NodeMCU MQTT Iot Project - Switch Button

by osoyooproduct

OSOYOO NodeMCU IOT Starter kit

Please follow us on facebook, find our new released item and share your idea and Video on how to creatively use our products. You can get cash back or giveaway from us!
Facebook: https://www.facebook.com/pg/OsoyooProducts

Youtube:

https://www.youtube.com/channel/UCZsnf-n1TR1MGQT-...

In this lesson, we will connect a switch button to the NodeMCU ,and send the switch status to a MQTT broker. When the button is pressed, NodeMCU will publish the button status “pressed” to MQTT broker and the MQTT client will subscribe to these messages. When the push button is released, “not pressed” will be sent.

https://www.youtube.com/watch?v=OahlZyuC8_o

Step 1: Preparation

Hardware:

NodeMCU board x 1

Switch Button x 1

1K resistor x 1

Breadboard x 1

Jumper wires

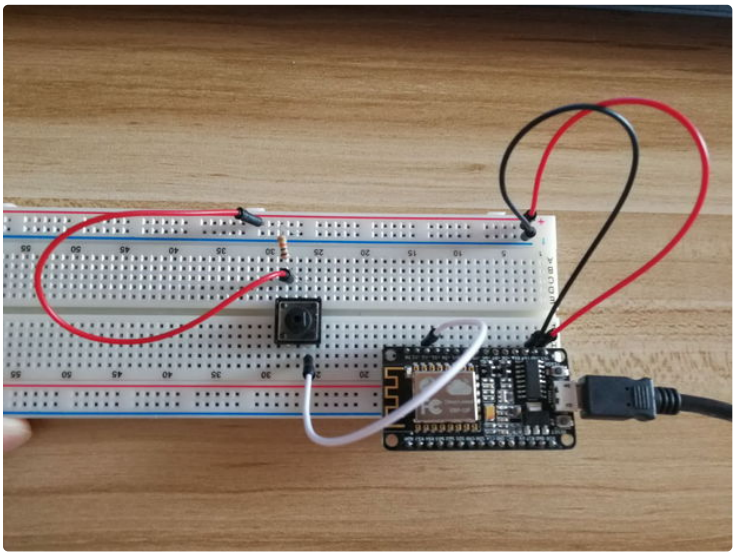
Software:

Arduino IDE(version 1.6.4+)

ESP8266 Board Package and the Serial Port Driver

MQTT Client (MQTTBox here)

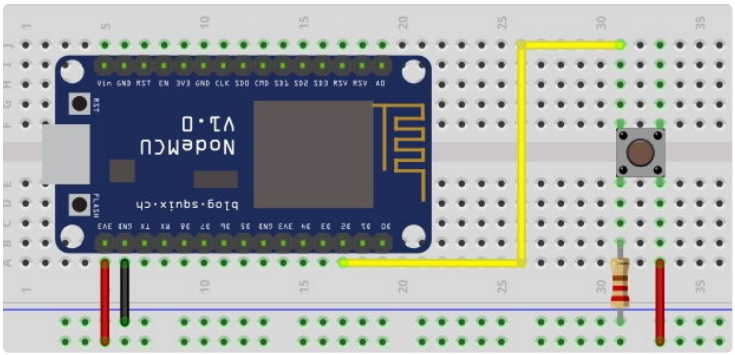
Arduino library: PubSubClient



Step 2: Connection Graph

In this lesson,we use D2(GPIO4) to control the switch,please setup the hardware according the connection graph.

Note: the 1k resistor is using as a pull down resistor, In such a circuit, when the switch is closed, the NodeMCU input is at a logical high value, but when the switch is open, the pull-down resistor pulls the input voltage down to ground (logical zero value), preventing an undefined state at the input.



Step 3: Code

Copy the below code to Arduino IDE:

```
/*  ___   ___  ___  _   _ ___   ___   ____ ___  ____
 * / _ \ /___)/ _ \| | | |/ _ \ / _ \ / ___) _ \|    \
*| |_| |___ | |_| | |_| | |_| | |_| ( (___| |_| | | | |
* \___/(___/ \___/ \__  |\___/ \___(_)____)___/|_|_|_|
*                (____/
*  Use the NodeMCU send switch button status to MQTT client via WiFi
* Tutorial URL:
* CopyRight www.osoyoo.com

*/
#include
#include


 int BUTTON_PIN = D2; //button is connected to GPIO pin D1
// Update these with values suitable for your network.
const char* ssid = "**********";//put your wifi ssid here
const char* password = "**********";//put your wifi password here.
const char* mqtt_server = "broker.mqttdashboard.com";
//const char* mqtt_server = "iot.eclipse.org";

 WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];

 void setup_wifi() {
   delay(100);
 // We start by connecting to a WiFi network
   Serial.print("Connecting to ");
   Serial.println(ssid);
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED)
   {
     delay(500);
     Serial.print(".");
   }
 randomSeed(micros());
```

```
 randomSeed(micros());
 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
}

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

} //end callback

 void reconnect() {
  // Loop until we're reconnected
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    //if you MQTT broker has clientID,username and password
    //please change following line to    if (client.connect(clientId,userName,passWord))
    if (client.connect(clientId.c_str()))
    {
      Serial.println("connected");
     //once connected to MQTT broker, subscribe command if any
      client.subscribe("OsoyooCommand");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
} //end reconnect()

 void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  pinMode(BUTTON_PIN,INPUT);
}

 void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  long now = millis();
  int status;
  //send message every 2 second
  if (now - lastMsg > 2000) {
     lastMsg = now;
     status=digitalRead(BUTTON_PIN);
     String msg="Button status: ";
     if(status==HIGH )
     {
        msg= msg+ "Pressed";
       char message[58];
       msg.toCharArray(message,58);
       Serial.println(message);
       //publish sensor data to MQTT broker
      client.publish("OsoyooData", message);
       }
     else
     {
      msg= msg+ " Not Press";
       char message[58];
       msg.toCharArray(message,58);
       Serial.println(message);
       //publish sensor data to MQTT broker
      client.publish("OsoyooData", message);
     }
```

```
  }

}
```

Edit the code to fit your own WiFi and MQTT settings as following operations:
1)Hotspot Configration:Find below code line,put your own ssid and password on there.

```
const char* ssid = “your_hotspot_ssid”;
const char* password = “your_hotspot_password”;
```

2)MQTT Server Address Setting: You can use your own MQTT broker URL or IP address to set above mqtt_server value. You can also use some famous free MQTT server to test the project such as “broker.mqtt-dashboard.com”, “iot.eclipse.org” etc.

```
const char* mqtt_server = “broker.mqtt-dashboard.com”;
```

3)MQTT Client Settings
If your MQTT broker require clientID,username and password authentication,you need to change

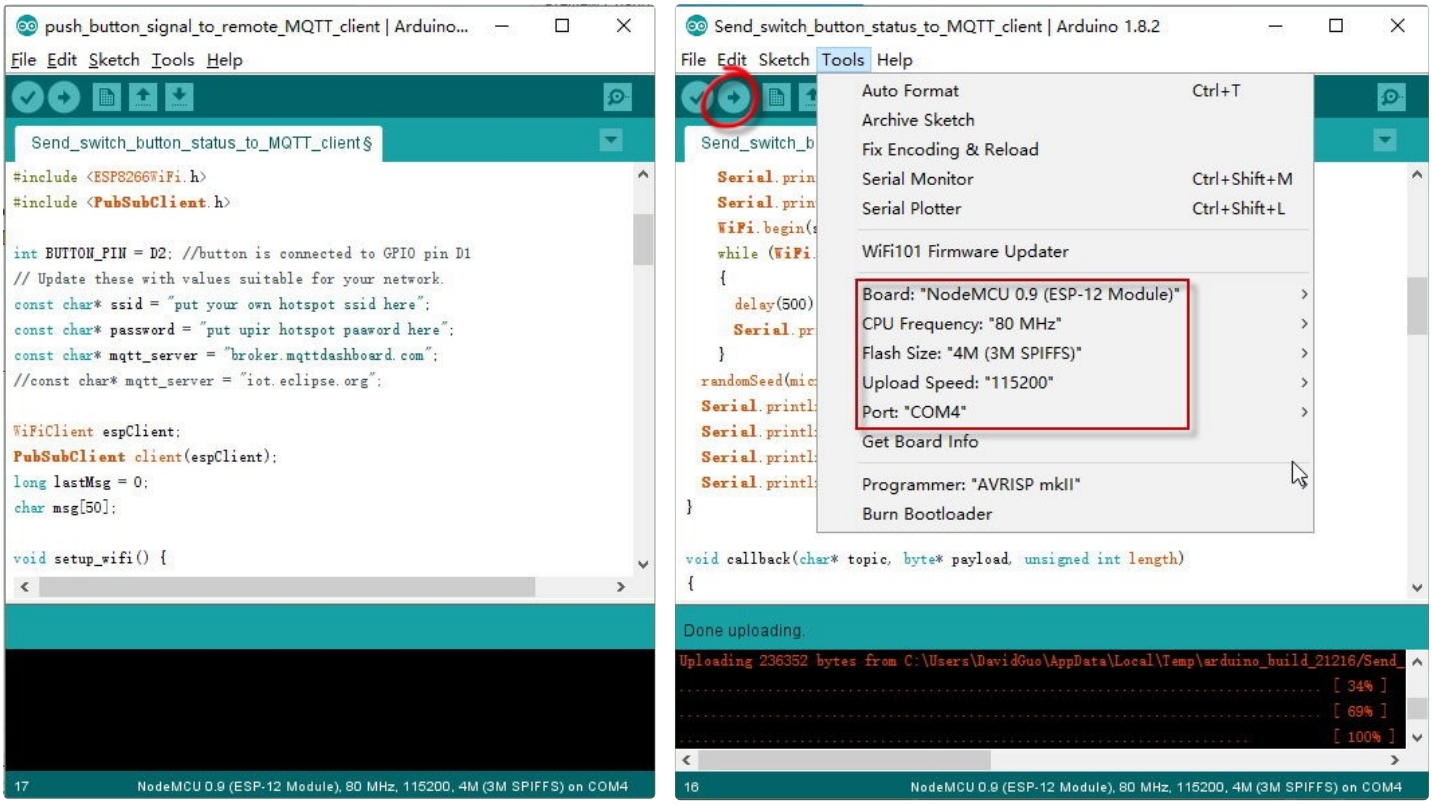
```
if (client.connect(clientId.c_str()))
```

To

```
if (client.connect(clientId,userName,passWord)) //put your clientId/userName/passWord here
```

If not,just keep them as default.
After do that,choose the coresponding board type and port type as below,then upload the sketch to the NodeMCU.

- Board:”NodeMCU 0.9(ESP-12 Module)”
- CPU Frequency:”80MHz”Flash Size:”
- 4M (3M SPIFFS)”
- Upload Speed:”115200″
- Port: Choose your own Serial Port for your NodeMCU



Step 4: MQTT Client Settings

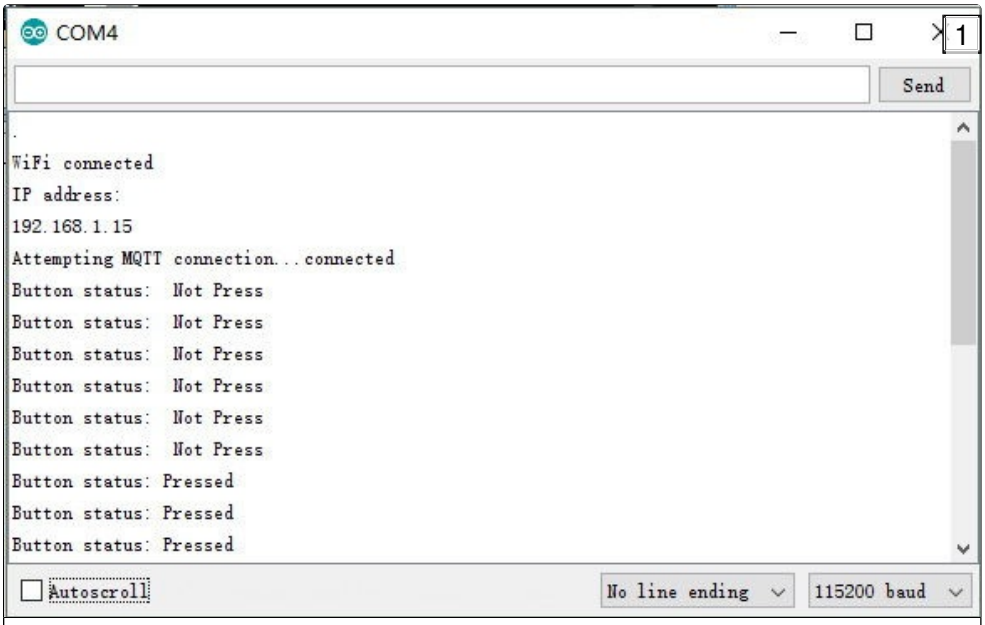
If you don't know how to config MQTT client, please visit our last article: https://www.instructables.com/id/NodeMCU-MQTT-Basi...

Topics Settings:
Topic to publish: OsoyooCommand

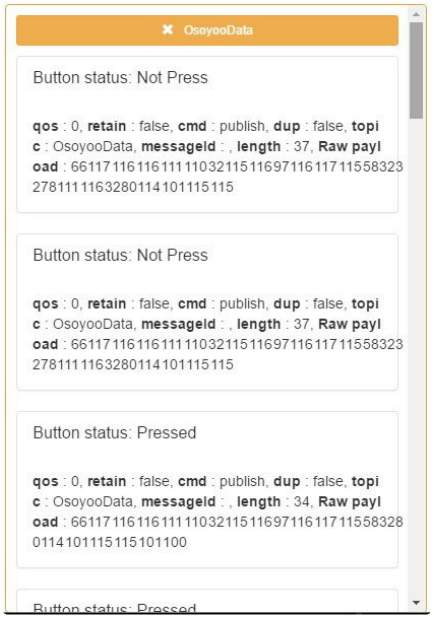
Topic to subscribe: OsoyooData

Running Result

Once the upload done,if wifi hotspot name and password setting is ok and MQTT broker is connected, open the Serial Monitor,you will see following result:Keep pressing this button,the Serial Monitor will output “Button status: Pressed” every 2 second;once release this button,the Serial Monitor will output “Button status: Not Pressed” every 2 second.


COM4
1
Send
.
WiFi connected
IP address:
192.168.1.15
Attempting MQTT connection...connected
Button status: Not Press
Button status: Not Press
Button status: Not Press
Button status: Not Press
Button status: Not Press
Button status: Not Press
Button status: Pressed
Button status: Pressed
Button status: Pressed
Autoscroll
No line ending
115200 baud

1. undefined


OsoyooData
Button status: Not Press
qos : 0, retain : false, cmd : publish, dup : false, topic : OsoyooData, messageId : , length : 37, Raw payload : 66117116116111110321151169711611711558323278111116328011410111 5115
Button status: Not Press
qos : 0, retain : false, cmd : publish, dup : false, topic : OsoyooData, messageId : , length : 37, Raw payload : 66117116116111110321151169711611711558323278111116328011410111 5115
Button status: Pressed
qos : 0, retain : false, cmd : publish, dup : false, topic : OsoyooData, messageId : , length : 34, Raw payload : 66117116116111110321151169711611711558328011410111 5115101100