MQTT Server

**io.adafruit.com**

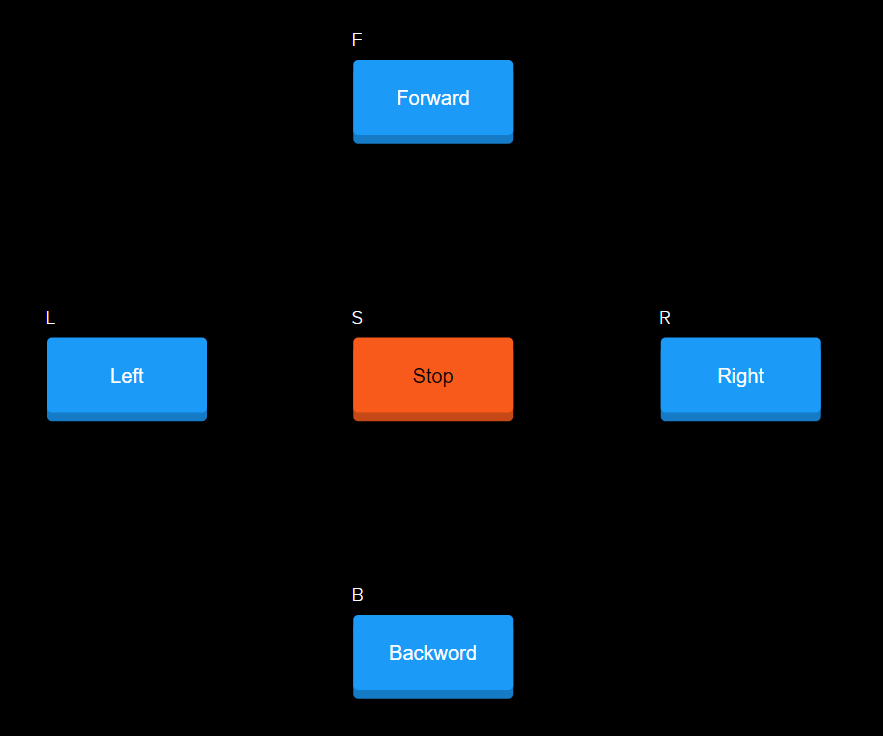


Fig No.8

**Code For This Project-**

#include <ESP8266WiFi.h>

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WiFi Access Point \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define WLAN\_SSID "NagRaj"

#define WLAN\_PASS "7060196036"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Adafruit.io Setup \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define AIO\_SERVER "io.adafruit.com"

#define AIO\_SERVERPORT 1883 // use 8883 for SSL

#define AIO\_USERNAME "NagRaj0308"

#define AIO\_KEY "f6949289e31843668555900441fb4396"

/\*\*\*\*\*\*\*\*\*\*\*\* Global State (you don't need to change this!) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

WiFiClient client;

Adafruit\_MQTT\_Client mqtt(&client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Feeds \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Adafruit\_MQTT\_Subscribe l = Adafruit\_MQTT\_Subscribe(&mqtt, AIO\_USERNAME "/feeds/left");

Adafruit\_MQTT\_Subscribe r = Adafruit\_MQTT\_Subscribe(&mqtt, AIO\_USERNAME "/feeds/right");

Adafruit\_MQTT\_Subscribe f = Adafruit\_MQTT\_Subscribe(&mqtt, AIO\_USERNAME "/feeds/forward");

Adafruit\_MQTT\_Subscribe b = Adafruit\_MQTT\_Subscribe(&mqtt, AIO\_USERNAME "/feeds/backward");

Adafruit\_MQTT\_Subscribe s = Adafruit\_MQTT\_Subscribe(&mqtt, AIO\_USERNAME "/feeds/stop");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sketch Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MQTT\_connect();

void setup() {

Serial.begin(115200);

delay(10);

pinMode(D0,OUTPUT);

pinMode(D1,OUTPUT);

pinMode(D2,OUTPUT);

pinMode(D3,OUTPUT);

pinMode(D5,OUTPUT);

Serial.println(F("Adafruit MQTT demo"));

// Connect to WiFi access point.

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());

// Setup MQTT subscription for onoff feed.

mqtt.subscribe(&l);

mqtt.subscribe(&r);

mqtt.subscribe(&s);

mqtt.subscribe(&f);

mqtt.subscribe(&b);

}

uint32\_t x=0;

void loop() {

// Ensure the connection to the MQTT server is alive (this will make the first

// connection and automatically reconnect when disconnected). See the MQTT\_connect

// function definition further below.

MQTT\_connect();

// this is our 'wait for incoming subscription packets' busy subloop

// try to spend your time here

Adafruit\_MQTT\_Subscribe \*subscription;

while ((subscription = mqtt.readSubscription(5000))) {

if (subscription == &l) {

Serial.println((char \*)l.lastread);

uint16\_t num1 = atoi((char \*)l.lastread);

if ( num1 == 1) {

digitalWrite(D0,HIGH);

digitalWrite(D1,LOW);

digitalWrite(D2,LOW);

digitalWrite(D3,LOW);

digitalWrite(D5,LOW);

delay(500);

digitalWrite(D0,HIGH);

digitalWrite(D1,LOW);

digitalWrite(D2,HIGH);

digitalWrite(D3,LOW);

}}

if (subscription == &r) {

Serial.println((char \*)r.lastread);

uint16\_t num1 = atoi((char \*)r.lastread);

if ( num1 == 1) {

digitalWrite(D5,LOW);

digitalWrite(D0,LOW);

digitalWrite(D1,LOW);

digitalWrite(D2,HIGH);

digitalWrite(D3,LOW);

delay(500);

digitalWrite(D0,HIGH);

digitalWrite(D1,LOW);

digitalWrite(D2,HIGH);

digitalWrite(D3,LOW);

}}

if (subscription == &f) {

Serial.println((char \*)f.lastread);

uint16\_t num1 = atoi((char \*)f.lastread);

if ( num1 == 1) {

digitalWrite(D0,HIGH);

digitalWrite(D1,LOW);

digitalWrite(D2,HIGH);

digitalWrite(D3,LOW);

digitalWrite(D5,LOW);

}}

if (subscription == &b) {

Serial.println((char \*)b.lastread);

uint16\_t num1 = atoi((char \*)b.lastread);

if ( num1 == 1) {

digitalWrite(D0,LOW);

digitalWrite(D1,HIGH);

digitalWrite(D2,LOW);

digitalWrite(D3,HIGH);

digitalWrite(D5,LOW);

}}

if (subscription == &s) {

Serial.println((char \*)s.lastread);

uint16\_t num1 = atoi((char \*)s.lastread);

if ( num1 == 1) {

digitalWrite(D0,LOW);

digitalWrite(D1,LOW);

digitalWrite(D2,LOW);

digitalWrite(D3,LOW);

digitalWrite(D5,HIGH);

}

}

}

}

// Function to connect and reconnect as necessary to the MQTT server.

// Should be called in the loop function and it will take care if connecting.

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!");

}