**Iot\_edge\_light ap,wifi를 활용한 mqtt를 위한 변경사항**

소스코드 구조설명서

**2024.** **09**

**㈜쿠노소프트**

- 목차 –

1. [wificonnecttest.ino 4](#_bookmark0)
   1. [파일 설명 4](#_bookmark1)
2. [testv1.ino 16](#_bookmark31)
   1. [파일 설명 16](#_bookmark32)
   2. [setup,connect\_wifi 16](#_bookmark33)
3. [testv2.ino 19](#_bookmark35)
   1. [파일 설명 19](#_bookmark36)
   2. setup [19](#_bookmark37)
   3. loop
4. [testv5.ino 21](#_bookmark42)
   1. [파일 설명 21](#_bookmark43)
   2. Setup
   3. loop
5. [iot\_edge\_light.cpp 22](#_bookmark44)
   1. [파일 설명 22](#_bookmark45)
   2. 라이브러리 추가 및 변수선언
   3. Setup
   4. loop

# wificonnecttest.ino

## 파일 설명

wifi연결 예제를 활용한 connection test

#include <WiFi.h>

#include <ESPAsyncWebServer.h>

const char\* wifi\_network\_ssid = "Kouno\_GIGA\_2.4G";

const char\* wifi\_network\_password =  "1092724855";

const char \*soft\_ap\_ssid = "Kouno\_Allsense";

const char \*soft\_ap\_password = "1092724855";

AsyncWebServer server(80);

void setup() {

  Serial.begin(115200);

  WiFi.mode(WIFI\_MODE\_APSTA);

  WiFi.softAP(soft\_ap\_ssid, soft\_ap\_password);

  WiFi.begin(wifi\_network\_ssid, wifi\_network\_password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.println("Connecting to WiFi..");

  }

  Serial.print("ESP32 IP as soft AP: ");

  Serial.println(WiFi.softAPIP());

  Serial.print("ESP32 IP on the WiFi network: ");

  Serial.println(WiFi.localIP());

  server.on("/hello", HTTP\_GET, [](AsyncWebServerRequest \* request) {

    if (ON\_STA\_FILTER(request)) {

      request->send(200, "text/plain", "Hello from STA");

      return;

    } else if (ON\_AP\_FILTER(request)) {

      request->send(200, "text/plain", "Hello from AP");

      return;

    }

    request->send(200, "text/plain", "Hello from undefined");

  });

  server.begin();

}

void loop() {}

# Testv1.ino

## 파일 설명

Ap를 열고 해당 ap의 접속화면 구현을 위해 html코드를 추가

* 1. Setup,connect\_wifi  
     : ap를 활성화, wifi연결을 위한 connect\_wifi함수

void Connect\_WiFi(){

  WiFi.begin(ssid, password);

  while(WiFi.status() != WL\_CONNECTED){

    Serial.print(".");

    delay(500);

  }

}

void setup(){

  Serial.begin(115200);

  delay(1000);

  Serial.print("Setting soft access point mode");

  WiFi.softAP(ssid, password);

  IPAddress IP = WiFi.softAPIP();

  Serial.print("AP IP address: ");

  Serial.println(IP);

  server.begin();

}

* 1. Loop  
     :서버를열고wifi연결

void loop(){

  WiFiClient client=server.available();

  if(client){

    String request = client.readStringUntil('\r');

    Serial.println(request); // 요청을 출력하여 디버깅

    if (request.indexOf("wifissid=") != -1 && request.indexOf("wifipassword=") != -1) {

      int ssidStartIndex = request.indexOf("wifissid=") + 9; // '=' 이후에서 시작

      int ssidEndIndex = request.indexOf('&', ssidStartIndex);

      if (ssidEndIndex == -1) ssidEndIndex = request.length();

      wifissid = request.substring(ssidStartIndex, ssidEndIndex);

      int passwordStartIndex = request.indexOf("wifipassword=") + 13; // '=' 이후에서 시작

      int passwordEndIndex = request.indexOf(' ', passwordStartIndex);

      if (passwordEndIndex == -1) passwordEndIndex = request.length();

      wifipassword = request.substring(passwordStartIndex, passwordEndIndex);

      Serial.print("WiFi SSID: ");

      Serial.println(wifissid); // SSID 출력

      Serial.print("WiFi Password: ");

      Serial.println(wifipassword); // 비밀번호 출력

      // WiFi 재연결

      // Connect\_WiFi();

    }

    String html ="<!DOCTYPE html> \

    <html> \

    <head> \

    <style> \

    .scrollbox { \

      width: 300px; \

      height: 150px; \

      overflow: auto; \

      border: 1px solid #ccc; \

      padding: 10px; \

    } \

    </style> \

    </head> \

    <body> \

    <center><h1>Konuno Allsense Soft access point</h1></center> \

    <center><h2>Web Server</h2></center> \

    <form action=\"/\" method=\"GET\"> \

    <input type=\"text\" name=\"wifissid\" placeholder=\"WiFi SSID\" required> \

    <br> \

    <input type=\"text\" name=\"wifipassword\" placeholder=\"WiFi Password\" required> \

    <br> \

    <button type=\"submit\">Change WiFi</button><br><br> \

    </form> \

    <div class=\"scrollbox\"> \

    <p>" + wifissid + "</p> \

    <p>" + wifipassword + "</p> \

    </div> \

    </body> \

    </html>";

    client.print(html);

    request="";

  }

}

# Testv2.ino

## 파일 설명

Ap 에 접속 여러 페이지를 띄우기 위해 webserver를 사용하는 버전 scannetworks를 추가하여 wifi리스트를 볼 수 있다.

* 1. Setup,scanNetworks  
     : ap를 활성화, wifi연결을 위한 connect\_wifi함수

void scanNetworks() {

  n = WiFi.scanNetworks();

  for (int i = 0; i < n; ++i) {

    ssidList[i] = WiFi.SSID(i);

    rssiList[i] = String(WiFi.RSSI(i));

    channelList[i] = String(WiFi.channel(i));

    encryptionList[i] = getEncryptionType(WiFi.encryptionType(i));

  }

  // 메모리 해제

  WiFi.scanDelete();

}

void setup(){

  Serial.begin(115200);

  delay(1000);

  Serial.print("Setting soft access point mode");

  WiFi.softAP(ssid, password);

  IPAddress IP = WiFi.softAPIP();

  Serial.print("AP IP address: ");

  Serial.println(IP);

  server.begin();

}

* 1. Loop  
     :서버를열고wifi연결

void loop() {

  WiFiClient client1=server.available();

  if(mqtt\_server.length()!=0){

    // && !client.connected()

    client.setServer(mqtt\_server.c\_str(), 1883);

    if (!client.connected()) {

      reconnect();

    }

    String messageChar = "hellow";

    client.publish(topic.c\_str(), messageChar.c\_str());

    Serial.print("hellow");

  }

  if(client1){

    String request = client1.readStringUntil('\r');

    Serial.println(request); // 요청을 출력하여 디버깅

    // "wifissid=", "wifipassword=", "mqtt\_server="가 포함되어 있는지 확인

    if (request.indexOf("wifissid=") != -1 && request.indexOf("wifipassword=") != -1 && request.indexOf("mqtt\_server=") != -1) {

        int ssidStartIndex = request.indexOf("wifissid=") + 9; // '=' 이후에서 시작

        int ssidEndIndex = request.indexOf('&', ssidStartIndex);

        if (ssidEndIndex == -1) ssidEndIndex = request.length();

        wifissid = request.substring(ssidStartIndex, ssidEndIndex);

        int passwordStartIndex = request.indexOf("wifipassword=") + 13; // '=' 이후에서 시작

        int passwordEndIndex = request.indexOf('&', passwordStartIndex);

        if (passwordEndIndex == -1) passwordEndIndex = request.length();

        wifipassword = request.substring(passwordStartIndex, passwordEndIndex);

        int topicStartIndex = request.indexOf("topic=") + 6; // '=' 이후에서 시작

        int topicEndIndex = request.indexOf('&', topicStartIndex);

        if (topicEndIndex == -1) topicEndIndex = request.length();

        topic = request.substring(topicStartIndex, topicEndIndex);

        int mqttStartIndex = request.indexOf("mqtt\_server=") + 12; // '=' 이후에서 시작

        int mqttEndIndex = request.indexOf(' ', mqttStartIndex);

        if (mqttEndIndex == -1) mqttEndIndex = request.length();

        mqtt\_server = request.substring(mqttStartIndex, mqttEndIndex); // mqtt\_server 변수 추가

        Serial.print("WiFi SSID: ");

        Serial.println(wifissid); // SSID 출력

        Serial.print("WiFi Password: ");

        Serial.println(wifipassword); // 비밀번호 출력

        Serial.print("Topic: ");

        Serial.println(topic); // 비밀번호 출력

        Serial.print("MQTT Server: ");

        Serial.println(mqtt\_server); // MQTT 서버 출력

    }

    html = "<!DOCTYPE html>\

    <html>\

    <head>\

    <style>\

    .scrollbox {\

      width: 750px;\

      height: 550px;\

      overflow: auto;\

      border: 1px solid #ccc;\

      padding: 10px;\

      margin: 0 auto; /\* 중앙 정렬을 위한 자동 마진 \*/\

    }\

    body {\

      text-align: center; /\* 모든 텍스트를 가운데 정렬 \*/\

    }\

    form {\

      display: inline-block; /\* 폼을 인라인 블록으로 설정하여 가운데 정렬 \*/\

    }\

    button {\

      margin: 10px; /\* 버튼 간격 조정 \*/\

    }\

    </style>\

    </head>\

    <body>\

    <center><h1>WiFi Networks</h1></center>\

    <center>\

      <button onclick=\"location.href='current\_page.html'\">current page</button>\

      <button onclick=\"location.href='graph.html'\">grape page</button>\

    </center>\

    <div class=\"scrollbox\">";

    // WiFi 네트워크 스캔

    if (n == 0) {

      html += "<p>No networks found</p>";

    } else {

      html += "<table>\

      <tr><th>Nr</th><th>SSID</th><th>RSSI</th><th>Channel</th><th>Encryption</th></tr>";

      for (int i = 0; i < n; ++i) {

        html += "<tr>";

        html += "<td>" + String(i + 1) + "</td>";

        html += "<td>" + ssidList[i] + "</td>";

        html += "<td>" + rssiList[i] + "</td>";

        html += "<td>" + channelList[i] + "</td>";

        html += "<td>" + encryptionList[i] + "</td>";

        html += "</tr>";

      }

      html += "</table>";

    }

    html += "</div>\

    <br>\

    <center><h1>Kouno Allsense Soft access point</h1></center> \

    <center><h2>Web Server</h2></center> \

    <form action=\"/\" method=\"GET\"> \

    <input type=\"text\" name=\"wifissid\" placeholder=\"WiFi SSID\" required> \

    <br> \

    <input type=\"text\" name=\"wifipassword\" placeholder=\"WiFi Password\" required> \

    <br> \

    <input type=\"text\" name=\"topic\" placeholder=\"Topic\" required> \

    <br> \

    <input type=\"text\" name=\"mqtt\_server\" placeholder=\"Mqtt Server\" required> \

    <br> \

    <button type=\"submit\">Change WiFi</button><br><br> \

    </form> \

    <div class=\"scrollbox\"> \

    <p>wifissid: " + wifissid + "</p> \

    <p>wifipassword: " + wifipassword + "</p> \

    <p>topic: " + topic + "</p> \

    <p>Mqtt Server: " + mqtt\_server + "</p> \

    ";

    // WiFi 연결 상태 확인

    if (WiFi.status() == WL\_CONNECTED) {

        html += "<p>WiFi connected</p>";

    }

    // MQTT 연결 상태 확인

    if (client.connected()) {

        html += "<p>MQTT connected</p>";

    }

    html += "</div> \

    </body>\

    </html>";

    client1.print(html);

    if ((wifissid.length() != 0) && (wifipassword.length() != 0) && (WiFi.status() != WL\_CONNECTED)) {

      Serial.println("\nSetting Station configuration ... ");

      WiFi.begin(wifissid.c\_str(), wifipassword.c\_str());

      Serial.println(String("Connecting to ")+ wifissid);

      int attemptCount = 0; // 시도 횟수 카운터

      while (WiFi.status() != WL\_CONNECTED && attemptCount < 10){

        delay(500);

        Serial.print(".");

        attemptCount++;

      }

      if (WiFi.status() == WL\_CONNECTED) {

          Serial.println("\nConnected, IP address: ");

          Serial.println(WiFi.localIP());

      } else {

          Serial.println("\nFailed to connect after 10 attempts.");

      }

    }

  }

}

# Testv5.ino

## 파일 설명

Ap 에 접속 여러 페이지를 띄우고 실시간 처리를 위해 asyncwebserver를 사용하는 버전 scannetworks를 추가하여 wifi리스트를 볼 수 있다.

* 1. Setup   
     : ap를 활성화 , server on

void setup() {

  Serial.begin(115200);

  delay(1000);

  Serial.print("Setting soft access point mode");

  WiFi.softAP(ssid, password);

  WiFi.mode(WIFI\_MODE\_APSTA);

  // WiFi.mode(WIFI\_AP\_STA);

  // WiFi.mode(WIFI\_AP);

  //ap모드로 ap를 열고 값을 전달받은후 sta모드로 mqtt연결??

  // WiFi.mode(WIFI\_STA);

  IPAddress IP = WiFi.softAPIP();

  Serial.print("AP IP address: ");

  Serial.println(IP);

  scanNetworks();

  server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

    // 쿼리 파라미터를 가져옵니다.

    if (request->hasParam("wifissid")) {

      wifissid = request->getParam("wifissid")->value();

    }

    if (request->hasParam("wifipassword")) {

      wifipassword = request->getParam("wifipassword")->value();

    }

    if (request->hasParam("topic")) {

      topic = request->getParam("topic")->value();

    }

    if (request->hasParam("mqtt\_server")) {

      mqtt\_server = request->getParam("mqtt\_server")->value();

    }

    String html = "<!DOCTYPE html>\

    <html>\

    <head>\

    <style>\

    .scrollbox {\

      width: 750px;\

      height: 250px;\

      overflow: auto;\

      border: 1px solid #ccc;\

      padding: 10px;\

      margin: 0 auto;\

    }\

    body {\

      text-align: center;\

    }\

    .header {\

      text-align: center;\

    }\

    .button-container {\

      text-align: right;\

      margin-top: 10px;\

    }\

    button {\

      margin: 10px;\

    }\

    </style>\

    </head>\

    <body>\

      <div class=\"header\">\

        <h1>WiFi Networks</h1>\

      </div>\

      <div class=\"button-container\">\

        <button onclick=\"location.href='/'\">Main Page</button>\

        <button onclick=\"location.href='/data'\">Sensor Data</button>\

      </div>\

      <div class=\"scrollbox\">\

    ";

    if (n == 0) {

      html += "<p>No networks found</p>";

    } else {

      html += "<table>\

      <tr><th>Nr</th><th>SSID</th><th>RSSI</th><th>Channel</th><th>Encryption</th></tr>";

      for (int i = 0; i < n; ++i) {

        html += "<tr>";

        html += "<td>" + String(i + 1) + "</td>";

        html += "<td>" + ssidList[i] + "</td>";

        html += "<td>" + rssiList[i] + "</td>";

        html += "<td>" + channelList[i] + "</td>";

        html += "<td>" + encryptionList[i] + "</td>";

        html += "</tr>";

      }

      html += "</table>";

    }

    html += "</div>\

    <br>\

    <center><h1>Kouno Allsense Soft access point</h1></center>\

    <center><h2>Web Server</h2></center>\

    <form id=\"wifiForm\">\

      <input type=\"text\" name=\"wifissid\" placeholder=\"WiFi SSID\" required>\

      <br>\

      <input type=\"text\" name=\"wifipassword\" placeholder=\"WiFi Password\" required>\

      <br>\

      <input type=\"text\" name=\"topic\" placeholder=\"Topic\" required>\

      <br>\

      <input type=\"text\" name=\"mqtt\_server\" placeholder=\"Mqtt Server\" required>\

      <br>\

      <button type=\"submit\">Change WiFi</button>\

      <br><br>\

    </form>\

    <div id=\"responseBox\" style=\"border:1px solid #ccc; height:200px; overflow-y:scroll;\">\

      <p id='wifiSsid'>Wi-Fi SSID: " + wifissid + "</p>\

      <p id='wifiPassword'>Wi-Fi password: " + wifipassword + "</p>\

      <p id='topic'>topic: " + topic + "</p>\

      <p id='mqttServer'>MQTT server: " + mqtt\_server + "</p>\

      <p id='connectionStatus'>" + getConnectionStatus() + "</p>\

    </div>\

    <script>\

    function updateStatus() {\

      fetch('/config')\

        .then(response => response.json())\

        .then(data => {\

          document.getElementById('connectionStatus').innerHTML = data.connectionStatus;\

        });\

    }\

    setInterval(updateStatus, 5000); // 5초마다 상태 업데이트\

    </script>\

    ";

    html += "</body>\

    </html>";

    request->send(200, "text/html", html);

  });

  server.on("/data", HTTP\_GET, [](AsyncWebServerRequest \*request){

    String dataHtml = "<!DOCTYPE html>\

    <html>\

    <head>\

    <style>\

    .scrollbox {\

      width: 750px;\

      height: 250px;\

      overflow: auto;\

      border: 1px solid #ccc;\

      padding: 10px;\

      margin: 0 auto;\

    }\

    body {\

      text-align: center;\

    }\

    .header {\

      text-align: center;\

    }\

    .button-container {\

      text-align: right;\

      margin-top: 10px;\

    }\

    button {\

      margin: 10px;\

    }\

    </style>\

    </head>\

    <body>\

      <div class=\"header\">\

        <h1>WiFi Networks</h1>\

      </div>\

      <div class=\"button-container\">\

        <button onclick=\"location.href='/'\">Main Page</button>\

        <button onclick=\"location.href='/data'\">Sensor Data</button>\

      </div>\

    ";

    dataHtml += "<center><h1>Allsense Data</h1></center>\

    <center><h2>Web Server</h2></center>\

    <div id=\"responseBox\" style=\"border:1px solid #ccc; height:200px; overflow-y:scroll;\">\

      <p id='wifiSsid'>Wi-Fi SSID: " + wifissid + "</p>\

      <p id='wifiPassword'>Wi-Fi password: " + wifipassword + "</p>\

      <p id='topic'>topic: " + topic + "</p>\

      <p id='mqttServer'>MQTT server: " + mqtt\_server + "</p>\

      <p id='connectionStatus'>" + getConnectionStatus() + "</p>\

    </div>\

    <script>\

    function updateStatus() {\

      fetch('/config')\

        .then(response => response.json())\

        .then(data => {\

          document.getElementById('connectionStatus').innerHTML = data.connectionStatus;\

        });\

    }\

    setInterval(updateStatus, 5000); // 5초마다 상태 업데이트\

    </script>\

    ";

    dataHtml += "</body>\

    </html>";

    request->send(200, "text/html", dataHtml);

  });

  server.on("/config", HTTP\_GET, [](AsyncWebServerRequest \*request){

    String jsonResponse = "{";

    jsonResponse += "\"connectionStatus\": \"" + getConnectionStatus() + "\"";

    jsonResponse += "}";

    request->send(200, "application/json", jsonResponse);

  });

  server.begin();

}

* 1. Loop  
     : 실시간 처리

void loop() {

  if ((wifissid.length() != 0) && (wifipassword.length() != 0) && (WiFi.status() != WL\_CONNECTED)) {

    Serial.println("\nSetting Station configuration ... ");

    WiFi.begin(wifissid.c\_str(), wifipassword.c\_str());

    Serial.println(String("Connecting to ")+ wifissid);

    int attemptCount = 0; // 시도 횟수 카운터

    while (WiFi.status() != WL\_CONNECTED && attemptCount < 10){

      delay(500);

      Serial.print(".");

      attemptCount++;

    }

    if (WiFi.status() == WL\_CONNECTED) {

        Serial.println("\nConnected, IP address: ");

        Serial.println(WiFi.localIP());

    } else {

        Serial.println("\nFailed to connect after 10 attempts.");

    }

  }

  if(mqtt\_server.length()!=0 && !client.connected()){

    // && !client.connected()

    client.setServer(mqtt\_server.c\_str(), 1883);

    Serial.println(mqtt\_server.c\_str());

    if (!client.connected()) {

      reconnect();

    }

    String messageChar = "hellow";

    client.publish(topic.c\_str(), messageChar.c\_str());

    Serial.print("hellow");

  }

}

# Iot\_edge\_light.cpp

## 파일 설명

기존 코드에 testv5까지 진행하며 만든 코드를 추가로 붙여넣었다. 기존코드 에선 변경한 것이 없다. 기존코드로는 D:\internSeungju\jsproject1\kouno\_light\_r1 의 firmware code.zip을 사용

{

  "name": "electronactivate",

  "version": "1.0.0",

  "main": "main.js",

  "scripts": {

    "start": "electron .",

    "build" : "electron-builder --win --x64",

    "deploy": "electron-builder --windows nsis:x64"

  },

  "build": {

    "productName": "test",

    "appId": "com.electron.hello",

    "asar": true,

    "protocols": {

      "name": "test",

      "schemes": [

        "test"

      ]

    },

    "win": {

      "target": [

        "zip",

        "nsis"

      ],

      "icon": "./resources/installer/Icon.ico"

    },

    "nsis": {

      "oneClick": false,

      "allowToChangeInstallationDirectory": true

    },

    "directories": {

      "buildResources": "./resources/installer/",

      "output": "./exeResult/",

      "app": "."

    },

    "extraResources": [

      "node\_modules/\*\*\*/\*\*/\*"

    ]

  },

  "author": "",

  "license": "ISC",

  "description": "",

  "devDependencies": {

    "electron": "^31.3.0",

    "electron-builder": "^24.13.3",

    "express": "^4.19.2",

    "mqtt": "^5.9.0",

    "mysql": "^2.18.1",

    "pkg": "^5.8.1",

    "serialport": "^12.0.0",

    "sqlite3": "^5.1.7",

    "webpack": "^5.93.0",

    "webpack-cli": "^5.1.4",

    "mqtt-packet": "^9.0.0"

  }

}

## 라이브러리 추가 및 변수 선언 38~131 줄 사용할 변수들을 선언하고 wifi 재접속을 위한 reconnect 함수 및 연결가능한 wifi리스트를 저장하기 위한 함수들이다.

*// mqtt*

*#include* <WiFi.h>

*#include* <WiFiClient.h>

*#include* <WebServer.h>

*#include* <PubSubClient.h>

*#include* <ArduinoJson.h>

*#include* <AsyncTCP.h>

*#include* <ESPAsyncWebServer.h>

*// WiFi 설정*

const char\* ssid = "Kouno\_Allsense\_AP";

const char\* password = "1092724855";

WiFiClient espClient;

PubSubClient client(espClient);

*// 웹서버 초기화*

AsyncWebServer server(80);

int n = 0; *// WiFi 네트워크 수*

String ssidList[100]; *// 최대 20개의 SSID 저장*

String rssiList[100]; *// RSSI 값 저장*

String channelList[100]; *// 채널 저장*

String encryptionList[100]; *// 암호화 타입 저장*

String wifissid; *// WiFi SSID를 저장할 변수*

String wifipassword; *// WiFi 비밀번호를 저장할 변수*

String topic;

String mqtt\_server;

void reconnect() {

*// Loop until we're reconnected*

*while* (!client.connected()) {

    Serial.print("Attempting MQTT connection...");

*// Attempt to connect*

*if* (client.connect("ESP32Client")) { *// 고유 id 아무거나 해도됌*

      Serial.println("connected");

*// Subscribe*

      client.subscribe(topic.c\_str());

    } *else* {

      Serial.print("failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 5 seconds");

*// Wait 5 seconds before retrying*

      delay(5000);

    }

  }

}

String getEncryptionType(int *type*) {

*switch* (*type*) {

*case* WIFI\_AUTH\_OPEN: *return* "open";

*case* WIFI\_AUTH\_WEP: *return* "WEP";

*case* WIFI\_AUTH\_WPA\_PSK: *return* "WPA";

*case* WIFI\_AUTH\_WPA2\_PSK: *return* "WPA2";

*case* WIFI\_AUTH\_WPA\_WPA2\_PSK: *return* "WPA+WPA2";

*case* WIFI\_AUTH\_WPA2\_ENTERPRISE: *return* "WPA2-EAP";

*case* WIFI\_AUTH\_WPA3\_PSK: *return* "WPA3";

*case* WIFI\_AUTH\_WPA2\_WPA3\_PSK: *return* "WPA2+WPA3";

*case* WIFI\_AUTH\_WAPI\_PSK: *return* "WAPI";

*default*: *return* "unknown";

  }

}

void scanNetworks() {

  n = WiFi.scanNetworks();

*for* (int i = 0; i < n; ++i) {

    ssidList[i] = WiFi.SSID(i);

    rssiList[i] = String(WiFi.RSSI(i));

    channelList[i] = String(WiFi.channel(i));

    encryptionList[i] = getEncryptionType(WiFi.encryptionType(i));

  }

*// 메모리 해제*

  WiFi.scanDelete();

}

String getConnectionStatuss() {

  String status;

*if* (WiFi.status() == WL\_CONNECTED) {

    status += "WiFi connected<br>";

  } *else* {

    status += "WiFi not connected<br>";

  }

*if* (mqtt\_server.length() != 0 && client.connected()) {

    status += "MQTT connected<br>";

  } *else* {

    status += "MQTT not connect<br>";

  }

*return* status;

}

## Setup 1013~1248 ap를 열고 apsta를 동시에 사용할 수 있도록 모드를 지정ap서버를 열어준다.

*Serial.print("Setting soft access point mode");*

*WiFi.softAP(ssid, password);*

*// wifi\_ap모드가 wifi 와 ap연결을 동시에 할 수 있도록 함.*

*WiFi.mode(WIFI\_MODE\_APSTA);*

*IPAddress IP = WiFi.softAPIP();*

*Serial.print("AP IP address: ");*

*Serial.println(IP);*

*scanNetworks();*

*server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){*

*// 쿼리 파라미터를 가져옵니다.*

*if (request->hasParam("wifissid")) {*

*wifissid = request->getParam("wifissid")->value();*

*}*

*if (request->hasParam("wifipassword")) {*

*wifipassword = request->getParam("wifipassword")->value();*

*}*

*if (request->hasParam("topic")) {*

*topic = request->getParam("topic")->value();*

*}*

*if (request->hasParam("mqtt\_server")) {*

*mqtt\_server = request->getParam("mqtt\_server")->value();*

*}*

*String html = "<!DOCTYPE html>\*

*<html>\*

*<head>\*

*<style>\*

*.scrollbox {\*

*width: 750px;\*

*height: 250px;\*

*overflow: auto;\*

*border: 1px solid #ccc;\*

*padding: 10px;\*

*margin: 0 auto;\*

*}\*

*body {\*

*text-align: center;\*

*}\*

*.header {\*

*text-align: center;\*

*}\*

*.button-container {\*

*text-align: right;\*

*margin-top: 10px;\*

*}\*

*button {\*

*margin: 10px;\*

*}\*

*</style>\*

*</head>\*

*<body>\*

*<div class=\"header\">\*

*<h1>WiFi Networks</h1>\*

*</div>\*

*<div class=\"button-container\">\*

*<button onclick=\"location.href='/'\">Main Page</button>\*

*<button onclick=\"location.href='/data'\">Sensor Data</button>\*

*</div>\*

*<div class=\"scrollbox\">\*

*";*

*if (n == 0) {*

*html += "<p>No networks found</p>";*

*} else {*

*html += "<table>\*

*<tr><th>Nr</th><th>SSID</th><th>RSSI</th><th>Channel</th><th>Encryption</th></tr>";*

*for (int i = 0; i < n; ++i) {*

*html += "<tr>";*

*html += "<td>" + String(i + 1) + "</td>";*

*html += "<td>" + ssidList[i] + "</td>";*

*html += "<td>" + rssiList[i] + "</td>";*

*html += "<td>" + channelList[i] + "</td>";*

*html += "<td>" + encryptionList[i] + "</td>";*

*html += "</tr>";*

*}*

*html += "</table>";*

*}*

*html += "</div>\*

*<br>\*

*<center><h1>Kouno Allsense Soft access point</h1></center>\*

*<center><h2>Web Server</h2></center>\*

*<form id=\"wifiForm\">\*

*<input type=\"text\" name=\"wifissid\" placeholder=\"WiFi SSID\" required>\*

*<br>\*

*<input type=\"text\" name=\"wifipassword\" placeholder=\"WiFi Password\" required>\*

*<br>\*

*<input type=\"text\" name=\"topic\" placeholder=\"Topic\" required>\*

*<br>\*

*<input type=\"text\" name=\"mqtt\_server\" placeholder=\"Mqtt Server\" required>\*

*<br>\*

*<button type=\"submit\">Change WiFi</button>\*

*<br><br>\*

*</form>\*

*<div id=\"responseBox\" style=\"border:1px solid #ccc; height:200px; overflow-y:scroll;\">\*

*<p id='wifiSsid'>Wi-Fi SSID: " + wifissid + "</p>\*

*<p id='wifiPassword'>Wi-Fi password: " + wifipassword + "</p>\*

*<p id='topic'>topic: " + topic + "</p>\*

*<p id='mqttServer'>MQTT server: " + mqtt\_server + "</p>\*

*<p id='connectionStatus'>" + getConnectionStatuss() + "</p>\*

*</div>\*

*<script>\*

*function updateStatus() {\*

*fetch('/config')\*

*.then(response => response.json())\*

*.then(data => {\*

*document.getElementById('connectionStatus').innerHTML = data.connectionStatus;\*

*});\*

*}\*

*setInterval(updateStatus, 5000); // 5초마다 상태 업데이트\*

*</script>\*

*";*

*html += "</body>\*

*</html>";*

*request->send(200, "text/html", html);*

*});*

*server.on("/data", HTTP\_GET, [](AsyncWebServerRequest \*request){*

*String dataHtml = "<!DOCTYPE html>\*

*<html>\*

*<head>\*

*<style>\*

*.scrollbox {\*

*width: 750px;\*

*height: 250px;\*

*overflow: auto;\*

*border: 1px solid #ccc;\*

*padding: 10px;\*

*margin: 0 auto;\*

*}\*

*body {\*

*text-align: center;\*

*}\*

*.header {\*

*text-align: center;\*

*}\*

*.button-container {\*

*text-align: right;\*

*margin-top: 10px;\*

*}\*

*button {\*

*margin: 10px;\*

*}\*

*</style>\*

*</head>\*

*<body>\*

*<div class=\"header\">\*

*<h1>WiFi Networks</h1>\*

*</div>\*

*<div class=\"button-container\">\*

*<button onclick=\"location.href='/'\">Main Page</button>\*

*<button onclick=\"location.href='/data'\">Sensor Data</button>\*

*</div>\*

*";*

*dataHtml += "<center><h1>Allsense Data</h1></center>\*

*<center><h2>Web Server</h2></center>\*

*<div id=\"responseBox\" style=\"border:1px solid #ccc; height:300px; overflow-y:scroll;\">\*

*<p id='wifiSsid'>Wi-Fi SSID: " + wifissid + "</p>\*

*<p id='wifiPassword'>Wi-Fi password: " + wifipassword + "</p>\*

*<p id='topic'>topic: " + topic + "</p>\*

*<p id='mqttServer'>MQTT server: " + mqtt\_server + "</p>\*

*<p id='connectionStatus'>" + getConnectionStatuss() + "</p>\*

*<p id='o3'></p>\*

*<p id='co'></p>\*

*<p id='no2'></p>\*

*<p id='nh3'></p>\*

*<p id='c3h8'></p>\*

*<p id='c4h10'></p>\*

*<p id='ch4'></p>\*

*<p id='h2'></p>\*

*<p id='c2h5oh'></p>\*

*<p id='ch2o'></p>\*

*<p id='srawVoc'></p>\*

*<p id='co2'></p>\*

*<p id='temperature'></p>\*

*<p id='humidity'></p>\*

*</div>\*

*<script>\*

*function updateStatus() {\*

*fetch('/config')\*

*.then(response => response.json())\*

*.then(data => {\*

*document.getElementById('connectionStatus').innerHTML = data.connectionStatus;\*

*document.getElementById('o3').innerHTML = 'O3: ' + data.o3 + ' ppm';\*

*document.getElementById('co').innerHTML = 'CO: ' + data.co + ' ppm';\*

*document.getElementById('no2').innerHTML = 'NO2: ' + data.no2 + ' ppm';\*

*document.getElementById('nh3').innerHTML = 'NH3: ' + data.nh3 + ' ppm';\*

*document.getElementById('c3h8').innerHTML = 'C3H8: ' + data.c3h8 + ' ppm';\*

*document.getElementById('c4h10').innerHTML = 'C4H10: ' + data.c4h10 + ' ppm';\*

*document.getElementById('ch4').innerHTML = 'CH4: ' + data.ch4 + ' ppm';\*

*document.getElementById('h2').innerHTML = 'H2: ' + data.h2 + ' ppm';\*

*document.getElementById('c2h5oh').innerHTML = 'C2H5OH: ' + data.c2h5oh + ' ppm';\*

*document.getElementById('ch2o').innerHTML = 'CH2O: ' + data.ch2o + ' ppm';\*

*document.getElementById('srawVoc').innerHTML = 'sRaw VOC: ' + data.srawVoc;\*

*document.getElementById('co2').innerHTML = 'CO2: ' + data.co2 + ' ppm';\*

*document.getElementById('temperature').innerHTML = 'Temperature: ' + data.temperature + ' °C';\*

*document.getElementById('humidity').innerHTML = 'Humidity: ' + data.humidity + ' %';\*

*});\*

*}\*

*setInterval(updateStatus, 500);\*

*</script>\*

*";*

*dataHtml += "</body>\*

*</html>";*

*request->send(200, "text/html", dataHtml);*

*});*

*server.on("/config", HTTP\_GET, [](AsyncWebServerRequest \*request) {*

*String jsonResponse = "{";*

*jsonResponse += "\"connectionStatus\": \"" + getConnectionStatuss() + "\",";*

*jsonResponse += "\"o3\": " + String(taskParm2.lpPktData->mq131) + ",";*

*jsonResponse += "\"co\": " + String(taskParm2.lpPktData->co) + ",";*

*jsonResponse += "\"no2\": " + String(taskParm2.lpPktData->no2) + ",";*

*jsonResponse += "\"nh3\": " + String(taskParm2.lpPktData->nh3) + ",";*

*jsonResponse += "\"c3h8\": " + String(taskParm2.lpPktData->c3h8) + ",";*

*jsonResponse += "\"c4h10\": " + String(taskParm2.lpPktData->c4h10) + ",";*

*jsonResponse += "\"ch4\": " + String(taskParm2.lpPktData->ch4) + ",";*

*jsonResponse += "\"h2\": " + String(taskParm2.lpPktData->h2) + ",";*

*jsonResponse += "\"c2h5oh\": " + String(taskParm2.lpPktData->c2h5oh) + ",";*

*jsonResponse += "\"ch2o\": " + String(taskParm2.lpPktData->ch2o) + ",";*

*jsonResponse += "\"srawVoc\": " + String(taskParm2.lpPktData->voc) + ",";*

*jsonResponse += "\"co2\": " + String(taskParm2.lpPktData->co2) + ",";*

*jsonResponse += "\"temperature\": " + String(taskParm2.lpPktData->temp) + ",";*

*jsonResponse += "\"humidity\": " + String(taskParm2.lpPktData->humi);*

*jsonResponse += "}";*

*request->send(200, "application/json", jsonResponse);*

*});*

*server.begin();*

## Loop 2228~2285 전달된 값으로 wifi와 mqtt연결 task의 센서값들을 mqtt 발행

*if ((wifissid.length() != 0) && (wifipassword.length() != 0) && (WiFi.status() != WL\_CONNECTED)) {*

*Serial.println("\nSetting Station configuration ... ");*

*WiFi.begin(wifissid.c\_str(), wifipassword.c\_str());*

*Serial.println(String("Connecting to ")+ wifissid);*

*int attemptCount = 0; // 시도 횟수 카운터*

*while (WiFi.status() != WL\_CONNECTED && attemptCount < 1000){*

*delay(500);*

*Serial.print(".");*

*attemptCount++;*

*}*

*if (WiFi.status() == WL\_CONNECTED) {*

*Serial.println("\nConnected, IP address: ");*

*Serial.println(WiFi.localIP());*

*} else {*

*Serial.println("\nFailed to connect after 10 attempts.");*

*}*

*}*

*if(mqtt\_server.length()!=0 && !client.connected()){*

*// && !client.connected()*

*client.setServer(mqtt\_server.c\_str(), 1883);*

*if (!client.connected()) {*

*reconnect();*

*}*

*// String messageChar = "hellow";*

*// client.publish(topic.c\_str(), messageChar.c\_str());*

*// Serial.print("hellow");*

*}*

*if (client.connected() && (WiFi.status() == WL\_CONNECTED)) {*

*float o3 = taskParm2.lpPktData->mq131;*

*float co = taskParm2.lpPktData->co;*

*float no2 = taskParm2.lpPktData->no2;*

*float nh3 = taskParm2.lpPktData->nh3;*

*float c3h8 = taskParm2.lpPktData->c3h8;*

*float c4h10 = taskParm2.lpPktData->c4h10;*

*float ch4 = taskParm2.lpPktData->ch4;*

*float h2 = taskParm2.lpPktData->h2;*

*float c2h5oh = taskParm2.lpPktData->c2h5oh;*

*float ch2o = taskParm2.lpPktData->ch2o;*

*uint16\_t srawVoc = taskParm2.lpPktData->voc;*

*float co2 = taskParm2.lpPktData->co2;*

*float temperature = taskParm2.lpPktData->temp;*

*float humidity = taskParm2.lpPktData->humi;*

*char messageChar[255];*

*int len = snprintf(messageChar, sizeof(messageChar),"VOC:%u CH2O:%.4f MQ131:%.4f CO:%.4f NO2:%.4f NH3:%.4f C3H8:%.4f C4H10:%.4f CH4:%.4f H2:%.4f C2H5OH:%.4f CO2:%.4f Temperature:%.4f Humidity:%.4f", srawVoc, ch2o, o3, co, no2, nh3, c3h8, c4h10, ch4, h2, c2h5oh, co2, temperature, humidity);*

*if (len < 0) {*

*// snprintf에서 에러 발생*

*Serial.printf("Error generating message\n");*

*} else if (len >= sizeof(messageChar)) {*

*// 메시지가 버퍼를 초과함*

*Serial.printf("Message truncated\n");*

*} else {*

*// 정상적으로 메시지 생성됨*

*delay(500);*

*client.publish(topic.c\_str(), messageChar);*

*}*

*}*