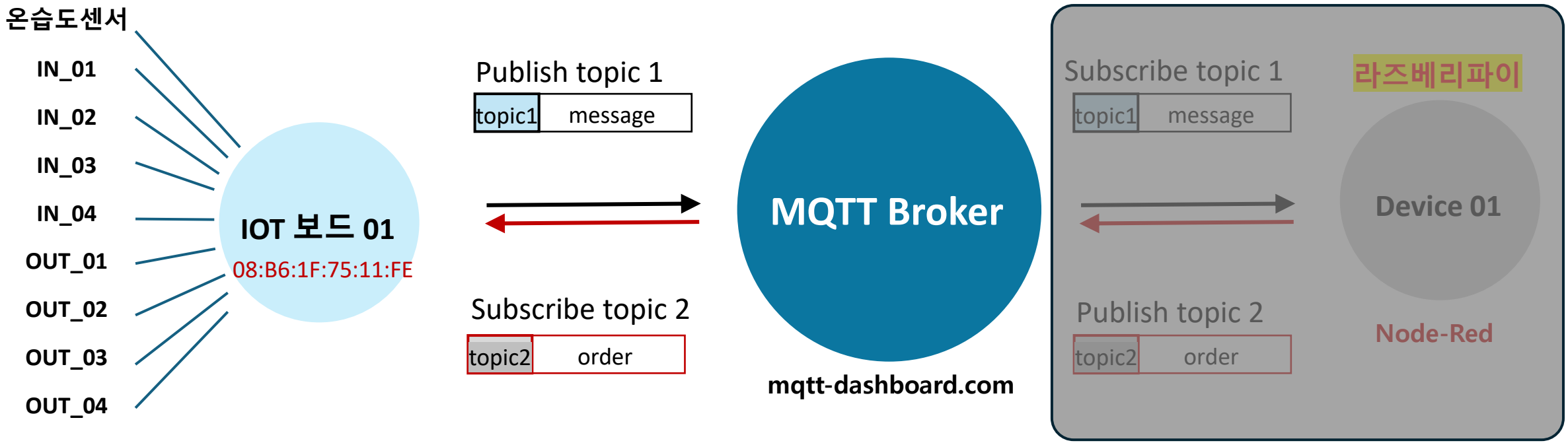
74 ESP32 Wrover Module, Huge APP (3MB No OTA/1MB SPIFFS), QIO, 80MHz, 115200, None, Disabled on COM3






```
Temperature: 30.64 degrees C
Humidity: 37.62% rH
Pin 16: 0
Pin 17: 0
Pin 18: 0
Pin 19: 0
Temperature: 30.57 degrees C
Humidity: 37.71% rH
Pin 16: 0
Pin 17: 0
Pin 18: 0
Pin 19: 0
Temperature: 30.50 degrees C
Humidity: 37.99% rH
```





void connectToWiFi()

// WiFI 연결

void connectToMQTT()

// Mqtt 연결

void publishMsg()

// MQTT 메시지 송신 함수

void callback(char* topic, byte* payload, unsigned int length)

// MQTT 메시지 수신 시 호출되는 콜백 함수

IoT 보드 MQTT 통신

```
#include <WiFi.h>
#include <PubSubClient.h>

// WiFi 및 MQTT 설정
const char* ssid = "your_wifi_ssid";              // WiFi SSID
const char* password = "your_wifi_password";      // WiFi 비밀번호
const char* mqttBroker = "your_mqtt_broker_ip";   // MQTT 브로커 주소
const char* mqttPublishTopic = "test/topic";      // 발행할 MQTT 주제
const char* mqttSubscriptionTopic = "test/subscription"; // 구독할 MQTT 주제

// MQTT 클라이언트 객체
WiFiClient espClient;
PubSubClient client(espClient);

// 메시지 발행 주기
unsigned long lastMsgTime = 0;
const unsigned long publishInterval = 5000;  // 5초

// WiFi 연결
void connectToWiFi() {
  Serial.print("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected to WiFi");
}

// MQTT 연결
void connectToMQTT() {
  client.setServer(mqttBroker, 1883);
  while (!client.connected()) {
    Serial.print("Connecting to MQTT...");
    if (client.connect("ESP32_Client")) { // 클라이언트 이름 지정
      Serial.println("Connected to MQTT");

      // 연결 후 주제를 구독
      client.subscribe(mqttSubscriptionTopic);
      Serial.println("Subscribed to topic: " + String(mqttSubscriptionTopic));
    } else {
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      Serial.println(" trying again in 5 seconds");
      delay(5000);
    }
  }
}
```

WiFi 연결

MQTT 연결

```
// MQTT 메시지 수신 시 호출되는 콜백 함수
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message received on topic: ");
  Serial.println(topic);

  Serial.print("Message: ");
  for (unsigned int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

// 메시지를 발행하는 함수
void publishMsg() {
  String message = "Hello from ESP32!";
  client.publish(mqttPublishTopic, message.c_str());  // 메시지 발행
  Serial.println("Published: " + message);
}

void setup() {
  Serial.begin(115200);
  connectToWiFi();
  client.setCallback(callback);  // 콜백 함수 설정
  connectToMQTT();
}

void loop() {
  // MQTT 연결 확인 및 재연결
  if (!client.connected()) {
    connectToMQTT();
  }
  client.loop();

  // 주기적으로 메시지 발행
  if (millis() - lastMsgTime >= publishInterval) {
    lastMsgTime = millis();
    publishMsg();
  }
}
```

명령어 수신

데이터 송신

setup

loop

https://github.com/BumjuAhn/2024_IOT_lecture/blob/main/lecture_08_week/02_mqtt_setup/02_mqtt_setup.ino

해결과제

아래 규칙으로 MQTT 통신하는 IoT 보드를 디자인하고 Node-red를 이용하여 제어하려라

IoT 보드 (주찬)

- Broker : test.mosquitto.org
- Mac address : D4:8A:FC:B5:34:64
- Topic out : i2r/a@gmail.com/out
- Topic in : i2r/a@gmail.com/in

IOT 보드 (솔파)

- Broker : ai.doowon.ac.kr
- Mac address : 08:B6:1F:75:11:FE
- Topic out : i2r/b@gmail.com/out
- Topic in : i2r/b@gmail.com/in

	Topic	Message
Data		
topic1 message →	iot/b@gmail.com/out	{"type":3, “mac”:” D4:8A:FC:B5:34:64”, "email":“a@gmail.com","temp":28.4,"humi":38,"in":[0,0,0,0],"out":[0,0,0,0]}
Order		
topic2 message →	iot/b@gmail.com/in	{"order":2, ”mac”:” D4:8A:FC:B5:34:64”, "no":1,"value":true}