

```

#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <WiFiUdp.h>
#include <PubSubClient.h>

/* WiFi Settings */
const char* ssid      = "xxxxx";
const char* password = "xxxxx";
const char* topic="/test/sonoff";
const char* topic2="test/sonoffstat";
const char* otahostname="Sonoff01";
const char* clientname="Sonoff01";

/* Sonoff Outputs */
const int relayPin = 12; // Active high
const int ledPin   = 13; // Active low
const int button = 0;
int outstatus = 0;
int laststat = -1;
static char pubstat[4];
long wdttime = 0;
boolean wd1 = false;
boolean wd0 = false;

/* MQTT Settings */
IPAddress broker(192,168,1,201); // Address of the MQTT broker

/**
 * MQTT callback to process messages
 */
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();

  // Examine only the first character of the message
  if(payload[0] == 49) // Message "1" in ASCII (turn outputs ON)
  {
    digitalWrite(ledPin, LOW); // LED is active-low, so this turns it on
    digitalWrite(relayPin, HIGH);
    outstatus=1;
    wd1=false;
  }
}

```

```

    wd0=false;
}
if(payload[0] == 48)          // Message "0" in ASCII (turn outputs OFF)
{
    digitalWrite(ledPin, HIGH);    // LED is active-low, so this turns it off
    digitalWrite(relayPin, LOW);
    outstatus=0;
    wd1=false;
    wd0=false;
}
if(payload[0] == 87)
{
    if(payload[1]==49)
    {
        digitalWrite(ledPin, LOW);    // LED is active-low, so this turns it on
        digitalWrite(relayPin, HIGH);
        outstatus=1;
        wdttime=millis();
        wd1=true;
    }
    if(payload[1]==48)
    {
        digitalWrite(ledPin, HIGH);    // LED is active-low, so this turns it off
        digitalWrite(relayPin, LOW);
        outstatus=0;
        wdttime=millis();
        wd0=true;
    }
}
}
}

```

```
WiFiClient wificlient;
```

```
PubSubClient client(wificlient);
```

```
/**
```

```
* Attempt connection to MQTT broker and subscribe to command topic
```

```
*/
```

```
void reconnect() {
```

```
    // Loop until we're reconnected
```

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

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

```
        // Attempt to connect
```

```
        if (client.connect(clientname)) {
```

```
            Serial.println("connected");
```

```
            client.subscribe(topic);
```

```
        } else {
```

```
            Serial.print("failed, rc=");
```

```
            Serial.print(client.state());
```

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

```

    // Wait 5 seconds before retrying
    delay(5000);
}
}
}

/**
 * Setup
 */
void setup() {
  Serial.begin(115200);
  Serial.println("Booting");
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("WiFi begun");
  while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.println("Connection Failed! Rebooting...");
    delay(5000);
    ESP.restart();
  }
  Serial.println("Proceeding");
  // Port defaults to 8266
  // ArduinoOTA.setPort(8266);

  // Hostname defaults to esp8266-[ChipID]
  ArduinoOTA.setHostname(otahostname);

  // No authentication by default
  //ArduinoOTA.setPassword((const char *)"123");

  ArduinoOTA.onStart([]() {
    Serial.println("Start");
  });
  ArduinoOTA.onEnd([]() {
    Serial.println("\nEnd");
  });
  ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
  });
  ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
  });
  ArduinoOTA.begin();
  Serial.println("Ready");
}

```

```

Serial.print("IP address: ");
Serial.println(WiFi.localIP());

/* Set up the outputs. LED is active-low */
pinMode(ledPin, OUTPUT);
pinMode(relayPin, OUTPUT);
digitalWrite(ledPin, HIGH);
digitalWrite(relayPin, LOW);
pinMode(button, INPUT_PULLUP);
/* Prepare MQTT client */
client.setServer(broker, 1883);
client.setCallback(callback);
}

/**
 * Main
 */
void loop() {
  ArduinoOTA.handle();
  if (WiFi.status() != WL_CONNECTED)
  {
    Serial.print("Connecting to ");
    Serial.print(ssid);
    Serial.println("...");
    WiFi.begin(ssid, password);

    if (WiFi.waitForConnectResult() != WL_CONNECTED)
      return;
    Serial.println("WiFi connected");
  }

  if (WiFi.status() == WL_CONNECTED) {
    if (!client.connected()) {
      reconnect();
    }
  }

  if (client.connected())
  {
    client.loop();

    int pus=digitalRead(button);
    if (pus==LOW) {
      Serial.println("Push");
      if (outstatus==0){
        outstatus=1;
        digitalWrite(ledPin, LOW);      // LED is active-low, so this turns it on
        digitalWrite(relayPin, HIGH);}
      else{

```

```

    digitalWrite(ledPin, HIGH);        // LED is active-low, so this turns it on
    digitalWrite(relayPin, LOW);
    outstatus=0;}
delay(400);
}
if (wd1==true)
{
    if (millis()>wdtime+60000)
    {
        wd1=false;
        digitalWrite(ledPin, HIGH);    // LED is active-low, so this turns it on
        digitalWrite(relayPin, LOW);
        outstatus=0;}
    }
if (wd0==true)
{
    if (millis()>wdtime+60000)
    {
        wd0=false;
        digitalWrite(ledPin, LOW);     // LED is active-low, so this turns it on
        digitalWrite(relayPin, HIGH);
        outstatus=1;}
    }
if (laststat!=outstatus){
    laststat=outstatus;
    if(outstatus==0){
        strcpy(pubstat,"Off");
    }
    if(outstatus==1){
        strcpy(pubstat,"On");
    }
    client.publish(topic2,pubstat);
}
}
}
}
}

```