```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include"Adafruit_MQTT_Client.h"

#define MOOD_RED_PIN           D1
#define MOOD_GREEN_PIN           D2
#define MOOD_BLUE_PIN           D3

#define CENTRE_LIGHT_PIN           D4
#define NIGHT_LIGHT_PIN           D5
#define TELEVISION_PIN           D6
#define COOKER_PIN           D7
#define FAN_PIN           D8
/************************* WIFI Configuration *********************************/

#define WLAN_SSID      "EazyLife"            // Your SSID
#define WLAN_PASS      "eazylife7"       // Your password

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

#define AIO_SERVER     "io.adafruit.com" //Adafruit Server
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "DEAZYLIFE"           // Username
#define AIO_KEY        "aio_EstY17lNwMlRImaeJ0n6gvFw4iWw"   // Auth Key

//WIFI CLIENT
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME,
AIO_KEY);

Adafruit_MQTT_Subscribe FEED_1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/Assistant"); // Feeds name should be same everywhere

char *value;
String message, CENTRE_MESS = "OFF", NIGHT_MESS = "OFF", TV_MESS = "OFF",
COOKER_MESS = "OFF", FAN_MESS = "OFF", MOOD_MESS = "OFF";

int level = 0, i = 0;
bool FASTEST = false, FAST = false, SLOW = false;
unsigned long previousMillis = 0;
const long interval = 500;

void MQTT_connect();

void setup() {
  Serial.begin(115200);

  pinMode(CENTRE_LIGHT_PIN, OUTPUT);
  pinMode(NIGHT_LIGHT_PIN, OUTPUT);
  pinMode(TELEVISION_PIN, OUTPUT);
  pinMode(COOKER_PIN, OUTPUT);
  pinMode(FAN_PIN, OUTPUT);
  pinMode(MOOD_RED_PIN, OUTPUT);
  pinMode(MOOD_GREEN_PIN, OUTPUT);
  pinMode(MOOD_BLUE_PIN, OUTPUT);

  analogWrite(CENTRE_LIGHT_PIN, 0);
  analogWrite(NIGHT_LIGHT_PIN, 0);
  analogWrite(TELEVISION_PIN, 0);
  analogWrite(COOKER_PIN, 0);
  analogWrite(FAN_PIN, 0);
  analogWrite(MOOD_RED_PIN, 0);
  analogWrite(MOOD_GREEN_PIN, 0);
  analogWrite(MOOD_BLUE_PIN, 0);

  // 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());

  mqtt.subscribe(&FEED_1);
}

void loop() {

  /********* START LOOP *********/

  MQTT_connect();

  /**************************************** CHECK ALL SUSCRIPTIONS
```

```
*******************************/

 Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(500))) {

    /******************** CHECK FEED1 ***********************/

    if (subscription == &FEED_1) {

      /******** GET MESSAGE OVER ********/

      value = (char *)FEED_1.lastread;
      message = String(value);
      message.trim();
      message.toUpperCase();

      Serial.print(F("Got: "));
      Serial.println(message);

      /********* GET MESSAGE OVER *********/

      /********* FUNCTION ON MESSAGE *********/

      // CENTRE LIGHT FUNCTION

      if (message.indexOf("CENTRE") > -1) {
        Serial.print("CENTRE LIGHT ");
        if (message.indexOf("MOOD") > -1 || message.indexOf("MODE") > -1) {
          CENTRE_MESS = "MOOD";
        }
        else if (message.indexOf("ON") > -1) {
          CENTRE_MESS = "ON";
        }
        if (message.indexOf("OFF") > -1) {
          CENTRE_MESS = "OFF";
        }
        Serial.println(CENTRE_MESS);
      }


      // NIGHT LIGHT FUNCTION

      if (message.indexOf("NIGHT") > -1) {
        Serial.print("NIGHT LIGHT ");
        if (message.indexOf("MOOD") > -1 || message.indexOf("MODE") > -1) {
          NIGHT_MESS = "MOOD";
        }
        else if (message.indexOf("ON") > -1) {
          NIGHT_MESS = "ON";
        }
        if (message.indexOf("OFF") > -1) {
          NIGHT_MESS = "OFF";
        }
        Serial.println(NIGHT_MESS);
      }


      // TELEVISION OR HOME THEATRE FUNCTION

      if (message.indexOf("TELEVISION") > -1 || message.indexOf("TV") > -1 ||
message.indexOf("THEATRE") > -1 || message.indexOf("DVD") > -1) {
        Serial.print("TELEVISION OR HOME THEATRE ");
        if (message.indexOf("ON") > -1) {
          TV_MESS = "ON";
        }
        if (message.indexOf("OFF") > -1) {
          TV_MESS = "OFF";
        }
        Serial.println(TV_MESS);
      }


      // COOKER FUNCTION

      if (message.indexOf("COOKER") > -1) {
        Serial.print("COOKER UNIT ");
        if (message.indexOf("ON") > -1) {
          COOKER_MESS = "ON";
        }
        if (message.indexOf("OFF") > -1) {
          COOKER_MESS = "OFF";
        }
        Serial.println(COOKER_MESS);
      }


      // FAN OR AC FUNCTION

      if (message.indexOf("FAN") > -1) {
        Serial.print("FAN OR AC ");
        if (message.indexOf("HIGH") > -1 || message.indexOf("MAX") > -1 || (message.
indexOf("LEVEL") > -1 && message.indexOf("3") > -1)) {
```

```
          FAN_MESS = "MAX";
        }
       else if (message.indexOf("MID") > -1 || message.indexOf("AVERAGE") > -1 ||
(message.indexOf("LEVEL") > -1 && message.indexOf("2") > -1)) {
          FAN_MESS = "MID";
        }
       else if (message.indexOf("ON") > -1 || message.indexOf("LOW") > -1 ||
(message.indexOf("LEVEL") > -1 && message.indexOf("1") > -1)) {
          FAN_MESS = "LOW";
        }
        if (message.indexOf("OFF") > -1) {
          FAN_MESS = "OFF";
        }
        Serial.println(FAN_MESS);
      }


      // MOOD LIGHT FUNCTION

     if ((message.indexOf("MOOD") > -1 || message.indexOf("MODE") > -1) && (message.
indexOf("CENTRE") < 0 && message.indexOf("NIGHT") < 0)) {
        Serial.print("MOOD LIGHT ");
        if (message.indexOf("RED") > -1) {
          MOOD_MESS = "RED";
          FASTEST = false;
          FAST = false;
          SLOW = false;
        }
        else if (message.indexOf("GREEN") > -1) {
          MOOD_MESS = "GREEN";
          FASTEST = false;
          FAST = false;
          SLOW = false;
        }
        else if (message.indexOf("BLUE") > -1) {
          MOOD_MESS = "BLUE";
          FASTEST = false;
          FAST = false;
          SLOW = false;
        }
        else if (message.indexOf("YELLOW") > -1) {
          MOOD_MESS = "YELLOW";
          FASTEST = false;
          FAST = false;
          SLOW = false;
        }
        else if (message.indexOf("PINK") > -1) {
          MOOD_MESS = "PINK";
          FASTEST = false;
          FAST = false;
          SLOW = false;
         }
        else if (message.indexOf("ORANGE") > -1) {
          MOOD_MESS = "ORANGE";
          FASTEST = false;
          FAST = false;
          SLOW = false;
         }
        else if (message.indexOf("WHITE") > -1) {
          MOOD_MESS = "WHITE";
          FASTEST = false;
          FAST = false;
          SLOW = false;
         }
        else if (message.indexOf("VIOLET") > -1 || message.indexOf("VIOLENT") > -1
 || message.indexOf("PURPLE") > -1) {
          MOOD_MESS = "VIOLET";
          FASTEST = false;
          FAST = false;
          SLOW = false;
         }
        else if (message.indexOf("FASTEST") > -1 && message.indexOf("RGB") > -1) {
          MOOD_MESS = "RGB FASTEST";
          FASTEST = true;
          FAST = false;
          SLOW = false;
         }
        else if (message.indexOf("FAST") > -1 && message.indexOf("RGB") > -1) {
          MOOD_MESS = "RGB FAST";
          FASTEST = false;
          FAST = true;
          SLOW = false;
         }
        else if (message.indexOf("ON") > -1 || message.indexOf("SLOW") > -1 ||
 message.indexOf("RGB") > -1) {
          MOOD_MESS = "RGB SLOW";
          FASTEST = false;
          FAST = false;
          SLOW = true;
         }
        if (message.indexOf("OFF") > -1 || message.indexOf("BLACK") > -1 || message.
 indexOf("BLANK") > -1) {
         MOOD_MESS = "OFF";
         FASTEST = false;
         FAST = false;
         SLOW = false;
       }
      Serial.println(MOOD_MESS);
     }

     /********* FUNCTION ON MESSAGE OVER *********/

    }

   /******************** CHECK FEED1 OVER ***********************/

  }

  /***************************************** CHECK ALL SUSCRIPTIONS OVER
 **********************************/



  /********* CONTINUE LOOP *********/
  //DO ALL OTHER LOOP HERE

  if (CENTRE_MESS == "MOOD") {
   analogWrite(CENTRE_LIGHT_PIN, 1023);
  }
  else if (CENTRE_MESS == "OFF") {
   analogWrite(CENTRE_LIGHT_PIN, 0);
  }
  else if (CENTRE_MESS == "ON") {
   analogWrite(CENTRE_LIGHT_PIN, 1023);
  }



  if (NIGHT_MESS == "MOOD") {
   analogWrite(NIGHT_LIGHT_PIN, 1023);
  }
  else if (NIGHT_MESS == "OFF") {
   analogWrite(NIGHT_LIGHT_PIN, 0);
  }
  else if (NIGHT_MESS == "ON") {
   analogWrite(NIGHT_LIGHT_PIN, 1023);
  }



  if (TV_MESS == "OFF") {
```

```
 analogWrite(TELEVISION_PIN, 0);
}
else if (TV_MESS == "ON") {
 analogWrite(TELEVISION_PIN, 1023);
}


if (COOKER_MESS == "OFF") {
 analogWrite(COOKER_PIN, 0);
}
else if (COOKER_MESS == "ON") {
 analogWrite(COOKER_PIN, 1023);
}


if (FAN_MESS == "OFF") {
 analogWrite(FAN_PIN, 0);
}
else if (FAN_MESS == "LOW") {
 analogWrite(FAN_PIN, 341);
}
else if (FAN_MESS == "MID") {
 analogWrite(FAN_PIN, 682);
}
else if (FAN_MESS == "HIGH") {
 analogWrite(FAN_PIN, 1023);
}


if (MOOD_MESS == "OFF") {
 analogWrite(MOOD_RED_PIN, 0);
 analogWrite(MOOD_GREEN_PIN, 0);
 analogWrite(MOOD_BLUE_PIN, 0);
}
else if (MOOD_MESS == "RED") {
 analogWrite(MOOD_RED_PIN, 1023);
 analogWrite(MOOD_GREEN_PIN, 0);
 analogWrite(MOOD_BLUE_PIN, 0);
}
else if (MOOD_MESS == "GREEN") {
 analogWrite(MOOD_RED_PIN, 0);
 analogWrite(MOOD_GREEN_PIN, 1023);
 analogWrite(MOOD_BLUE_PIN, 0);
}
else if (MOOD_MESS == "BLUE") {
 analogWrite(MOOD_RED_PIN, 0);
```

```
    analogWrite(MOOD_GREEN_PIN, 0);
    analogWrite(MOOD_BLUE_PIN, 1023);
  }
  else if (MOOD_MESS == "YELLOW") {
    analogWrite(MOOD_RED_PIN, 0);
    analogWrite(MOOD_GREEN_PIN, 1023);
    analogWrite(MOOD_BLUE_PIN, 1023);
  }
  else if (MOOD_MESS == "PINK") {
    analogWrite(MOOD_RED_PIN, 1013);
    analogWrite(MOOD_GREEN_PIN, 297);
    analogWrite(MOOD_BLUE_PIN, 621);
  }
  else if (MOOD_MESS == "ORANGE") {
    analogWrite(MOOD_RED_PIN, 1023);
    analogWrite(MOOD_GREEN_PIN, 1023);
    analogWrite(MOOD_BLUE_PIN, 0);
  }
  else if (MOOD_MESS == "WHITE") {
    analogWrite(MOOD_RED_PIN, 1023);
    analogWrite(MOOD_GREEN_PIN, 1023);
    analogWrite(MOOD_BLUE_PIN, 1023);
  }
  else if (MOOD_MESS == "VIOLET") {
    analogWrite(MOOD_RED_PIN, 1023);
    analogWrite(MOOD_GREEN_PIN, 0);
    analogWrite(MOOD_BLUE_PIN, 1023);
  }
  else if (MOOD_MESS == "RGB SLOW" && SLOW == true) {
    delay(500);
    R_G_B();
  }
  else if (MOOD_MESS == "RGB FAST" && FAST == true) {
    delay(250);
    R_G_B();
  }
  else if (MOOD_MESS == "RGB FAST" && FASTEST == true) {
    delay(10);
    R_G_B();
  }

  /********* END LOOP *********/

}

void MQTT_connect() {
  int8_t ret;

  if (mqtt.connected()) {
    return;
  }

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

  uint8_t retries = 3;

  while ((ret = mqtt.connect()) != 0) {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);
    retries--;
    if (retries == 0) {
      while (1);
    }
  }
  Serial.println("MQTT Connected!");
}

void R_G_B() {
  switch (level) {
    case 0:
      analogWrite(MOOD_RED_PIN, 1023);
      analogWrite(MOOD_GREEN_PIN, i);
      analogWrite(MOOD_BLUE_PIN, i);
      i--;
      if (i <= 0) {
        i = 0;
        level = 1;
      }
      break;
    case 1:
      analogWrite(MOOD_RED_PIN, 1023);
      analogWrite(MOOD_GREEN_PIN, i);
      analogWrite(MOOD_BLUE_PIN, 0);
      i++;
      if (i >= 1024) {
        i = 1023;
        level = 2;
      }
      break;
    case 2:
```

```
   analogWrite(MOOD_RED_PIN, i);
   analogWrite(MOOD_GREEN_PIN, 1023);
   analogWrite(MOOD_BLUE_PIN, 0);
    i--;
    if (i <= 0) {
      i = 0;
      level = 3;
    }
    break;
 case 3:
   analogWrite(MOOD_RED_PIN, 0);
   analogWrite(MOOD_GREEN_PIN, 1023);
   analogWrite(MOOD_BLUE_PIN, i);
    i++;
    if (i >= 1024) {
      i = 1023;
      level = 4;
    }
    break;
 case 4:
   analogWrite(MOOD_RED_PIN, 0);
   analogWrite(MOOD_GREEN_PIN, i);
   analogWrite(MOOD_BLUE_PIN, 1023);
    i--;
    if (i <= 0) {
      i = 0;
      level = 5;
    }
    break;
 case 5:
   analogWrite(MOOD_RED_PIN, i);
   analogWrite(MOOD_GREEN_PIN, 0);
   analogWrite(MOOD_BLUE_PIN, 1023);
    i++;
    if (i >= 1024) {
      i = 1023;
      level = 6;
    }
    break;
 case 6:
   analogWrite(MOOD_RED_PIN, 1023);
   analogWrite(MOOD_GREEN_PIN, 0);
   analogWrite(MOOD_BLUE_PIN, i);
    i--;
    if (i <= 0) {
      i = 0;
```

```
        level = 7;
      }
      break;
    case 7:
      analogWrite(MOOD_RED_PIN, 1023);
      analogWrite(MOOD_GREEN_PIN, i);
      analogWrite(MOOD_BLUE_PIN, i);
      if (i >= 1024) {
        i = 1023;
        level = 0;
      }
      break;
  }
}
```