

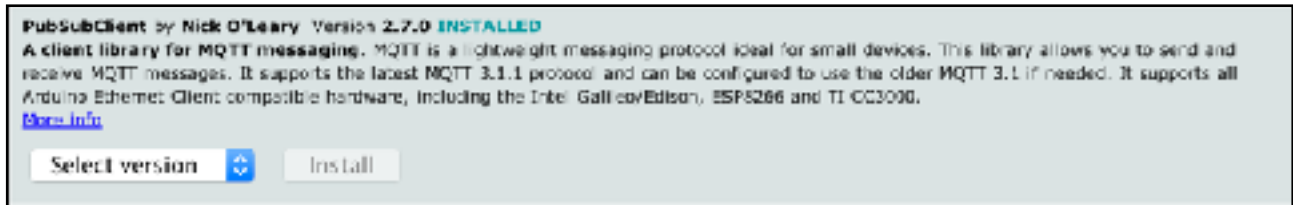
## Lab Sesi 4 - INIXINDO KEJAR

### Prasyarat:

Software MQTT Broker sudah terinstalasi di Laptop masing-masing

Sudah mempunyai Akun di situs CLOUD MQTT

Persiapkan Obyek dari **PubSubClient**. Modul ini harus lebih dahulu diinstalsi dari Library Manager Arduino IDE.



### Lab 1: Publisher-Subscriber pada Broker localhost

Buat Publisher dan Subscriber dengan host = localhost. Gunakan opsi -d untuk debugging.

```
$ mosquitto_pub -h localhost -t INIX -d -m "Pesan dari Inixindo"
```

Di terminal yang berbeda, tulis subscriber:

```
$ mosquitto_sub -h localhost -t INIX -d
```

Hasil: Tulis langkah DEBUG yang ditampilkan oleh Subscriber

PUBLISHER :

sending CONNECT

received CONNACK (0)

sending PUBLISH (d0, q0, r0, m1, 'INIX', ... (1 bytes))

sending DISCONN

SUBSCRIBER:


## Lab 2: Koneksi Cloud MQTT

Pindahkan Broker ke CLOUD MQTT, gunakan informasi account masing-masing.

```
$ mosquitto_pub -h soldier.cloudmqtt.com -u fjja -P jbWBZXt6W -p 17177 -t INIX -d -m "Pesan dari Inixindo"
```

Di terminal yang berbeda, tulis subscriber:

```
$ mosquitto_sub -h soldier.cloudmqtt.com -u fjja -P jbWBZXt6W -p 17177 -t INIX -d
```

## Lab 3: ESP32 Publisher ke Cloud MQTT

Sambungkan Microcontroller ESP32 ke laptop, buat sketch untuk menjadikan **ESP32 sebagai Publisher** dengan mengirim angka random setiap 10 detik.

Selipkan code berikut untuk mendapatkan angka random:

```
int nilai_acak;
```

```
randomseed( micros)); // mendapat angka sembarang
```

```
loop () {  
    nilai_acak = random(0,1000); // angka acak antar 0 sd 1000  
    .. ..  
}
```

Program pertama yang Anda harus buat adalah koneksi ke Internet melalui WiFi. Di sesi 3 telah didemonstrasikan *fungsi connectToHotspot()*;  
*Gunakan program yang sama !*

Setelah itu gunakan informasi dari Cloud MQTT untuk login:

```
const char* mqttServer = "soldier.cloudmqtt.com";  
const int mqttPort = 17177;  
const char* mqttUser = "fjja";  
const char* mqttPassword = "jbWBZX";
```

Kemudian login dengan menggunakan Obyek PubSubClient.

```
WiFiClient espClient;  
PubSubClient client(espClient);
```

```
void connectToMqttCloud() {  
    client.setServer(mqttServer, mqttPort);  
    while ( !client.connected() ) {  
        Serial.println("Mencoba koneksi dengan Broker MQTT...");  
        if (client.connect("ESP32Client", mqttUser, mqttPassword) ) {  
            Serial.println("tersambung dengan MQTT ");  
        } else {
```

```

    Serial.print("Error ");
    delay(5000); // coba 5 detik kemudian
  }
}
}

```

Kemudian sebagai payload (pesan yang dikirim) adalah angka acak yang diciptakan oleh fungsi random.

Siapkan variable value dan str\_value secara global (memudahkan).

```

int value; // nilai random
char str_value[8]; // asumsi tidak melebihi 8 karakter

void loop() {
  value= random(1,10000);
  sprintf(str_value,"%04d", value); // konversi integer ke string
  client.publish("INIX", str_value); // publish dengan topic="INIX"
  delay(10000); // 10 detik untuk angka random berikutnya
}

```

Gunakan terminal sebagai subscriber:

```

$ mosquitto_sub -h soldier.cloudmqtt.com -u fjjafgsy -P jbWBZXt6W75r -p 17177 -t INIX -d
Client mosq/dyV4kNNcges9cEgGfL sending CONNECT
Client mosq/dyV4kNNcges9cEgGfL received CONNACK (0)
Client mosq/dyV4kNNcges9cEgGfL sending SUBSCRIBE (Mid: 1, Topic: INIX, QoS: 0, Options:
0x00)
Client mosq/dyV4kNNcges9cEgGfL received SUBACK
Subscribed (mid: 1): 0
Client mosq/dyV4kNNcges9cEgGfL received PUBLISH (d0, q0, r0, m0, 'INIX', ... (5 bytes))
7337
Client mosq/dyV4kNNcges9cEgGfL received PUBLISH (d0, q0, r0, m0, 'INIX', ... (5 bytes))
8663
Client mosq/dyV4kNNcges9cEgGfL received PUBLISH (d0, q0, r0, m0, 'INIX', ... (5 bytes))
2437

```

## Challenge:

Ganti nilai random tersebut dengan membaca SENSOR (pilih salah satu yang memberikan nilai analog)

## Lab 4 (opsional): ESP32 sebagai Subscriber dari Cloud MQTT

Peran sebagai Subscriber sama dengan Publisher, hanya harus dideklarasikan sebagai subscriber.

```
void connectToMqttCloud() {
    String topic = "INIX";
    client.setServer(mqttServer, mqttPort);
    client.setCallback(receiveMessage);

    while ( !client.connected() ) {
        ...
        if (client.connect("ESP32Client", mqttUser, mqttPassword )) {
            ...
            client.subscribe(topic);
        } else {
            delay(5000); // coba 5 detik lagi
        }
    }
}
```

Karena bersifat pasif, yaitu menunggu pesan dari Publisher, program menggunakan teknik Callback, yang mana sistem akan memberikan interrupt pada event tersebut, dan kemudian mengaktifkan program **receiveMessage** .

```
void receiveMessage(char* topic, byte* payload, unsigned int length) {
    int ntimes;
    // hanya tertarik pada byte pertama saja --> payload [0]
    //konversi bytes menjadi integer
    ntimes = (int) ( (char)payload[0] - '0'); // input harus karakter '0' sd '9'
    toBlink(ntimes);
}
```

Payload berisi pesan dalam byte (karena itu harus dikonversi ke char). Length menyatakan panjang atau jumlah byte(s) yang diterima . Topic adalah topic yang sama dengan topic "subscribe". Topic ini menjadi parameter, karena subscribe dapat dilakukan pada lebih dari 1 topic.

Program tersebut menyatakan jumlah blink yang harus dilakukan ESP32 pada Internal LED. Contoh berikut adalah salah satu ide aplikasi.

```
void toBlink(int n) { // blink sebanyak n kali .....
    if (n<0 || n> 9) n= 1; // jika nilai input kacau, maka jadikan n=1
    for (int i=0; i<n; i++) {
        digitalWrite(BUILTIN_LED, HIGH);
        delay(1000);
        digitalWrite(BUILTIN_LED, LOW);
        delay(1000);
    }
}
```

Pada loop() perlu dicantumkan client.loop(), agar client tetap menunggu event dari Publisher. Bila Anda lupa mencantumkan ini, maka subscriber tidak berfungsi.

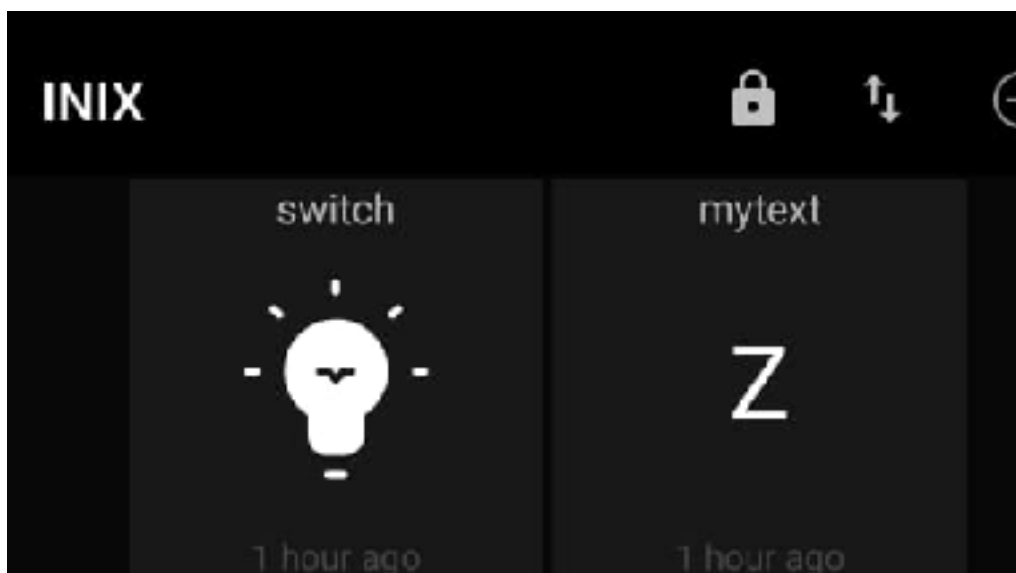
```
void loop() {  
  client.loop();  
  if (! client.connected() ) {  
    Serial.println("Disconnected ...Pls reconnect");  
    // we need to write some code to reconnect here  
  }  
}
```

Satu lagi, bila ternyata koneksi terputus dengan Cloud (untuk satu alasan tertentu), maka perlu dilakukan rekoneksi. Coding untuk reconnection belum ada, oleh karena itu silahkan disempurnakan. :-)

## Challenge (Optional):

Unduh program DASH-MQTT di handphone Anda. Jadikan sebagai Publisher pada Cloud MQTT tersebut.

Informasi dapat diambil dari sini : <http://blog.magicblocks.io/2017/06/lesson-18-mqtt-dash-mobile-app-with.html>



Bila Switch dari Handphone dipilih melalui icon Lampu, maka Publisher mengirim angka "1" dan menghidupkan LED di ESP32, blinking sebanyak 1 x.

Sesuaikan semua konfigurasi dengan Cloud MQTT yang anda miliki.

