LAMPIRAN-1

LISTING PROGRAM

```
int trig1Pin = D7;
int echo1Pin = D8;
void setup() {

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  pinMode(trig1Pin, OUTPUT);
  pinMode(echo1Pin, INPUT);
  Serial.begin(115200);
  }

void loop(){
     Product1();
     Product2();
  }

void Product1(){
delay(500);// reading will be taken after ....miliseconds
int duration, distance;
    digitalWrite (trigPin, HIGH);
    delayMicroseconds (10);
    digitalWrite (trigPin, LOW);
    duration = pulseIn (echoPin, HIGH);
    distance = (duration/2) / 29.1;
  Serial.println("Hasil Sensor 1 :");
      Serial.print( distance);
     Serial.println("cm     |     ");
      }

void Product2(){
delay(500);// reading will be taken after ....miliseconds
int duration, distance;
    digitalWrite (trig1Pin, HIGH);
    delayMicroseconds (10);
    digitalWrite (trig1Pin, LOW);
    duration = pulseIn (echo1Pin, HIGH);
    distance = (duration/2) / 29.1;
   Serial.println("Hasil Sensor 2 :");
      Serial.print( distance);
```

```
  Serial.println("cm     |     ");
}
```

```
//Gas Sensor Pins

#define MQ135 14
#define F2602 15
#define F2600 A3

//Gas Sensor Load Resistance (RL)
#define RL_F2600 7.5
#define RL_MQ135 10
#define RL_F2602 7.5

/*2600 GASSES*/
float CO_A = 1.144997421;
float CO_B = -0.21687423;

float CH4_A = 1.05777824;
float CH4_B = -0.0795645;

/*2602 GASSES*/
float NH3_A = 0.92372844;
float NH3_B = -0.291578925;

float H2S_A = 0.38881036;
float H2S_B = -0.35010059;

float VOC_A = 0.3220722;
float VOC_B = -0.6007520;


/*MQ-135 GASSES*/
//MQ-135 CO2
float A_MQ135_CO2 = 112.89808;
float B_MQ135_CO2 = -2.868463517;


//MQ-135 NOx
float A_MQ135_NOx = 34.69756084;
float B_MQ135_NOx = -3.422829698;

void setup()
{
Serial.begin(9600); // opens serial port, sets data rate 9600 bps
}
```

```
void loop()
{
 //Refer to the documentation mentioned on this page to understand this better

 float VRL_F2600;
 float Rs_F2600;
 float Ro_F2600 = 11.26;
 float ratio_F2600;

 VRL_F2600 = analogRead(F2600)*(5.0/1023.0);
 Rs_F2600 = ((5.0/VRL_F2600)-1)*(RL_F2600);
 ratio_F2600 = Rs_F2600/Ro_F2600;


 float ppm_CO = CO_A * pow(ratio_F2600, CO_B);
 float ppm_CH4 = CH4_A * pow(ratio_F2600, CH4_B);

 float VRL_MQ135;
 float Rs_MQ135;
 float Ro_MQ135 = 20.1;
 float ratio_MQ135;

 VRL_MQ135 = analogRead(MQ135)*(5.0/1023.0);
 Rs_MQ135 = ((5.0/VRL_MQ135)-1)*(RL_MQ135);
 ratio_MQ135 = Rs_MQ135/Ro_MQ135;


 float ppm_CO2 = A_MQ135_CO2 * pow(ratio_MQ135, B_MQ135_CO2);
 float ppm_NOx = A_MQ135_NOx * pow(ratio_MQ135, B_MQ135_NOx);

 float VRL_F2602;
 float Rs_F2602;
 float Ro_F2602 = 64.88;
 float ratio_F2602;

 VRL_F2602 = analogRead(F2602)*(5.0/1023.0);
 Rs_F2602 = ((5.0/VRL_F2602)-1)*(RL_F2602);
 ratio_F2602 = Rs_F2602/Ro_F2602;

 float ppm_VOC = VOC_A * pow(ratio_F2602, VOC_B);
 float ppm_H2S = H2S_A * pow(ratio_F2602, H2S_B);
 float ppm_NH3 = NH3_A * pow(ratio_F2602, NH3_B);

Serial.print("NH3: ");
```

```
 Serial.println(ppm_NH3);
 Serial.print("CO2: ");
 Serial.println(ppm_CO2);
 Serial.print("VOC : ");
 Serial.println(ppm_CO );
 delay(500);
}
```

```
//Include the library
#include <MQUnifiedsensor.h>

//Definitions
#define placa "Arduino UNO"
#define Voltage_Resolution 5
#define pin 33 //Analog input 0 of your arduino
#define type "MQ-135" //MQ135
#define ADC_Bit_Resolution 10 // For arduino UNO/MEGA/NANO
#define RatioMQ135CleanAir 3.6//RS / R0 = 3.6 ppm
//#define calibration_button 13 //Pin to calibrate your sensor

//Declare Sensor
MQUnifiedsensor MQ135(placa, Voltage_Resolution, ADC_Bit_Resolution, pin,
type);

char jenisgas[6][10] = {"CO","Alcohol","CO2","Tolueno","NH4","Aceton"};
float gasA[6] = {605.18, 77.255, 110.47, 44.947, 102.2, 34.668};
float gasB[6] = {-3.937, -3.18, -2.862, -3.445, -2.473};
int itemcheck = 4;

void setup() {
  //Init the serial port communication - to debug the library
  Serial.begin(9600); //Init serial port

  //Set math model to calculate the PPM concentration and the value of constants
  MQ135.setRegressionMethod(1); //_PPM =  a*ratio^b
  MQ135.setA(gasA[itemcheck]); MQ135.setB(gasB[itemcheck]); // Configurate
the ecuation values to get CO2 concentration
  //MQ135.setA(102.2); MQ135.setB(-2.473); // Configurate the ecuation values
to get NH4 concentration

  /*
    Exponential regression:
  GAS      | a      | b
  CO       | 605.18 | -3.937
  Alcohol  | 77.255 | -3.18
  CO2      | 110.47 | -2.862
  Tolueno  | 44.947 | -3.445
  NH4      | 102.2  | -2.473
  Acetona  | 34.668 | -3.369
  */

  /*****************************  MQ Init
********************************************/
```

```
  //Remarks: Configure the pin of arduino as input.

/************************************************************************
*******************/
  MQ135.init();
  /*
    //If the RL value is different from 10K please assign your RL value with the
following method:
    MQ135.setRL(10);
  */
  /*****************************  MQ CAlibration
********************************************/
  // Explanation:
  // In this routine the sensor will measure the resistance of the sensor supposing
before was pre-heated
  // and now is on clean air (Calibration conditions), and it will setup R0 value.
  // We recomend execute this routine only on setup or on the laboratory and save
on the eeprom of your arduino
  // This routine not need to execute to every restart, you can load your R0 if you
know the value
  // Acknowledgements: https://jayconsystems.com/blog/understanding-a-gas-
sensor
  Serial.print("Calibrating please wait.");
  float calcR0 = 0;
  for(int i = 1; i<=10; i ++)
  {
    MQ135.update(); // Update data, the arduino will be read the voltage on the
analog pin
    calcR0 += MQ135.calibrate(RatioMQ135CleanAir);
    Serial.print(".");
  }
  MQ135.setR0(calcR0/10);
  Serial.println("  done!.");
  if(isinf(calcR0)) {Serial.println("Warning: Conection issue founded, R0 is infite
(Open circuit detected) please check your wiring and supply"); while(1);}
  if(calcR0 == 0){Serial.println("Warning: Conection issue founded, R0 is zero
(Analog pin with short circuit to ground) please check your wiring and supply");
while(1);}
  /*****************************  MQ CAlibration
********************************************/

  //MQ135.serialDebug(false);
}

void loop() {
```

```
 // MQ135.update(); // Update data, the arduino will be read the voltage on the
analog pin
  float hasil = MQ135.readSensor(); // Sensor will read PPM concentration using
the model and a and b values setted before or in the setup
  //Serial.print(jenisgas[itemcheck]);Serial.print(" :
");Serial.print(hasil);Serial.println(" PPM");
  MQ135.serialDebug(); // Will print the table on the serial port
  delay(500); //Sampling frequency
}
```

```
#include <ArduinoJson.h>

//Gas Sensor Pins
#define F2600 A1
#define MQ135 A2
#define F2602 A3
const byte speaker = 13;

//Gas Sensor Load Resistance (RL)
#define RL_F2600 7.5
#define RL_MQ135 10
#define RL_F2602 7.5

/*2600 GASSES*/
float CO_A = 1.144997421;
float CO_B = -0.21687423;

float CH4_A = 1.05777824;
float CH4_B = -0.0795645;

/*2602 GASSES*/
float NH3_A = 0.92372844;
float NH3_B = -0.291578925;

float H2S_A = 0.38881036;
float H2S_B = -0.35010059;

float VOC_A = 0.3220722;
float VOC_B = -0.6007520;


/*MQ-135 GASSES*/
//MQ-135 CO2
float A_MQ135_CO2 = 112.89808;
float B_MQ135_CO2 = -2.868463517;


//MQ-135 NOx
float A_MQ135_NOx = 34.69756084;
float B_MQ135_NOx = -3.422829698;

void setup()
{
Serial.begin(9600); // opens serial port, sets data rate128000 bps
```

```
  pinMode (speaker,OUTPUT);
}

void loop()
{
  float VRL_F2600;
  float Rs_F2600;
  float Ro_F2600 = 11.26;
  float ratio_F2600;

  VRL_F2600 = analogRead(F2600)*(5.0/1023.0);
  Rs_F2600 = ((5.0/VRL_F2600)-1)*(RL_F2600);
  ratio_F2600 = Rs_F2600/Ro_F2600;


  float ppm_CO = CO_A * pow(ratio_F2600, CO_B);
  float ppm_CH4 = CH4_A * pow(ratio_F2600, CH4_B);

  float VRL_MQ135;
  float Rs_MQ135;
  float Ro_MQ135 = 20.1;
  float ratio_MQ135;

  VRL_MQ135 = analogRead(MQ135)*(5.0/1023.0);
  Rs_MQ135 = ((5.0/VRL_MQ135)-1)*(RL_MQ135);
  ratio_MQ135 = Rs_MQ135/Ro_MQ135;


  float ppm_CO2 = A_MQ135_CO2 * pow(ratio_MQ135, B_MQ135_CO2);
  float ppm_NOx = A_MQ135_NOx * pow(ratio_MQ135, B_MQ135_NOx);

  float VRL_F2602;
  float Rs_F2602;
  float Ro_F2602 = 64.88;
  float ratio_F2602;

  VRL_F2602 = analogRead(F2602)*(5.0/1023.0);
  Rs_F2602 = ((5.0/VRL_F2602)-1)*(RL_F2602);
  ratio_F2602 = Rs_F2602/Ro_F2602;

  float ppm_VOC = VOC_A * pow(ratio_F2602, VOC_B);
  float ppm_H2S = H2S_A * pow(ratio_F2602, H2S_B);
  float ppm_NH3 = NH3_A * pow(ratio_F2602, NH3_B);
```

```
  // put your main code here, to run repeatedly:


  DynamicJsonBuffer jBuffer;
  JsonObject& root = jBuffer.createObject();

  root["CO2"] = ppm_CO2;
  root["CO"] = ppm_CO;
  root["CH4"] = ppm_CH4;
  root["NOx"] = ppm_NOx;
  root["NH3"] = ppm_NH3;
  root["H2S"] = ppm_H2S;
  root["VOC"] = ppm_VOC;

  root.prettyPrintTo(Serial);
  Serial.println("");
delay(500);
 if ((ppm_CO2 > 150.00)||(ppm_H2S > 1.00) ||(ppm_CO > 1.00)){
     Serial.println ("KADAR BAHAYA");
    digitalWrite(speaker, HIGH);

  }
  else{
      digitalWrite(speaker, LOW);
  }
}
```

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>


SoftwareSerial mySerial(17, 18);
char auth[] = "Jo6WRY1x-aFPDt_3mAALy2pBEU9NL74S";
char ssid[] = "Apaapa";
char pass[] = "apaapaapa";


void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  WiFi.begin(ssid, pass);
  int wifi_ctr = 0;
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("WiFi connected");

  Blynk.begin(auth, ssid, pass);

}

void loop() {
  Blynk.run();
  // Check WiFi Status
  while (mySerial.available())
  {
    //To understand this section better, refer to the author: https://arduinojson.org/
    const size_t capacity = JSON_OBJECT_SIZE(7) + 100;
    DynamicJsonBuffer jsonBuffer(capacity);
    JsonObject& root = jsonBuffer.parseObject(mySerial);
    if (!root.success()) {
      Serial.println("parseObject() failed");
      return;

    }
```

```
    float CO2 = root["CO2"];
    float CO = root["CO"];
    float CH4 = root["CH4"];
    float NOx = root["NOx"];
    float NH3 = root["NH3"];
    float H2S = root["H2S"];
    float VOC = root["VOC"];

    Serial.print(CO2, 5);  Serial.print(",");
    Serial.print(CO, 5);  Serial.print(",");
    Serial.print(CH4, 5);  Serial.print(",");
    Serial.print(NOx, 5);  Serial.print(",");
    Serial.print(NH3, 5);  Serial.print(",");
    Serial.print(H2S, 5); Serial.print(",");
    Serial.println(VOC, 5);
  Blynk.virtualWrite(V0, CO2);
  Blynk.virtualWrite(V1, H2S);
  Blynk.virtualWrite(V2, NH3);
  Blynk.virtualWrite(V4, CO);
  if ((CO2 > 150.00)||(H2S > 1.00)|| (CO > 1.00)){
     Serial.println ("KADAR BAHAYA");
        Blynk.notify("KADAR BAHAYA!!!");

  }
  else{
  }
  }

}
```

LAMPIRAN-2

GAMBAR BENTUK FISIK ALAT