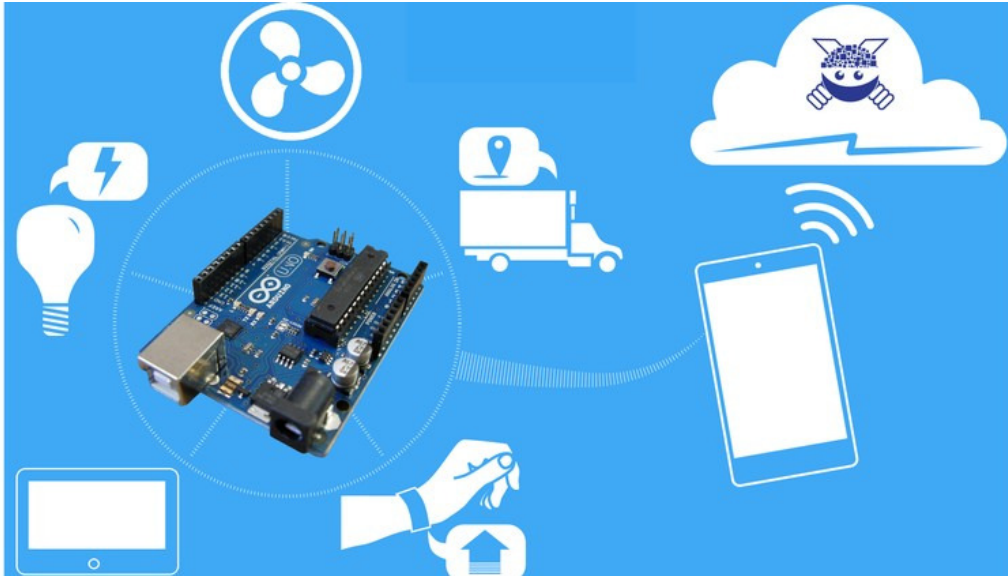
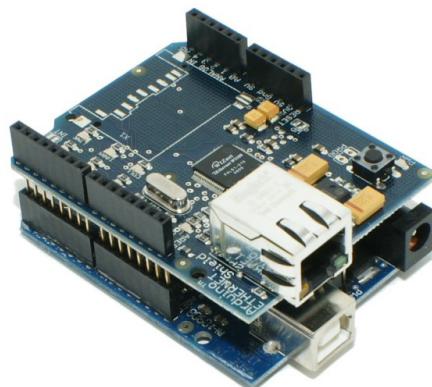


Building IoT with Arduino



Ethernet Shield

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip. The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield.



Ethernet.begin()

Initializes the ethernet library and network settings.

- **Syntax**

```
Ethernet.begin(mac);  
Ethernet.begin(mac, ip);  
Ethernet.begin(mac, ip, dns);  
Ethernet.begin(mac, ip, dns, gateway);  
Ethernet.begin(mac, ip, dns, gateway, subnet);
```

- **Example** รับค่า IP จาก DHCP Server

```
#include <SPI.h>  
#include <Ethernet.h>  
  
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };  
  
void setup()  
{  
    Ethernet.begin(mac);  
    Serial.println(Ethernet.localIP());  
}  
  
void loop () {  
}
```

Note

Ethernet.begin(mac), returns an int: 1 on a successful DHCP connection, Int: 0 on failure

Note: Ethernet.localIP () ดึงค่า IP จาก Ethernet shield

Ethernet.begin()

Example ตั้งค่าแบบ Fix IP

```
#include <SPI.h>  
#include <Ethernet.h>  
  
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };  
IPAddress dnServer(192, 168, 0, 1);  
IPAddress gateway(192, 168, 0, 1);  
IPAddress subnet(255, 255, 255, 0);  
IPAddress ip(192, 168, 0, 2);  
  
void setup() {  
    Serial.begin(9600);  
  
    Ethernet.begin(mac, ip, dnServer, gateway, subnet);  
    Serial.println(Ethernet.localIP());  
}  
  
void loop () {  
}
```

Note: - Ethernet.localIP () ดึงค่า IP จาก Ethernet shield
 - IPAddress ประกาศค่า IP

Client

Functions

- **EthernetClient()** คือ กำหนดให้บอร์ด arduino ทำหน้าที่เป็น client ซึ่งสามารถสร้างการเชื่อมต่อไปยัง IP และ port ที่กำหนดได้ (Creates a client which can connect to a specified internet IP address and port)

เช่น `EthernetClient client;`

- **connect()** คือ เชื่อมต่อไปยัง IP และ port ที่กำหนด (Connects to a specified IP address and port)

☞ ถ้าเชื่อมต่อสำเร็จจะให้ค่าเป็น 1

เช่น `client.connect("www.google.com", 80)`

- **available()** จะคืนค่าเป็นจำนวน byte ที่ถูกเขียนมายัง client ซึ่งถูกส่งมาโดย server

Returns the number of bytes available for reading (that is, the amount of data that has been written to the client by the server it is connected to).

เช่น `if (client.available())`

ถ้ามีการส่งข้อมูลมา จะมีความหมายเป็น `if(true)` หรือ `if(!0)`

Client

Functions

- **read()** คือ อ่านค่า byte ถัดไป ที่ได้รับจาก Server

(Read the next byte received from the server the client is connected to (after the last call to read()).)

เช่น `char c = client.read();`

- **connected()** คือ ตรวจสอบว่า client ยังเชื่อมต่ออยู่หรือไม่ (Whether or not the client is connected.)

☞ ถ้ายังเชื่อมต่ออยู่ จะให้ค่าเป็น 1 (true) , ถ้าไม่ได้เชื่อมต่อแล้ว จะให้ค่า เป็น 0 (false)

เช่น `if (!client.connected())`

- **stop()** คือ ปิดการเชื่อมต่อกับ server

เช่น `client.stop();`

- **flush()** คือ clear ค่าที่ถูกเขียนมายัง client แต่ยังไม่ได้อ่าน

เช่น `client.flush();`

ติดตั้ง Library ของ MQTT Protocol

- Download Library Pub sub จาก

<https://github.com/knolleary/pubsubclient>

โดยการ download เป็น .zip

- แล้วติดตั้งที่ sketch -> Include Library -> Add .zip library
- เมื่อติดตั้งเสร็จแล้ว จะมี Library PubSubClient ปรากฏที่

sketch -> Include Library

ข้อจำกัด

- It can only publish QoS 0 messages. It can subscribe at QoS 0 or QoS 1.
- The maximum message size, including header, is **128 bytes** by default.

PubSubClient Library

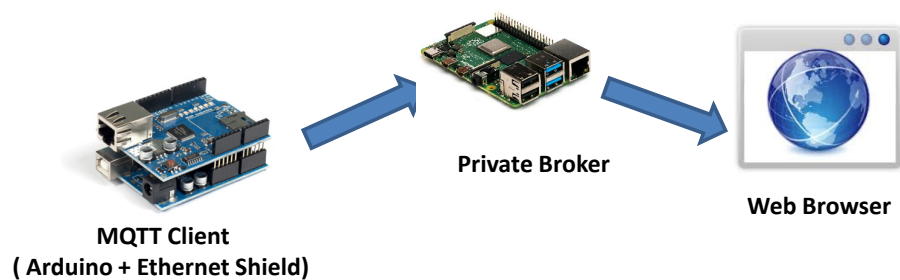
- **Constructors**
 - [PubSubClient \(\)](#)
 - [PubSubClient \(client\)](#)
 - [PubSubClient \(server, port, \[callback\], client, \[stream\]\)](#) ***
- **Functions**
 - [boolean connect \(clientID\)](#) ***
 - [boolean connect \(clientID, willTopic, willQoS, willRetain, willMessage\)](#)
 - [boolean connect \(clientID, username, password\)](#)
 - [boolean connect \(clientID, username, password, willTopic, willQoS, willRetain, willMessage\)](#)
 - [void disconnect \(\)](#)
 - [int publish \(topic, payload\)](#) ***
 - [int publish \(topic, payload, retained\)](#)
 - [int publish \(topic, payload, length\)](#)
 - [int publish \(topic, payload, length, retained\)](#)
 - [int publish_P \(topic, payload, length, retained\)](#)
 - [boolean subscribe \(topic, \[qos\]\)](#) ***
 - [boolean unsubscribe \(topic\)](#)
 - [boolean loop \(\)](#)
 - [int connected \(\)](#)
 - [int state \(\)](#)
 - [PubSubClient setServer \(server, port\)](#)
 - [PubSubClient setCallback \(callback\)](#)
 - [PubSubClient setClient \(client\)](#)
 - [PubSubClient setStream \(stream\)](#)
- **Other**
 - [Configuration Options](#)
 - [Subscription Callback](#)

การทดลองที่ 1. Publish

ขั้นตอนการทดลอง

1. ไปที่ <http://www.hivemq.com/demos/websocket-client/>
2. ใช้ host เป็น Private MQTT Broker ที่ URL 10.30.4.xxx
3. กด Connect (ระบบจะ random ค่า client ID อัตโนมัติ)
4. ไปที่ Subscriptions แล้ว Add new topic subscription

กำหนด Topic ที่ต้องการติดตาม

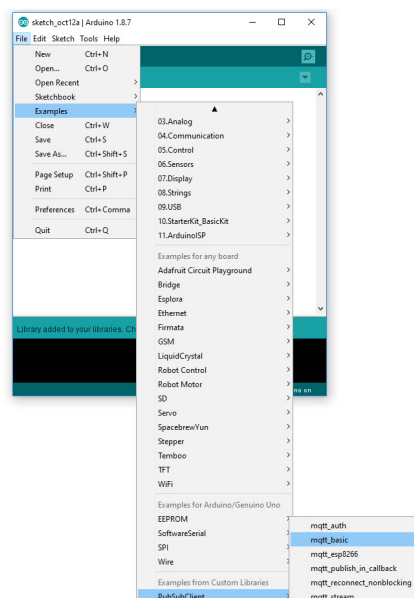


การทดลองที่ 1. Publish

4. เขียนโปรแกรมลง Arduino เพื่อทดลอง publish

เปิดโปรแกรมตัวอย่าง

โดยไปที่ File => Examples => PubSubclient => mqtt_basic



การทดลองที่ 1. Publish

แก้ไขโปรแกรมตัวอย่าง ดังนี้

```
#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
byte mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED }; 1. แก้ MAC Address อย่ายให้ซ้ำกับคนอื่น
IPAddress ip(172, 16, 0, 100); 2. ปิดบรรทัดนี้ โดยแก้เป็น // IPAddress ip(172,16,0,100)
IPAddress server(172, 16, 0, 2); 3. แก้เป็น IPAddress server(10,30,4,XXX)
                                ถ้าเป็น name ใช้เป็น char server[]="broker.mqttdashboard.com";

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();
}
```

การทดลองที่ 1. Publish

แก้ไขโปรแกรมตัวอย่าง ดังนี้

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("arduinoClient")) { 4. แก้ arduinoClient เป็นค่าอะไรก็ได้ อย่ายให้ซ้ำกันกับคนอื่น
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("outTopic", "hello world"); 5. แก้ outTopic เป็น Topic ที่ต้องการจะ Publish
      // ... and resubscribe                       แก้ข้อความ เป็นข้อความที่ต้องการ Publish
      client.subscribe("inTopic"); 6. แก้ inTopic เป็น Topic ที่ต้องการจะ Subscribe
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

การทดลองที่ 1. Publish

แก้ไขโปรแกรมตัวอย่าง ดังนี้

```
void setup()
{
  Serial.begin(57600);

  client.setServer(server, 1883);
  client.setCallback(callback);

  Ethernet.begin(mac, ip);    7. แก้เป็น Ethernet.begin(mac)
  // Allow the hardware to sort itself out
  delay(1500);
}

void loop()
{
  if (!client.connected()) {
    reconnect();
  }
  client.loop();  8. ปิดบรรทัดนี้ โดยแก้เป็น //client.loop()
                9. เพิ่ม code ดังนี้
                client.publish("outTopic", "hello world");
                delay(5000);
}
```

อย่าลืมแก้ outTopic เป็น Topic ที่ต้องการจะ Publish
แก้ข้อความ เป็นข้อความที่ต้องการ Publish

การทดลองที่ 1. Publish

5. เปิดโปรแกรม Serial monitor เพื่อดูสถานะต่างๆ
และ สังเกตข้อความที่ได้รับ (Message) จากการ publish โดย Arduino

The screenshot displays the MQTT client interface with the following sections:

- Connection:** Shows a green dot and the text "connected".
- Publish:** The "Topic" field contains "testtopic/1", "QoS" is set to 0, and the "Retain" checkbox is unchecked. The "Message" field contains the Thai text "โหว่า". A "Publish" button is visible.
- Subscriptions:** A list of subscriptions is shown, including "testtopic/1" with a QoS of 0.
- Messages:** A list of received messages is shown, including "but arduino can't T_T", "Publish from Arduino", and "test from panwit".

การทดลองที่ 2. Subscribe

1. เขียน โปรแกรมลง Arduino เพื่อทดลอง Subscribe

แก้โปรแกรมตัวอย่าง เหมือนการทดลองที่แล้ว แค่ข้อ 1-7

```
void setup()
{
    Serial.begin(57600);

    client.setServer(server, 1883);
    client.setCallback(callback);

    Ethernet.begin(mac, ip); 7. แก้เป็น Ethernet.begin(mac)
    // Allow the hardware to sort itself out
    delay(1500);
}

void loop()
{
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}
```

จะวน loop รอ เมื่อมีการ publish เข้ามา
โปรแกรมจะวิ่งไปทำงานที่ฟังก์ชัน callback

```
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();
}
```

การทดลองที่ 2. Subscribe

2. เปิดโปรแกรม Serial monitor โดย set baud rate เป็น 57600

และ publish Message โดย WebSocket ของ HiveMQ

และ สังเกตข้อความที่ได้รับ (Message) จากการ publish

The screenshot shows the HiveMQ web interface and an Arduino Serial Monitor window. The HiveMQ interface includes a 'Connection' status bar showing 'connected', a 'Publish' section with fields for 'Topic' (testtopic/1), 'QoS' (0), and 'Retain' (unchecked), and a 'Messages' section showing a list of received messages. The Arduino Serial Monitor window shows the output of the program, including the IP address (10.50.19.12), the MQTT connection status, and the received messages.

Connection ● connected

Publish

Topic: testtopic/1 QoS: 0 Retain: ☐ **Publish**

Message: test from websocket

Subscriptions **Add New Topic Subscription**

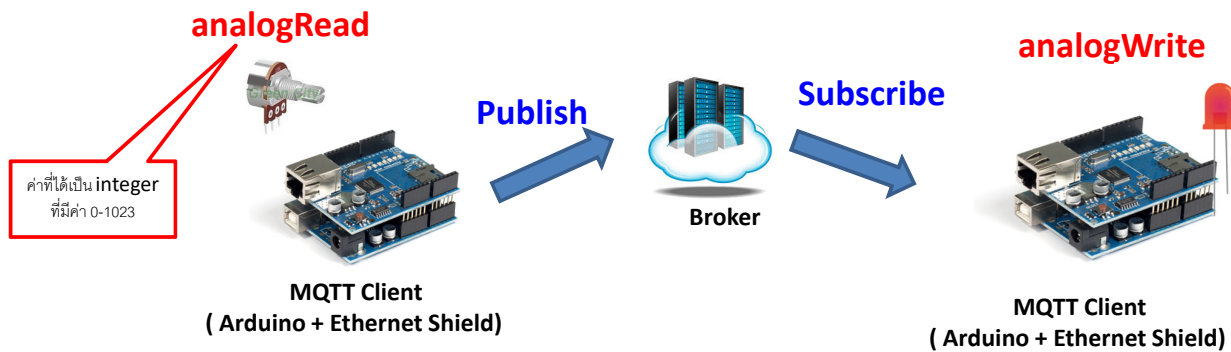
COM20 (Arduino Uno)

Net begin
IP: 10.50.19.12
Successfully connected with MQTT
testtopic/1
Publish from Arduino
testtopic/1
test from websocket

Messages

Time	Topic	Message
2015-04-08 09:02:25	testtopic/1	test from websocket
2015-04-08 09:01:34	testtopic/1	Publish from Arduino

ตัวอย่าง การรับส่งค่า



ฝั่งส่ง

ถ้าต้องการนำค่าที่ได้ที่เป็น integer มา publish จะต้องแปลงให้เป็น array ของ character ก่อน
เพิ่มโปรแกรมใน void loop ดังนี้

```
int val = analogRead(A2);  
String val_str = String(val);  
char val_char[10];  
val_str.toCharArray(val_char, 10);  
client.publish(outTopic, val_char);
```

รับค่ามาเป็นจำนวนเต็ม เลยต้องแปลงให้เป็น string โดยใช้คำสั่ง string จากนั้นแปลงจาก string เป็น array โดยใช้ toCharArray

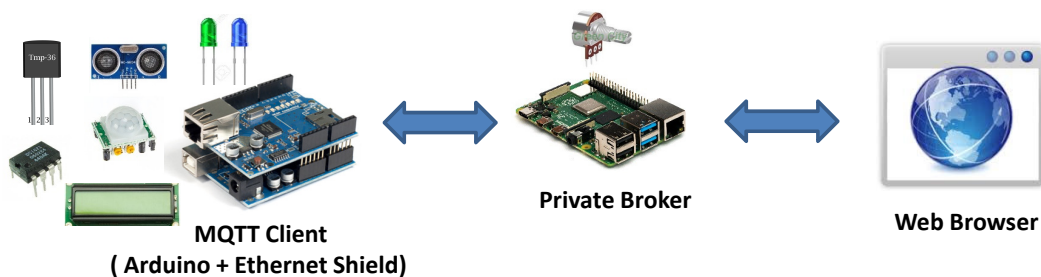
ฝั่งรับ

ถ้าต้องการนำค่าที่ subscribe มาได้ มาแปลงเป็น integer
เพิ่มโปรแกรมใน void callback ดังนี้

```
payload[length] = '\0';  
String payload_str = String((char*)payload);  
int val = payload_str.toInt();  
analogWrite(9, val);
```

รับมาเป็น array เลยต้องเพิ่ม \0 เพื่อให้เป็น string
จากนั้นแปลงจาก array เป็น string โดยใช้คำสั่ง string
และแปลงจาก string เป็น integer โดยใช้ toInt

การบ้านส่งสัปดาห์หน้า Put it all together



- Arduino อ่านค่าจาก sensor 4 ตัวได้แก่ PIR, Ultrasonic , TMP36 , DS1621
- แสดงผลออกทาง
 - จอ LCD แสดงสถานะ PIR, ระยะทาง , อุณหภูมิ
 - Website ไม่กำหนดรูปแบบ จะแสดงค่าอย่างไรก็ได้ ออกแบบให้สวยงาม
- ต่อตัวต้านทานปรับค่าได้ที่ raspberry pi และไปหรีหลอดไฟ LED ที่ติดตั้งที่ Arduino
- Website สั่งปิดเปิด (on-off) หลอด LED ที่ติดตั้งที่ Arduino ได้

สอบปลายภาคไม่มีสอบปฏิบัติ **ไม่มี project** มีแต่สอบทฤษฎีในห้องสอบ

การบ้านข้อนี้ก็จะเป็นคะแนนเอนะนิดนึงครับ ให้เวลาทำ 1 อาทิตย์