```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Servo.h>
// Update these with values suitable for your network.
const char* ssid = "....";// put your ssid
const char* password = ".....";// put your password
const char* mqtt_server = "broker.mqttdashboard.com";
//const char* mqtt_server = "iot.eclipse.org";
Servo myservo;  // create servo object to control a servo
WiFiClient espClient;
PubSubClient client(espClient);

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

```
void callback(char* topic, byte* payload, unsigned int length)
{
  String string;
  Serial.print("Command from MQTT broker is : [");
  Serial.print(topic);
  Serial.println("]");
  Serial.println((char)payload[0]);
  int p=(char)payload[0]-'0';
 if(p==1){
  Serial.println("Hello");
  myservo.write(180); //The servo will open on 180 degrees
  delay(5000);      //Wait 5 seconds before it close
  myservo.write(0); //The servo is closed
  }

}//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-";

    // 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("servo");
   } else {
     Serial.print("failed, rc=");
     Serial.print(client.state());
     Serial.println(" try again in 5 seconds");
     // Wait 6 seconds before retrying
     delay(6000);
   }
  }
} //end reconnect()

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  myservo.attach(D1)  // attaches the servo on pin D1 to the servo object
}

void loop() {

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

```
  client.loop();
  client.setCallback(callback);

}
```