

```

#include <SDS011.h>

#include <DHTesp.h>

#include "MQ135.h"

#include <MQ131.h>

#include <WiFi.h>

#include "DFRobot_MICS.h"

#define DHTpin 14 //D5 of NodeMCU is GPIO14

DHTesp dht;

#define CALIBRATION_TIME 3 // Default calibration time is three minutes

// When using the Breakout version, use the following program to construct an object from
DFRobot_MICS_ADC

/**!

    adcPin is A0~A5

    powerPin is General IO

*/

#define ADC_PIN 36

#define POWER_PIN 25

DFRobot_MICS_ADC mics(/adcPin/ADC_PIN, /powerPin/POWER_PIN);

float Temperature;

float Humidity;

int data = 34;

float p10,p25;

int error;

SDS011 my_sds;

```

```
void setup() {  
  
    // put your setup code here, to run once:  
  
    my_sds.begin(17,16 ); //D4 of node mcu  
  
    Serial.begin(115200);  
  
    dht.setup(DHTpin, DHTesp::DHT11);  
  
    // Init the sensor  
  
    // - Heater control on pin 2  
  
    // - Sensor analog read on pin A0  
  
    // - Model LOW_CONCENTRATION  
  
    // - Load resistance RL of 1MOhms (1000000 Ohms)  
  
    MQ131.begin(2,39, LOW_CONCENTRATION, 1000000);  
  
  
    Serial.println("Calibration in progress...");  
  
  
    MQ131.calibrate();  
  
  
    Serial.println("Calibration done!");  
  
    Serial.print("R0 = ");  
  
    Serial.print(MQ131.getR0());  
  
    Serial.println(" Ohms");  
  
    Serial.print("Time to heat = ");  
  
    Serial.print(MQ131.getTimeToRead());  
  
    Serial.println(" s");  
  
  
  
    // Serial.begin(115200);  
  
    while(!Serial);
```

```

while(!mics.begin()){

    Serial.println("NO Deivces !");

    delay(1000);

} Serial.println("Device connected successfully !");


/**!

Gets the power mode of the sensor

The sensor is in sleep mode when power is on,so it needs to wake up the sensor.

The data obtained in sleep mode is wrong

*/

uint8_t mode = mics.getPowerState();

if(mode == SLEEP_MODE){

    mics.wakeUpMode();

    Serial.println("wake up sensor success!");

}else{

    Serial.println("The sensor is wake up mode");

}


/**!

Do not touch the sensor probe when preheating the sensor.

Place the sensor in clean air.

The default calibration time is 3 minutes.

*/

while(!mics.warmUpTime(CALIBRATION_TIME)){

    Serial.println("Please wait until the warm-up time is over!");

    delay(1000);

}

```

```
}
```

```
void loop() {
```

```
    error = my_sds.read(&p25,&p10);
```

```
    data = analogRead(A0); //A0 of node mcu
```

```
    float humidity = dht.getHumidity();
```

```
    float temperature = dht.getTemperature();
```

```
    // int chk = DHT.read11(DHT11_PIN);
```

```
    if (! error) {
```

```
        Serial.println("P2.5: "+String(p25)+"microgm/metercube");
```

```
        Serial.println("P10: "+String(p10)+"microgm/metercube");
```

```
        Serial.print("MQ3:");
```

```
        Serial.println(data);
```

```
        //Serial.println(a);
```

```
    // Serial.print(dht.getStatusString());
```

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

```
    Serial.println(humidity, 1);
```

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

```
    Serial.println(temperature, 1);
```

```
    //Serial.print("\t\t");
```

```
    // Serial.print(dht.toFahrenheit(temperature), 1);
```

```

// Serial.print("\t\t");

// Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1);

// Serial.print("\t\t");

// Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);

/*MQ135 gasSensor = MQ135(A0);

float air_quality = gasSensor.getPPM();

Serial.print("Air Quality: ");

Serial.print(air_quality);

Serial.println(" PPM");

Serial.println();

delay(1000);*/

Serial.println("Sampling...");

MQ131.sample();

Serial.print("Concentration O3 : ");

Serial.print(MQ131.getO3(PPM));

Serial.println(" ppm");

Serial.print("Concentration O3 : ");

Serial.print(MQ131.getO3(PPB));

Serial.println(" ppb");

Serial.print("Concentration O3 : ");

Serial.print(MQ131.getO3(MG_M3));

Serial.println(" mg/m3");

Serial.print("Concentration O3 : ");

Serial.print(MQ131.getO3(UG_M3));

Serial.println(" ug/m3");

```

```
delay(2000);  
  
float gasdata = mics.getGasData(NO2);  
  
Serial.print(gasdata,1);  
  
Serial.println(" PPM");  
  
delay(1000);
```

```
}
```

```
}
```