

# Innovation And Embedded System (539305)

ภาคการศึกษาที่ 2 ปีการศึกษา 2564

อาจารย์ ดร.มานพ ธิมาปะโท

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีสุรนารี

Suranaree University of Technology

# Course Outline 539305 2(0-6-9)

Category	Item	Topic	Week																							
			0		1		2		3		4		5		6		7		8		9		10		11	
			0.1	0.2	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	6.1	6.2	7.1	7.2	8.1	8.2	9.1	9.2	10.1	10.2	11.1	11.2
Introduction	1	Introduction (Outline, Score)					C	-																		
	2	Project					C	-																		
MQTT	3	MQTT Broker							C	L																
	4	MQTT pub sub client							C	L																
JSON	5	JSON									C	L														
	6	Arduino JSON									C	L														
Python MQTT	7	Python MQTT											C	L												
	8	Pub sub											C	L												
	9	Python MQTT Application											C	L												
Python-MQTT Project	10	Work													M	-		P								
	11	Present													M	-		P								
LoRAWAN	12	LoRa																	C	C						
	13	LoRaWAN																	C	C						
	14	LoRaWAN practical																	C	C						
Node-red	15	Node-red																			C	L				
	16	JS																			C	L				
	17	Node-red LoRaWAN application																			C	L				
Node-red Project	18	Start																					F	-		P
	19	Update																					F	-		P
	20	Final																					F	-		P

C = Class  
L = Lab  
M=Midterm

P = Present  
F= Final

# Outline Week03

- ☐ JSON
- ☐ Arduino JSON library
- ☐ Pubsubclient library

# JASON

## ❑ **JSON: JavaScript Object Notation.**

- Lightweight data-interchange format
- Self-describing and easy to understand
- Language independent: JSON uses JavaScript syntax, but the **JSON format is text only**.
- Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

## ❑ **Syntax:** {"name":value}

## ❑ ในภาษา python JASON คือ String ของ Dictionary มี Syntax คือ "{"name":value}"

## ❑ ประเภทข้อมูลที่ JSON เก็บได้มีดังนี้

- string
- number
- object (JSON object)
- array
- boolean
- null

# JSON Syntax Rules

- ❑ ข้อมูลถูกจัดให้อยู่ในรูปคู่ของ name/value ซึ่งจะถูกแยกโดยเครื่องหมาย :
- ❑ ข้อมูลแต่ละชุดจะถูกแยกโดยเครื่องหมาย commas
- ❑ ข้อมูลชนิด JSON จะถูกบรรจุอยู่ในเครื่องหมาย { }
- ❑ ถ้านำไปเก็บข้อมูลชนิด array จะใช้เครื่องหมาย [ ] ในการบรรจุข้อมูล

## รูปแบบการเก็บข้อมูลชนิด JSON

```
{ "name": "John" }

{ "age": 30 }

{ name: "John", age: 31, city: "New York" };

{
  "employees": [ "Jennie", "Lisa", "Jisoo" ]
}

{
  "employee": { "name": "John", "age": 30, "city": "New York" }
}
```

## รูปแบบการเขียนเพื่อให้อ่านเข้าใจง่าย

```
{
  "name": "John",
  "age": 30,
  "cars": [ "Ford", "BMW", "Fiat" ]
}

{
  "name": "John",
  "age": 30,
  "cars": [
    { "brand": "Ford", "models": [ "Fiesta", "Focus", "Mustang" ] },
    { "brand": "BMW", "models": [ "320", "X3", "X5" ] },
    { "brand": "Nissan", "models": [ "March", "Note" ] }
  ]
}
```

# JSON Data Types

- ❑ String

```
{"name":"Lisa"}
```

- ❑ Number

```
{"age":35}
```

- ❑ object (JSON object)

```
{"student":{"name":"Marry", "age":20, "city":"Korat"}}
```

- ❑ Array

```
{"student":["Patty", "Anna", "John"]}
```

- ❑ Boolean

```
{"married":true}
```

- ❑ Null

```
{"middlename":null}
```

# การสร้าง MQTT client ด้วย Arduino

❑ การใช้งาน Arduino เพื่อสื่อสารผ่าน MQTT protocol นิยมใช้ library ดังต่อไปนี้

❖ Pubsubclient

- เป็น library ที่ใช้สำหรับสื่อสารผ่าน MQTT ซึ่งชื่อของ library มาจากคำว่า publish และ subscribe ซึ่งจะสามารถสร้าง MQTT client ขึ้นมาโดยใช้ Arduino บอร์ดรุ่นต่างๆได้

❖ ArduinoJSON

- เป็น library สำหรับช่วยในการบรรจุข้อมูลให้อยู่ในรูป JSON เพื่อส่งออกไปทาง MQTT protocol และ เป็นเครื่องมือที่ใช้ในการแยกข้อมูลที่อยู่ในรูป JSON ออกมาใช้งาน
- มีตัวช่วยในการสร้างโค้ดต้นแบบให้ตรงกับรุ่นของไมโครคอนโทรลเลอร์ที่ใช้โดยไม่ต้องเขียนโค้ดทั้งหมดเอง
- มีเครื่องมืออำนวยความสะดวกอื่นๆอีกมากมาย



**Arduino PubSubClient**  
**Arduino Client for MQTT**



The best JSON library for embedded C++.  
**ESP32**

# pubsubclient library

Pubsubclient เป็น library Arduino สำหรับสื่อสารด้วย MQTT สามารถศึกษาข้อมูลเพิ่มเติมได้ที่ [pubsubclient](#) และ [API Documentation](#)

Library version: 2.8

## Constructor

- **PubSubClient** ()
- **PubSubClient** (client)
- **PubSubClient** (server, port, [callback], client, [stream])

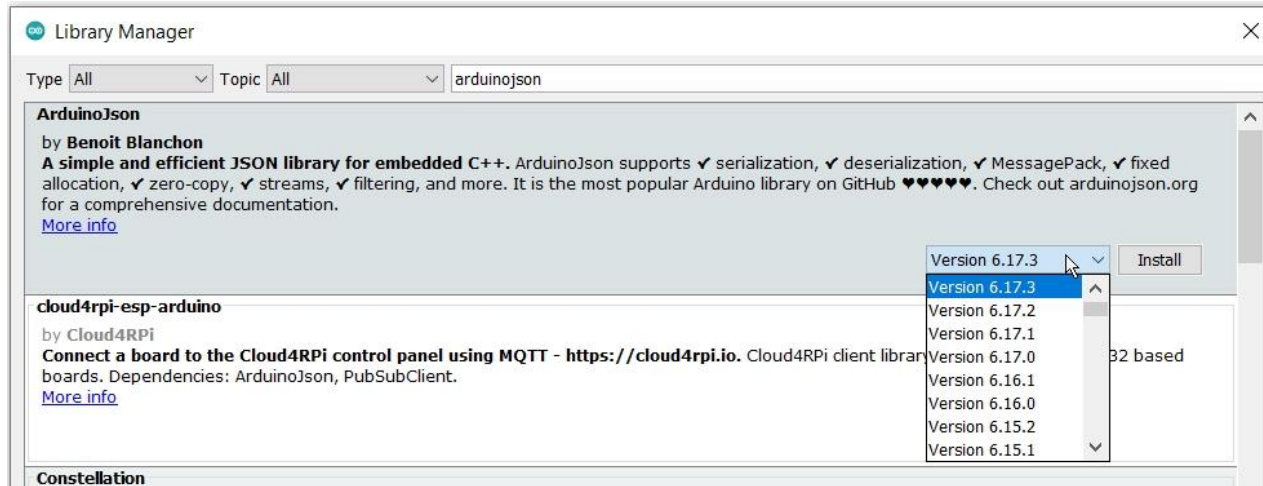
## Function

- boolean **connect** (clientId, [username, password], [willTopic, willQoS, willRetain, willMessage], [cleanSession])
- void **disconnect** ()
- boolean **publish** (topic, payload, [length], [retained])
- boolean **publish\_P** (topic, payload, [length], [retained])
- boolean **beginPublish** (topic, length, retained)
- int **write** (byte)
- int **write** (payload, length)
- boolean **endPublish** ()
- boolean **subscribe** (topic, [qos])
- boolean **unsubscribe** (topic)
- boolean **loop** ()
- boolean **connected** ()
- int **state** ()
- PubSubClient\* **setCallback** (callback)
- PubSubClient\* **setClient** (client)
- PubSubClient\* **setServer** (server, port)
- PubSubClient\* **setStream** (stream)
- uint16\_t **getBufferSize** ()
- boolean **setBufferSize** (size)
- PubSubClient\* **setKeepAlive** (keepAlive)
- PubSubClient\* **setSocketTimeout** (timeout)



# Arduino JSON library

## วิธีการติดตั้ง ArduinoJSON library



สามารถศึกษาข้อมูลเพิ่มเติมได้ที่ [ArduinoJSON.org](https://arduinojson.org)



The best JSON library for embedded C++.  
ESP32

# pubsubclient library

Pubsubclient เป็น library Arduino สำหรับสื่อสารด้วย MQTT สามารถศึกษาข้อมูลเพิ่มเติมได้ที่ [pubsubclient](#) และ [API Documentation](#)

Library version: 2.8

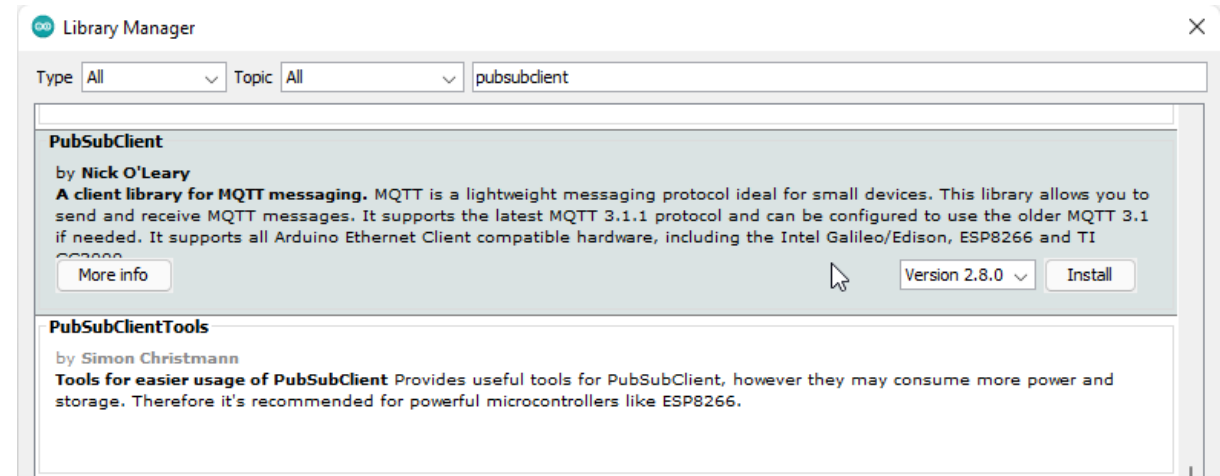
## Constructor

- **PubSubClient** ()
- **PubSubClient** (client)
- **PubSubClient** (server, port, [callback], client, [stream])

## Function

- boolean **connect** (clientId, [username, password], [willTopic, willQoS, willRetain, willMessage], [cleanSession])
- void **disconnect** ()
- boolean **publish** (topic, payload, [length], [retained])
- boolean **publish\_P** (topic, payload, [length], [retained])
- boolean **beginPublish** (topic, length, retained)
- int **write** (byte)
- int **write** (payload, length)
- boolean **endPublish** ()
- boolean **subscribe** (topic, [qos])
- boolean **unsubscribe** (topic)
- boolean **loop** ()
- boolean **connected** ()
- int **state** ()
- PubSubClient\* **setCallback** (callback)
- PubSubClient\* **setClient** (client)
- PubSubClient\* **setServer** (server, port)
- PubSubClient\* **setStream** (stream)
- uint16\_t **getBufferSize** ()
- boolean **setBufferSize** (size)
- PubSubClient\* **setKeepAlive** (keepAlive)
- PubSubClient\* **setSocketTimeout** (timeout)

## วิธีการติดตั้ง pubsubclient library



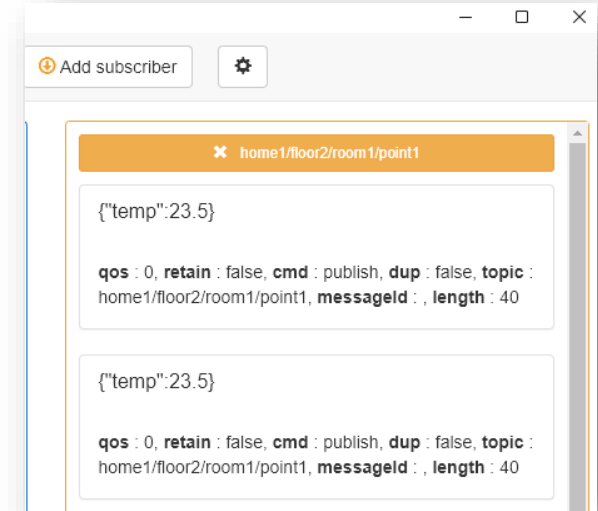
# การ **publish** ข้อมูล ด้วย PubSubClient library

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient client;
PubSubClient mqtt(client);
#define WIFI_STA_NAME "@SUT-IoT"
#define WIFI_STA_PASS ""
#define MQTT_SERVER "electsut.trueddns.com"
#define MQTT_PORT 27860
#define MQTT_USERNAME ""
#define MQTT_PASSWORD ""
#define MQTT_NAME "manotESP32"
#define LED_PIN 23
#define sw 5

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_STA_NAME);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
  }
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  mqtt.setServer(MQTT_SERVER, MQTT_PORT);
}
```

```
void loop() {
  if (mqtt.connected() == false) {
    Serial.print("MQTT connecting... ");
    if (mqtt.connect(MQTT_NAME, MQTT_USERNAME, MQTT_PASSWORD))
    {
      Serial.println("connected");
    }
    else {
      Serial.println("failed");
      delay(1000);
    }
  }
  else {
    mqtt.loop();
    mqtt.publish("home1/floor2/room1/point1", "{\"temp\":23.5}");
    delay(1000);
  }
}
```

## ข้อมูลที่รับทาง MQTT box



# การ subscribe ข้อมูล ด้วย PubSubClient library

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient client;
PubSubClient mqtt(client);
#define WIFI_STA_NAME "@SUT-IoT"
#define WIFI_STA_PASS ""
#define MQTT_SERVER "electsut.trueddns.com"
#define MQTT_PORT 27860
#define MQTT_USERNAME ""
#define MQTT_PASSWORD ""
#define MQTT_NAME "manotESP32"
#define LED_PIN 23
#define sw 5

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_STA_NAME);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
  }

  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  mqtt.setServer(MQTT_SERVER, MQTT_PORT);
  mqtt.setCallback(callback);
}
```

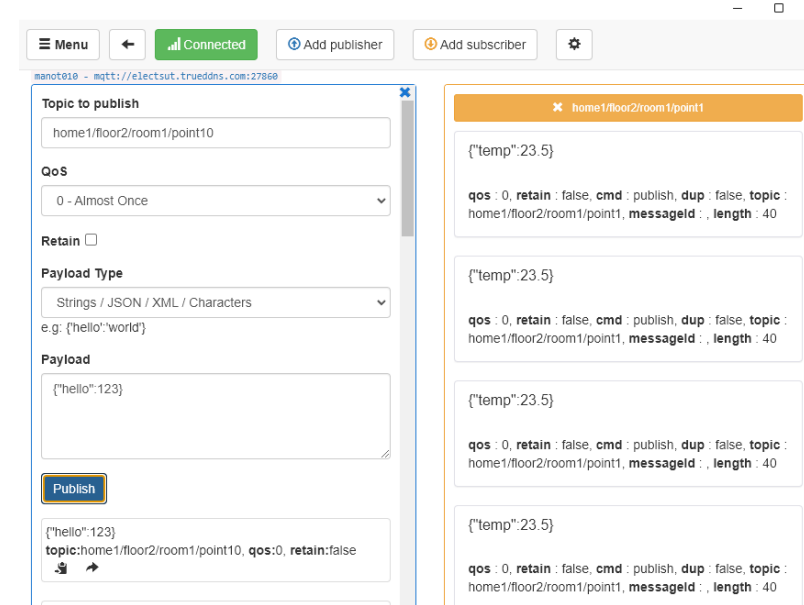
```
void loop() {
  if (mqtt.connected() == false) {
    Serial.print("MQTT connection... ");
    if (mqtt.connect(MQTT_NAME, MQTT_USERNAME, MQTT_PASSWORD))
    {
      Serial.println("connected");
      mqtt.subscribe("home1/floor2/room1/point3");
      mqtt.subscribe("home1/floor2/room1/point4");
      mqtt.subscribe("home1/floor2/room1/point5");
      mqtt.subscribe("home1/floor2/room1/point6");
      mqtt.subscribe("home1/floor2/room1/point7");
      mqtt.subscribe("home1/floor2/room1/point8");
      mqtt.subscribe("home1/floor2/room1/point9");
      mqtt.subscribe("home1/floor2/room1/point10");
    }
    else {
      Serial.println("failed");
      delay(1000);
    }
  }
  else {
    mqtt.loop();
    mqtt.publish("home1/floor2/room1/point1", "{\"temp\":23.5}");
    delay(1000);
  }
}
```

## ข้อมูลที่ได้รับ และพิมพ์ออกมาทาง Serial port

```
COM4

Message arrived [home1/floor2/room1/point3] {"hello":123}
Message arrived [home1/floor2/room1/point4] {"hello":123}
Message arrived [home1/floor2/room1/point5] {"hello":123}
Message arrived [home1/floor2/room1/point6] {"hello":123}
Message arrived [home1/floor2/room1/point7] {"hello":123}
Message arrived [home1/floor2/room1/point8] {"hello":123}
Message arrived [home1/floor2/room1/point9] {"hello":123}
Message arrived [home1/floor2/room1/point10] {"hello":123}
```

## ข้อมูลที่ publish จาก MQTT box



## ข้อมูลที่ Sub บน MQTT box

# การใช้ Arduino JSON library

- ❑ การใช้งาน Arduino JSON มีอยู่ 2 รูปแบบหลักได้แก่
  - ❖ Serialization เป็นการบรรจุข้อมูลให้อยู่ในรูปแบบของ JSON เช่น

```
float temp =25.8
float humid= 75.3
```

    - รูปแบบที่ได้หลังจากการทำ Serialize คือ

```
{"temp":25.8,"humid":75.3}
```
  - ❖ Deserialization เป็นการนำข้อมูลในรูปแบบ JSON ที่ได้รับ มาแยกออกเป็นชื่อตัวแปรและค่าที่ได้รับ เช่นข้อมูลที่ได้รับ 

```
{"temp":25.8,"humid":75.3}
```

    - สามารถแยกข้อมูลออกมาในรูปแบบของตัวแปร

```
float temp =25.8
float humid= 75.3
```
- ❑ หากไม่ใช้ Arduino JSON library การทำงานจะค่อนข้างลำบากในการบรรจุข้อมูลเพื่อส่งไปยัง MQTT broker นอกจากนี้การแยกข้อมูลจาก JSON เพื่อนำมาใช้งานบนไมโครคอนโทรลเลอร์ ก็ค่อนข้างลำบากด้วยเช่นกัน

# การใช้ serialize ข้อมูลให้ MQTT

```
DynamicJsonDocument doc(1024);

doc["sensor"] = "gps";
doc["time"] = 1351824120;
doc["data"][0] = 48.756080;
doc["data"][1] = 2.302038;

serializeJson(doc, Serial);
// This prints:
// {"sensor":"gps","time":1351824120,"data":[48.756080,2.302038]}
```

```
#include <ArduinoJson.h>

void setup() {
  Serial.begin(9600);
  while (!Serial) continue;

  StaticJsonDocument<200> doc;

  doc["sensor"] = "gps";
  doc["time"] = 1351824120;

  JsonArray data = doc.createNestedArray("data");
  data.add(48.756080);
  data.add(2.302038);

  serializeJson(doc, Serial);
  // The above line prints:
  // {"sensor":"gps","time":1351824120,"data":[48.756080,2.302038]}

  // Start a new line
  Serial.println();

  // Generate the prettified JSON and send it to the Serial port.
  //
  serializeJsonPretty(doc, Serial);
  // The above line prints:
  // {
  //   "sensor": "gps",
  //   "time": 1351824120,
  //   "data": [
  //     48.756080,
  //     2.302038
  //   ]
  // }
}

void loop() {
  // not used in this example
}
```

# Arduino JSON Serialization

## 1. ตัวอย่างการใช้งาน Serialization

```
#include <ArduinoJson.h>
StaticJsonDocument<200> doc;
doc["sensor"] = "gps";
doc["time"] = 1351824120;
JsonArray data = doc.createNestedArray("data");
data.add(48.756080);
data.add(2.302038);
String output;
serializeJson(doc, output);
serializeJson(doc, Serial);
serializeJsonPretty(doc, Serial);
```

จองพื้นที่ 200; สามารถคำนวณพื้นที่ได้จาก [Link](#)

```
{"sensor":"gps","time":1351824120,"data":[48.756080,2.302038]}
```

```
{
  "sensor": "gps",
  "time": 1351824120,
  "data": [
    48.756080,
    2.302038
  ]
}
```

# การ de-serialize ข้อมูล MQTT

```
char json[] = "{\"sensor\":\"gps\",\"time\":1351824120,\"data\":[48.756080,2.302038]}\"";
```

```
DynamicJsonDocument doc(1024);
deserializeJson(doc, json);
```

```
const char* sensor = doc["sensor"];
long time          = doc["time"];
double latitude    = doc["data"][0];
double longitude    = doc["data"][1];
```

```
#include <ArduinoJson.h>

void setup() {
  // Initialize serial port
  Serial.begin(9600);
  while (!Serial) continue;
  StaticJsonDocument<200> doc;

  char json[] = "{\"sensor\":\"gps\",\"time\":1351824120,\"data\":[48.756080,2.302038]}\"";
  DeserializationError error = deserializeJson(doc, json);

  if (error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.f_str());
    return;
  }

  const char* sensor = doc["sensor"];
  long time = doc["time"];
  double latitude = doc["data"][0];
  double longitude = doc["data"][1];

  // Print values.
  Serial.println(sensor);
  Serial.println(time);
  Serial.println(latitude, 6);
  Serial.println(longitude, 6);
}

void loop() {
}
```



# การใช้งาน Arduino JSON library

## 2. ตัวอย่างการใช้งาน De-serialize

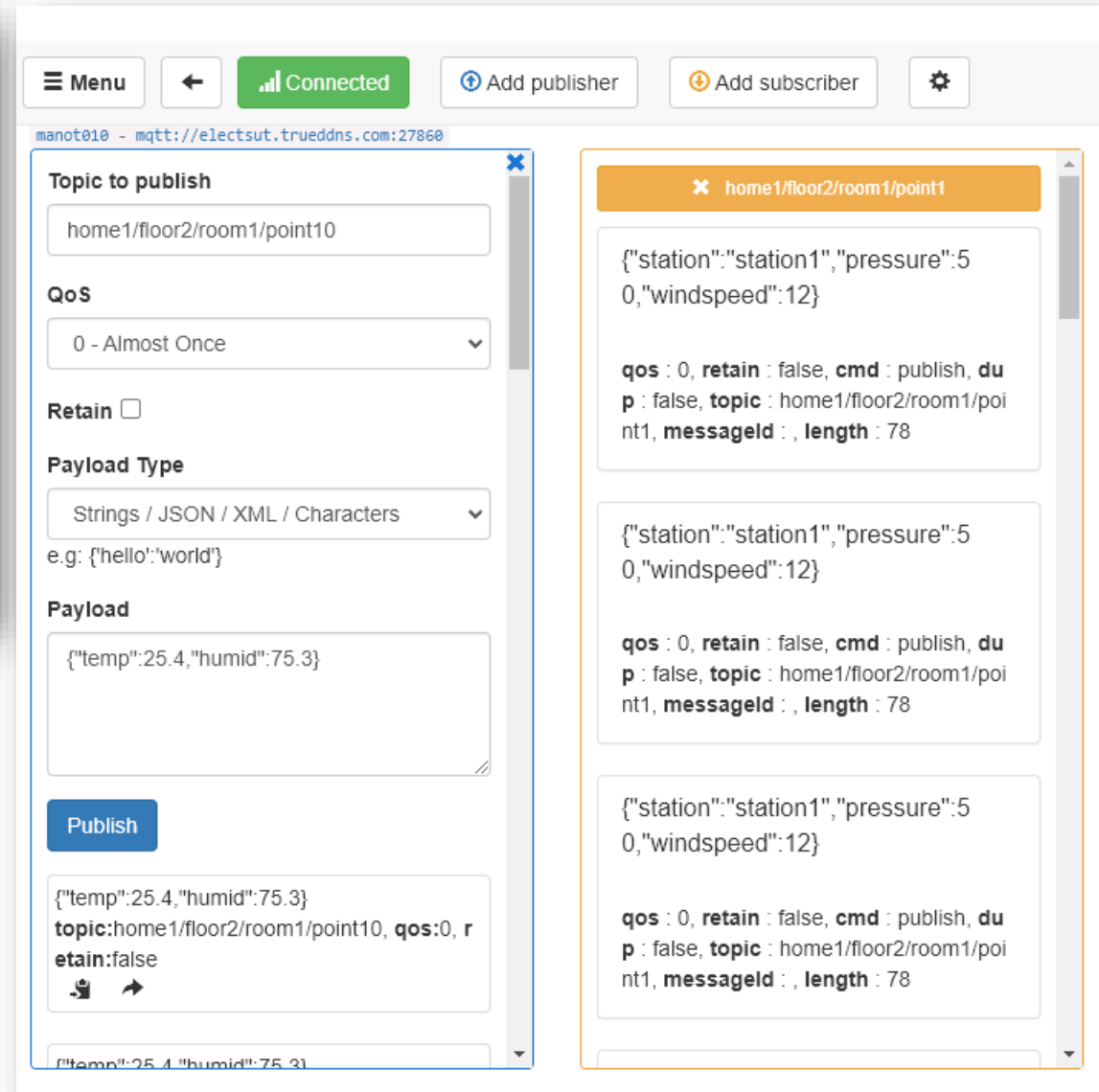
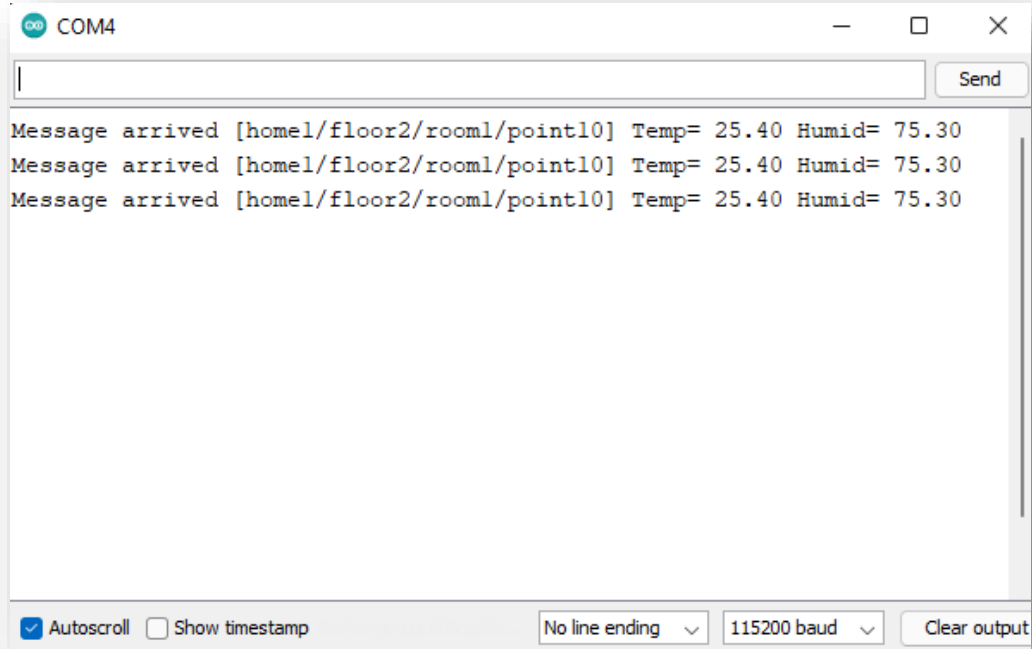
```
#include <ArduinoJson.h>
StaticJsonDocument<200> doc;
char json[] = "{\"sensor\":\"gps\",\"time\":1351824120,\"data\":[48.756080,2.302038]}";
DeserializationError error = deserializeJson(doc, json);
const char* sensor = doc["sensor"];
long time = doc["time"];
double latitude = doc["data"][0];
double longitude = doc["data"][1];

Serial.println(sensor);
Serial.println(time);
Serial.println(latitude);
Serial.println(longitude);

if (error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.f_str());
    return;
}
```

# ทดสอบใช้งาน Arduino JSON library

จากตัวอย่างการส่งข้อมูลด้วย MQTT จงใช้ Arduino JSON ในส่งและรับข้อมูล



## จากตัวอย่างการส่งข้อมูลด้วย MQTT จงใช้ Arduino JSON ในส่งและรับข้อมูล

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient client;
PubSubClient mqtt(client);
DynamicJsonDocument docd(1024);
DynamicJsonDocument docs(1024);
#define WIFI_STA_NAME "@SUT-IoT"
#define WIFI_STA_PASS ""
#define MQTT_SERVER "electsut.trueddns.com"
#define MQTT_PORT 27860
#define MQTT_USERNAME ""
#define MQTT_PASSWORD ""
#define MQTT_NAME "manotESP32"
#define LED_PIN 23
#define sw 5

//MQTT_NAME += String(random(0xffff), HEX);
void callback(char* topic, byte* payload, unsigned int length) {

    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    deserializeJson(docd, payload, length);
    float temp = docd["temp"];
    float humid = docd["humid"];
    Serial.print("Temp= ");
    Serial.print(temp);
    Serial.print(" Humid= ");
    Serial.println(humid);
}

void setup() {
    Serial.begin(115200);
    pinMode(LED_BUILTIN, OUTPUT);
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(WIFI_STA_NAME);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
        digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
    }

    digitalWrite(LED_BUILTIN, HIGH);
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    mqtt.setServer(MQTT_SERVER, MQTT_PORT);
    mqtt.setCallback(callback);
}

void loop() {
    if (mqtt.connected() == false) {
        Serial.print("MQTT connection... ");
        if (mqtt.connect(MQTT_NAME, MQTT_USERNAME, MQTT_PASSWORD))
        {
            Serial.println("connected");
            mqtt.subscribe("home1/floor2/room1/point10");
        }
        else {
            Serial.println("failed");
            delay(1000);
        }
    }

    else {
        mqtt.loop();
        String station="station1";
        float pressure=50;
        float windspeed=12;
        docs["station"]= "station1";
        docs["pressure"]=pressure;
        docs["windspeed"]=windspeed;
        char output[128];
        serializeJson(docs, output);
        mqtt.publish("home1/floor2/room1/point1", output);
        delay(1000);
    }
}
```

# ตัวอย่างเขียน serialize และ de-serialize

สามารถใช้ตัวสร้าง code อัตโนมัติได้ทั้ง **serialize** และ **de-serialize** ได้จาก [Link](#)

1

2

3

4

ConfigurationJSONSizeProgram

Step 4: Program

```
StaticJsonDocument<96> doc;

doc["sensor"] = "gps";
doc["time"] = 1351824120;

JsonArray data = doc.createNestedArray("data");
data.add(48.75608);
data.add(2.32038);

serializeJson(doc, output);
```

Copy

See also [Serialization Tutorial](#) [DynamicJsonDocument](#) [serializeJson\(\)](#)

1

2

3

4

ConfigurationJSONSizeProgram

Step 4: Program

```
// String input;

StaticJsonDocument<128> doc;

DeserializationError error = deserializeJson(doc, input);

if (error) {
  Serial.print(F("deserializeJson() failed: "));
  Serial.println(error.f_str());
  return;
}

const char* sensor = doc["sensor"]; // "gps"
long time = doc["time"]; // 1351824120

float data_0 = doc["data"][0]; // 48.75608
float data_1 = doc["data"][1]; // 2.32038
```

Copy

See also [Deserialization Tutorial](#) [DynamicJsonDocument](#) [deserializeJson\(\)](#)

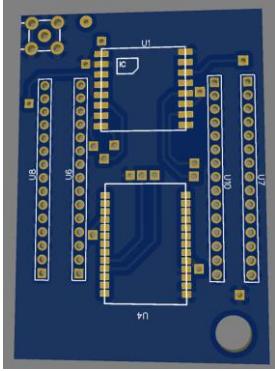
# Backup

Arduino sub sensor

Arduino sub LED on-off

Arduino pub Led status

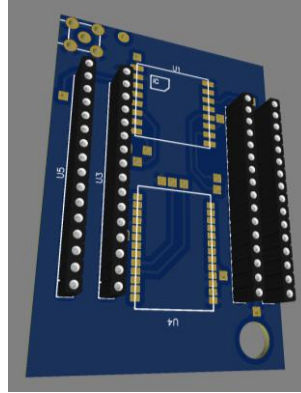
Arduino pub ADC val



PCB 2 Layer



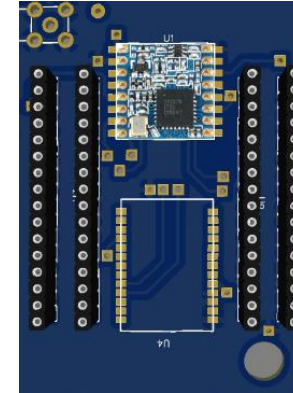
Socket 15 pin x4



PCB+Socket+Assembly



LaRa Sx1276



Fullset