

LISTING PROGRAM

NODEMCU

```
#include <WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"

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

#define WLAN_SSID "CV-COMETRONICA 4G"

#define WLAN_PASS "cometronica2021"

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

#define AIO_SERVER "io.adafruit.com"

#define AIO_SERVERPORT 1883 // use 8883 for SSL

#define AIO_USERNAME "dinatriazara"

#define AIO_KEY "aio_aneb49NACvvAjQTIwZ7EX3v7KHod"

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

WiFiClient client;

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

Adafruit_MQTT_Subscribe iot_motor1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/motor1");

Adafruit_MQTT_Subscribe iot_motor2 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/motor2");

Adafruit_MQTT_Subscribe iot_motor3 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/motor3");

Adafruit_MQTT_Subscribe iot_motor4 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/motor4");
```

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

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

```
Adafruit_MQTT_Publish    online    =    Adafruit_MQTT_Publish(&mqtt,  
AIO_USERNAME "/feeds/online");
```

```
void MQTT_connect();
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
#define LED_GPIO  26
```

```
#define PWM1_Ch   1
```

```
#define PWM1_Res  8
```

```
#define PWM1_Freq 1000
```

```
#define LED_GPIO2 27
```

```
#define PWM2_Ch   2
```

```
#define PWM2_Res  8
```

```
#define PWM2_Freq 1000
```

```
#define LED_GPIO3 25
```

```
#define PWM3_Ch   3
```

```
#define PWM3_Res  8
```

```
#define PWM3_Freq 1000

#define LED_GPIO4 14

#define PWM4_Ch 4

#define PWM4_Res 8

#define PWM4_Freq 1000

#define LED_GPIO5 12

#define PWM5_Ch 5

#define PWM5_Res 8

#define PWM5_Freq 1000

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)

#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it

#endif

BluetoothSerial SerialBT;

char dataBT ;

String data_waktu, data_motor1, data_reciver, data_motor2, data_motor3,
data_motor4, data_arah;

int motor1, motor2, motor3, motor4, waktu;

int arah = 0;

unsigned long previousMillis = 0; // will store last time LED was updated
```

```
const long interval = 1000;

int count;

void setup() {
  lcd.begin();
  lcd.backlight();
  lcd.print("Mulai");

  pinMode(4, INPUT_PULLUP);
  pinMode(15, INPUT_PULLUP);
  pinMode(18, INPUT_PULLUP);
  pinMode(19, INPUT_PULLUP);

  Serial.begin(115200);
  SerialBT.begin("Bluetooth Kerupuk"); //Bluetooth device name

  ledcAttachPin(LED_GPIO, PWM1_Ch);
  ledcSetup(PWM1_Ch, PWM1_Freq, PWM1_Res);
  ledcAttachPin(LED_GPIO2, PWM2_Ch);
  ledcSetup(PWM2_Ch, PWM2_Freq, PWM2_Res);
  ledcAttachPin(LED_GPIO3, PWM3_Ch);
  ledcSetup(PWM3_Ch, PWM3_Freq, PWM3_Res);
  ledcAttachPin(LED_GPIO4, PWM4_Ch);
  ledcSetup(PWM4_Ch, PWM4_Freq, PWM4_Res);
  ledcAttachPin(LED_GPIO5, PWM5_Ch);
  ledcSetup(PWM5_Ch, PWM5_Freq, PWM5_Res);
```

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

Serial.println("The device started, now you can pair it with bluetooth!");

Serial.print("Connecting to ");

Serial.println(WLAN_SSID);

WiFi.begin(WLAN_SSID, WLAN_PASS);

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

  lcd.print(".");

  Serial.print(".");

  delay(200);

}

lcd.clear();

Serial.println();

Serial.println("WiFi connected");

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

mqtt.subscribe(&iot_motor1);

mqtt.subscribe(&iot_motor2);

mqtt.subscribe(&iot_motor3);

mqtt.subscribe(&iot_motor4);

mqtt.subscribe(&iot_arah);

mqtt.subscribe(&iot_waktu);

delay(10);

}

String mode = "BLU";
```

```

uint32_t x = 0;

bool reset_iot = 0;

void loop() {

    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {

        previousMillis = currentMillis;

        tampil();

        if (waktu > 0) {

            waktu--;

        }

    }

    mode_input();

    if (digitalRead(15) == LOW) { // set motor

        if (digitalRead(4) == LOW) { // naik turun set motor

            set_motor(1);

        }

        else {

            set_motor(0);

        }

    }

    else { // kondisi mode jalan

        if (mode == "MAN") { // mode manual

            if (digitalRead(15) == LOW && digitalRead(18) == LOW) {

                gerak(1, 1, 0, 50, 0, 0);

            }

        }

    }

}

```

```
    }  
    else {  
        gerak(1, 1, 110, 0, 80, 80);  
    }  
}  
else {  
    if (mode == "BLU") {  
        baca_bluethooth();  
        gerak(1, arah, motor1, motor2, motor3, motor4);  
    }  
    else if (mode == "IOT") {  
        baca_iot();  
        gerak(1, arah, motor1, motor2, motor3, motor4);  
    }  
}  
}  
}
```

DATA

```
void gerak(bool kondisi, bool arah, int s1, int s2, int s3, int s4) {  
    if (kondisi == 1) {  
        ledcWrite(PWM1_Ch, s1);  
        ledcWrite(PWM2_Ch, s2);  
        ledcWrite(PWM3_Ch, s3);  
        if (arah == 0) {  
            ledcWrite(PWM4_Ch, s4);  
            ledcWrite(PWM5_Ch, 0);  
        }  
        else if (arah == 1) {  
            ledcWrite(PWM4_Ch, 0);  
            ledcWrite(PWM5_Ch, s4);  
        }  
    }  
    else {  
        ledcWrite(PWM1_Ch, 0);  
        ledcWrite(PWM2_Ch, 0);  
        ledcWrite(PWM3_Ch, 0);  
        ledcWrite(PWM4_Ch, 0);  
        ledcWrite(PWM5_Ch, 0);  
    }  
}
```



```

void set_motor(bool kondisi) {
    if (kondisi == 1) {
        Serial.println("Set motor Naik");
        ledcWrite(PWM1_Ch, 0);
        ledcWrite(PWM2_Ch, 0);
        ledcWrite(PWM3_Ch, 0);
        ledcWrite(PWM4_Ch, 100);
        ledcWrite(PWM5_Ch, 0);
    }
    else {
        Serial.println("Set motor turun");
        ledcWrite(PWM1_Ch, 0);
        ledcWrite(PWM2_Ch, 0);
        ledcWrite(PWM3_Ch, 0);
        ledcWrite(PWM4_Ch, 0);
        ledcWrite(PWM5_Ch, 100);
    }
}

void mode_input() {
    if (digitalRead(18) == HIGH) { // jika tombol tidak di tekan
        if (digitalRead(19) == LOW) { // jika saklar ke atas
            mode = "BLU";
            reset_iot = 1;
        }
    }
}

```

```

else { // saklar ke bawah
    if (reset_iot == 1) {
        ESP.restart();
    }
    mode = "IOT";
}
}

else { // mode tombol
    mode = "MAN";
}
}

void baca_iot() {
    MQTT_connect();
    Adafruit_MQTT_Subscribe *subscription;
    while ((subscription = mqtt.readSubscription(1000))) {
        if (subscription == &iot_motor1) {
            Serial.print(F("motor1: "));
            Serial.println((char *)iot_motor1.lastread);
            data_motor1 = (char *)iot_motor1.lastread;
            motor1 = data_motor1.toInt();
        }
        if (subscription == &iot_motor2) {
            Serial.print(F("motor2: "));
            Serial.println((char *)iot_motor2.lastread);

```

```
data_motor2 = (char *)iot_motor2.lastread;
motor2 = data_motor2.toInt();
}
if (subscription == &iot_motor3) {
    Serial.print(F("motor3: "));
    Serial.println((char *)iot_motor3.lastread);
    data_motor3 = (char *)iot_motor3.lastread;
    motor3 = data_motor3.toInt();
}
if (subscription == &iot_motor4) {
    Serial.print(F("motor4: "));
    Serial.println((char *)iot_motor4.lastread);
    data_motor4 = (char *)iot_motor4.lastread;
    motor4 = data_motor4.toInt();
}
if (subscription == &iot_arah) {
    Serial.print(F("arah: "));
    Serial.println((char *)iot_arah.lastread);
    data_arah = (char *)iot_arah.lastread;
    arah = data_arah.toInt();
}
if (subscription == &iot_waktu) {
    Serial.print(F("waktu: "));
    Serial.println((char *)iot_waktu.lastread);
    data_waktu = (char *)iot_waktu.lastread;
```

```
waktu = data_waktu.toInt();

motor1 = 110;

motor2 = 0;

motor3 = 80;

motor4 = 80;

arah = 1;

}

}

}

void baca_bluetooth() {
  if (SerialBT.available()) {
    dataBT = SerialBT.read();
    if (dataBT == 'a') {
      data_motor1 = data_reciver;
      data_reciver = "";
      motor1 = data_motor1.toInt();
    }
    else if (dataBT == 'b') {
      data_motor2 = data_reciver;
      data_reciver = "";
      motor2 = data_motor2.toInt();
    }
    else if (dataBT == 'c') {
```

```
data_motor3 = data_reciver;

data_reciver = "";

motor3 = data_motor3.toInt();

}

else if (dataBT == 'd') {

data_motor4 = data_reciver;

data_reciver = "";

motor4 = data_motor4.toInt();

}

else if (dataBT == 'e') {

data_arah = data_reciver;

data_reciver = "";

arah = data_arah.toInt();

}

else if (dataBT == 'f') {

data_waktu = data_reciver;

data_reciver = "";

waktu = data_waktu.toInt();

motor1 = 110;

motor2 = 0;

motor3 = 80;

motor4 = 80;

arah = 1;

}

else if (dataBT == 'R') {
```

```
        ESP.restart();
    }
    else {
        data_reciver += dataBT;
    }
    Serial.println(dataBT);
}
}

void tampil() {
    Serial.print("t1: ");
    Serial.print(digitalRead(15));
    Serial.print(" || t2: ");
    Serial.print(digitalRead(4));
    Serial.print(" || t3: ");
    Serial.print(digitalRead(18));
    Serial.print(" || t4: ");
    Serial.print(digitalRead(19));
    Serial.print(" || mode = ");
    Serial.print(mode);
    Serial.print(" || m1 = ");
    Serial.print(motor1);
    Serial.print(" || m2 = ");
    Serial.print(motor2);
    Serial.print(" || m3 = ");
```

```
Serial.print(motor3);  
Serial.print(" || m4 = ");  
Serial.print(motor4);  
Serial.print(" || A = ");  
Serial.print(arah);  
Serial.print(" || W = ");  
Serial.print(waktu);  
Serial.println();  
lcd.home();  
lcd.print("M:");  
lcd.print(mode);  
lcd.print(" ");  
lcd.setCursor(6, 0);  
lcd.print("1:");  
lcd.print(motor1);  
lcd.setCursor(11, 0);  
lcd.print("A:");  
lcd.print(waktu);  
lcd.setCursor(0, 1);  
lcd.print("2:");  
lcd.print(motor2);  
lcd.setCursor(6, 1);  
lcd.print("3:");  
lcd.print(motor3);  
lcd.setCursor(11, 1);
```

```

    lcd.print("4:");
    lcd.print(motor4);
}

void MQTT_connect() {
    int8_t ret;
    if (mqtt.connected()) {
        return;
    }
    Serial.print("Connecting to MQTT... ");
    lcd.clear();
    lcd.print("Connect to MQ");
    uint8_t retries = 5;
    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();
        for (int i = 0; i <= 10; i++) {
            Serial.println(".");
            delay(500);
        }
        retries--;
        if (retries == 0) {
            while (1);
        }
    }
}

```



```
}  
lcd.clear();  
Serial.println("MQTT Connected!");  
}
```