```
#include <ESP8266WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"

#define WLAN_SSID       "Workshop"

#define WLAN_PASS       "VIGYAN0030"

#define AIO_SERVER      "io.adafruit.com"

#define AIO_SERVERPORT  1883

#define AIO_USERNAME    "VAGateControl"

#define AIO_KEY         "aio_ZCrs674uVZywEwF2q3JnUL9LINEe"

// Assign output variables to GPIO pins

const int water_pump_in = 13;
const int water_pump_out = 12;

int gate_state = 0;
int timer = 0;



WiFiClient client;     // Create an ESP8266 WiFiClient class to connect to the MQTT server.

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
```

```
Adafruit_MQTT_Subscribe GATE_CONTROL = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/GATE_CONTROL");



void MQTT_connect();

void setup() {

  Serial.begin(115200);

  delay(10);

  pinMode(water_pump_in, OUTPUT);
  pinMode(water_pump_out, OUTPUT);

  digitalWrite(water_pump_in, HIGH);
  digitalWrite(water_pump_out, HIGH);



  Serial.println(); Serial.println();

  Serial.print("Connecting to ");

  Serial.println(WLAN_SSID);



  WiFi.begin(WLAN_SSID, WLAN_PASS);

  while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(500);
    Serial.print(".");
  }
  Serial.println();
 Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
  mqtt.subscribe(&GATE_CONTROL);

}
uint32_t x=0;
void loop() {
   MQTT_connect();
Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(5000))) {
    if (subscription == &GATE_CONTROL) {
      Serial.print(F("Got: "));
```

```
Serial.println((char *)GATE_CONTROL.lastread);

 if (!strcmp((char*) GATE_CONTROL.lastread, "ON") && gate_state == 0)
 {

  for(timer = 0; timer < 11; timer++)
  {
       digitalWrite(water_pump_in, LOW);
       digitalWrite(water_pump_out, HIGH);
       delay (1000);

  }
       digitalWrite(water_pump_in, HIGH);
       delay(10);
       gate_state = 1;
       Serial.println("Gate Open");
       timer = 0;

 }

else if (!strcmp((char*) GATE_CONTROL.lastread, "OFF") && gate_state == 1)
{
 for(timer = 0; timer < 90; timer++)
 {

       digitalWrite(water_pump_in, HIGH);
       digitalWrite(water_pump_out, LOW);
       delay (1000);

}
```

```
            timer = 0;
            digitalWrite(water_pump_out, HIGH);

            delay(6000);
            for(timer = 0; timer < 153; timer++)
            {
              digitalWrite(water_pump_in, LOW);
              delay(1000);
            }

            digitalWrite(water_pump_in, HIGH);
            delay(10);



            gate_state = 0;

            Serial.println("Gate Close");



        }

      }
    }
  }

void MQTT_connect() {

  int8_t ret;
```

```
 // Stop if already connected.

 if (mqtt.connected()) {

   return;

 }

Serial.print("Connecting to MQTT... ");

uint8_t retries = 3;

 while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected

     Serial.println(mqtt.connectErrorString(ret));

     Serial.println("Retrying MQTT connection in 5 seconds...");

     mqtt.disconnect();

     delay(5000);  // wait 5 seconds

     retries--;

     if (retries == 0) {

       // basically die and wait for WDT to reset me

       while (1);
```

```
    }

  }

  Serial.println("MQTT Connected!");

}
```