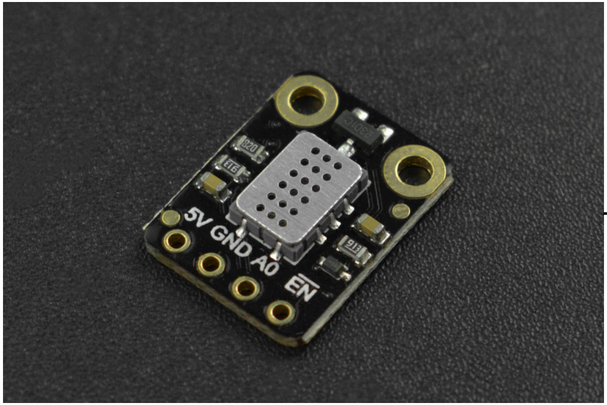
SKU:SEN0440 (https://www.dfrobot.com/product-2419.html)



(https://www.dfrobot.com/product-2419.html)

Introduction

This is a hydrogen and nitrogen oxide gas concentration sensor launched by DFRobot. The sensor uses MEMS technology to support gas concentration detection of CO, CH4, C2H5OH, C3H8, C4H10, H2, H2S and NH3. The matching sample code integrates the concentration

conversion formula of various gas to facilitate the testing and use of sensors. The product supports 5V power supply, analog voltage output, and has power supply enable/disable pins for low power consumption.

Features

- Support a variety of harmful gas detection
- Integrate the calculation formulas of various gas concentration
- Low power consumption
- I2C digital output
- Compatible with the 3.3~5.5V master controller

Specification

- Detection of Physical Quantities: Gas concentration or gas leakage of CO, CH4, C2H5OH, Detection of Physical Quantities: Gas concentration of gas leakage of CO, CH4, C2H5OH, C3H8, C4H10, H2, H2S, NH3

- Operating voltage: 4.9 ~ 5.1V DC
- Power dissipation: 0.45W
- Output signal: Analog quantity
- Measuring range:
  - 1 – 1000ppm(Carbon monoxide CO)
  - 10 – 500ppm(Ethanol C2H5OH)
  - 1 – 1000ppm(Hydrogen H2)
  - 1 – 500ppm(Ammonia NH3)
  - >1000ppm(Methane CH4)
- Working temperature: -30 ~ 85℃
- Working humidity: 5 ~ 95%RH （No condensation）
- Storage temperature: -40~85℃
- Lifespan: >2 years （in the air）
- Circuit board size：12mm×16mm
- Mounting hole size：Inner diameter 2mm/outer diameter 4mm
- Weight：7g

Board Overview

No.	Name	Function
1	5V	+
2	GND	-
3	A0	Analog quantity output
4	EN	Power Supply diable/enable pin(enable at a low level， disable at a high level)

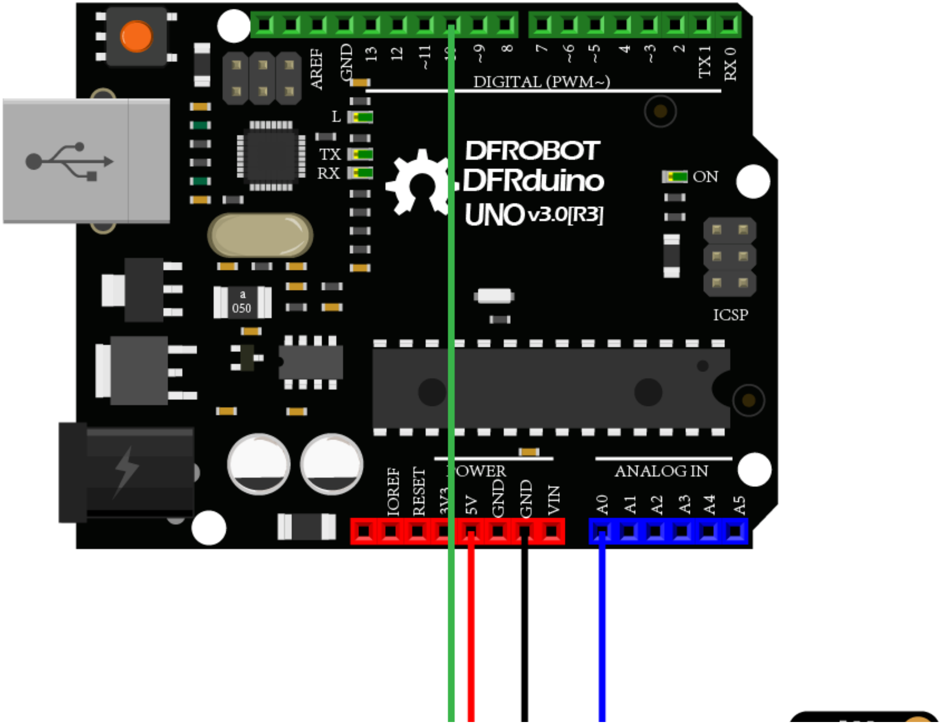
Tutorial for Arduino use

Download the program to the UNO, open Serial Port Monitor to view the gas concentration, raw data, and other parameters.

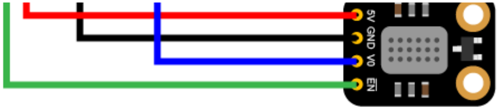
Software and hardware preparation

- **hardware**
    - DFRuino UNO R3 (https://www.dfrobot.com/product-838.html) x1
    - SEN0441 Fermion: MEMS Gas Sensor - MiCS-5524 (https://www.dfrobot.com/product-2419.html) x1
    - Several Dupont lines
- **Software**
    - Arduino IDE Arduino IDE (https://www.arduino.cc/en/Main/Software)
    - Download and install DFRobot_MICS Library (https://github.com/DFRobot/DFRobot_MICS/archive/refs/heads/master.zip) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))

Connection Diagram


AREF
GND
13
12
~11
10
~9
8
7
~6
~5
4
~3
2
TX 1
RX 0
DIGITAL (PWM~)
L
TX
RX
DFROBOT
DFRduino
UNO v3.0[R3]
ON
ICSP
a
050
POWER
IOREF
RESET
3V3
5V
GND
GND
VIN
ANALOG IN
A0
A1
A2
A3
A4
A5

()



Sample 1 - Read the gas concentration(PPM) data calculated by the sensor

Download the sample program to the Arduino UNO, and open Serial Port Monitor for viewing the NO2 gas concentration (PPM) data.

Steps

- Connect the module with Arduino according to the connection diagram above.
- Download and install DFRobot_MICS Library (https://github.com/DFRobot/DFRobot_MICS/archive/refs/heads/master.zip) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))
- Open Arduino IDE
  - Upload the following sample code to the Arduino UNO.
  - Or open the **getGasPPM.ino** code in the library file sample, revise the configuration in the code(comment part of the I2C link code, uncomment the codes for breakout version), and modify the detection gas to NO2, then burn the modified code to the

Arduino UNO.

- Open the serial port monitor of the Arduino IDE, adjust the baud rate to 115200, and observe the serial port print results.

```
 /*!
  * @file  getGasPPM.ino
  * @brief Reading Gas concentration, A concentration of one part per million (PPM).
  * @n When using IIC device, select I2C address, set the dialing switch A0, A1 (Address_0
  * @n When using the Breakout version, connect the adcPin and PowerPin
  * @copyright   Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
  * @licence     The MIT License (MIT)
  * @author      ZhixinLiu(zhixin.liu@dfrobot.com)
  * @version     V1.1
  * @date        2021-04-19
  * @get         from https://www.dfrobot.com
  * @url         https://github.com/dfrobot/DFRobot_MICS
  */
#include "DFRobot_MICS.h"

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

// When using the Breakout version, use the following program to construct an object from
/**!
```

```
/**!
  adcPin is A0~A5
  powerPin is General IO
*/
#define ADC_PIN   A0

#define POWER_PIN 10
DFRobot_MICS_ADC mics(/*adcPin*/ADC_PIN, /*powerPin*/POWER_PIN);

void setup()
{
  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();
    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()
{
  /**!
    Gas type:
    MICS-4514 You can get all gas concentration
    MICS-5524 You can get the concentration of CH4, C2H5OH, H2, NH3, CO
    MICS-2714 You can get the concentration of NO2
      Methane          (CH4)    (1000 - 25000)PPM
      Methane          (CH4)    (1000 - 25000)PPM
      Ethanol          (C2H5OH) (10   - 500)PPM
      Hydrogen         (H2)     (1    - 1000)PPM
      Ammonia          (NH3)    (1    - 500)PPM
      Carbon Monoxide  (CO)     (1    - 1000)PPM

      Nitrogen Dioxide (NO2)    (0.1  - 10)PPM
  */
  float gasdata = mics.getGasData(C2H5OH);
  Serial.print(gasdata,1);
  Serial.println(" PPM");
  delay(1000);
  //mics.sleepMode();
}
```

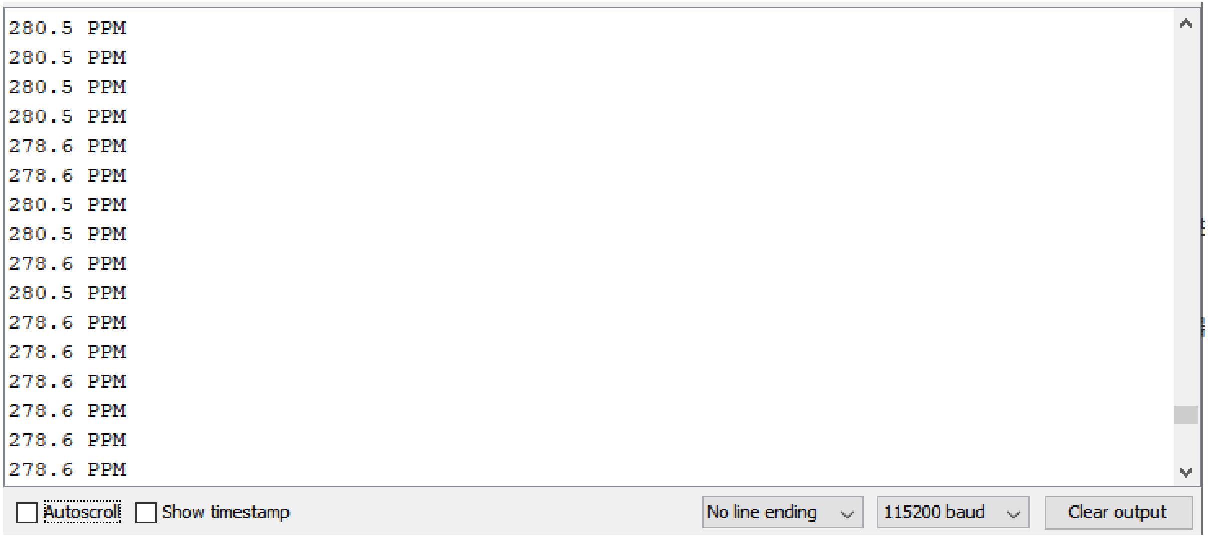
Result

Open the serial port monitor and warm up for approximately 3 minutes to obtain the alcohol (C2H5OH) gas concentration data.

Notes:

- **If you need to detect other gases, modify the detected gas settings in the sample code** yourself

- The sensor takes 3 minutes to warm up



```
280.5 PPM
280.5 PPM
280.5 PPM
280.5 PPM
278.6 PPM
278.6 PPM
280.5 PPM
280.5 PPM
278.6 PPM
280.5 PPM
278.6 PPM
278.6 PPM
278.6 PPM
278.6 PPM
278.6 PPM
278.6 PPM
```

☐ Autoscroll ☐ Show timestamp   No line ending   115200 baud   Clear output

()

Sample 2 - Detection of Gas Leakage

The MEMS chip MiCS-5524 reacts to the NO2, H2 and NO gas, but the linearity of some gas concentration feedback value is poor. Therefore, the concentration value does not have a reference value, but can be used as the determination basis of the gas leakage.

If you need a leak detection of the relevant gas, you can download the sample program to the Arduino UNO, open Serial Port monitor to see if the set gas is leaking.

Steps

- Connect the module with Arduino according to the connection diagram above.
- Download and install DFRobot_MICS Library (https://github.com/DFRobot/DFRobot_MICS/archive/refs/heads/master.zip).
- Open Arduino IDE
  - Upload the following sample code to the Arduino UNO.
  - Or open the **getGasExist.ino** code in the library file sample，revise the configuration in the code (comment part of the I2C link code, uncomment the codes for breakout version), and modify the detection gas to NO2, then burn the modified code to the Arduino UNO.
- Open the serial port monitor of the Arduino IDE, adjust the baud rate to 115200, and observe the serial port print results.

```
 /*!
  * @file  getGasExist.ino
  * @brief Reading Gas concentration, A concentration of one part per million (PPM).
  * @n When using IIC device, select I2C address, set the dialing switch A0, A1 (Address_0
  * @n When using the Breakout version, connect the adcPin and PowerPin
  * @copyright   Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
  * @licence     The MIT License (MIT)
  * @author      ZhixinLiu(zhixin.liu@dfrobot.com)
  * @version     V1.1
  * @date        2021-04-19
  * @get         from https://www.dfrobot.com
  * @url         https://github.com/dfrobot/DFRobot_MICS
  */
#include "DFRobot_MICS.h"
#define CALIBRATION_TIME   3                      // Default calibration time is three mir
// When using the Breakout version, use the following program to construct an object from
  adcPin is A0~A5
  powerPin is General IO
*/#define ADC_PIN   A0#define POWER_PIN 10
```

```
*/#define ADC_PIN   A0#define POWER_PIN 10
DFRobot_MICS_ADC mics(/*adcPin*/ADC_PIN, /*powerPin*/POWER_PIN);
void setup() {
  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");
  }

  /**!
```

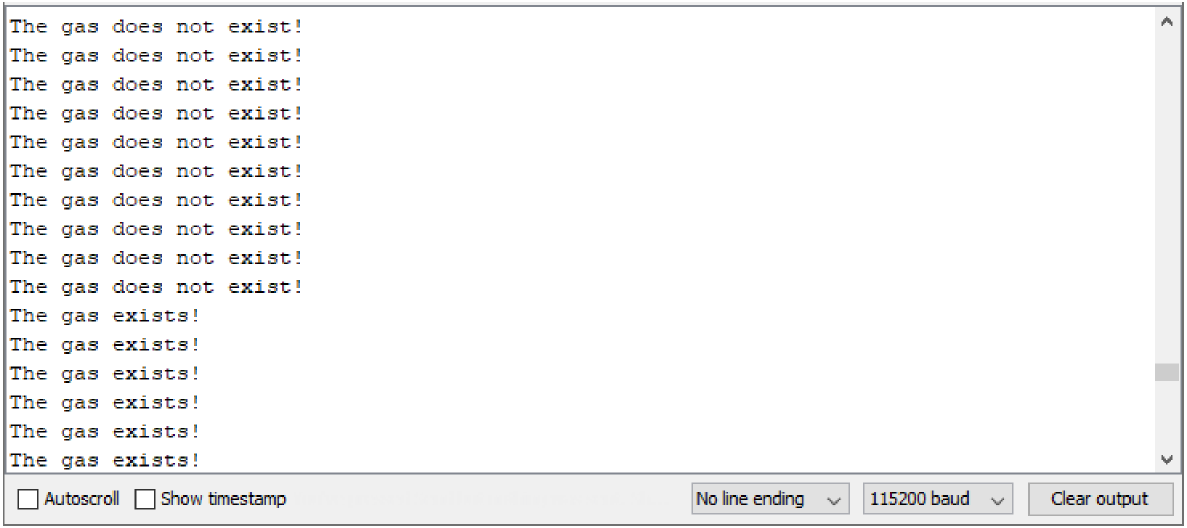
```
     Do not touch the sensor probe when preheating the sensor.
     Place the sensor in clean air.
     The default calibration time is 3 minutes.
  */
  while(!mics.warmUpTime(CALIBRATION_TIME)){

    Serial.println("Please wait until the warm-up time is over!");
    delay(1000);
  }}
void loop() {
  /**!
    Type of detection gas
    MICS-4514 You can get all gas state
    MICS-5524 You can get the state of CO, CH4, C2H5OH, C3H8, C4H10, H2, H2S, NH3
    MICS-2714 You can get the state of NO2, H2 ,NO
      CO         = 0x01  (Carbon Monoxide)
      CH4        = 0x02  (Methane)
      C2H5OH     = 0x03  (Ethanol)
      C3H8       = 0x04  (Propane)
      C4H10      = 0x05  (Iso Butane)
      H2         = 0x06  (Hydrogen)
      H2S        = 0x07  (Hydrothion)
      NH3        = 0x08  (Ammonia)
      NO         = 0x09  (Nitric Oxide)
      NO2        = 0x0A  (Nitrogen Dioxide)
  */
  */
  int8_t gasFlag = mics.getGasExist(CH4);
  if(gasFlag == EXIST){
    Serial.println("The gas exists!");
  }else{

    Serial.println("The gas does not exist!");
  }
  delay(1000);
  //mics.sleepMode();}
```

Result

Open the serial port monitor and warm up for about 3 minutes, you can detect methane (CH4) leakage in real time.

Notes:

- **If you need to detect other gases, modify the detected gas settings in the sample code yourself**
- **The sensor takes 3 minutes to warm up**



```
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas does not exist!
The gas exists!
The gas exists!
The gas exists!
The gas exists!
The gas exists!
The gas exists!
```

()

Sample 3 - Obtain Sensor Analog Value

If you want to obtain the original sensor data, calculate the gas concentration or add your own temperature compensation functions, you can download the sample program to the Arduino

UNO, open serial port monitor to view the original voltage output of the MEMS chip MiCS-5524.

Steps

- Connect the module with Arduino according to the connection diagram above.
- Download and install DFRobot_MICS Library (https://github.com/DFRobot/DFRobot_MICS/archive/refs/heads/master.zip) (About how to install the library?)
- Open Arduino IDE
    - Upload the following sample code to the Arduino UNO.
    - Or open the getGasPPM.ino code in the library file sample, revise the configuration in the code (comment part of the I2C link code, uncomment the codes for breakout version), and modify the detection gas to NO2, then burn the modified code to the Arduino UNO.
- Open the serial port monitor of the Arduino IDE, adjust the baud rate to 115200, and observe the serial port print results.

```
 /*!
  * @file  getADCData.ino
  * @brief Reading MICS sensor ADC original value
  * @n When using IIC device, select I2C address, set the dialing switch A0, A1 (Address_
  * @n When using the Breakout version, connect the adcPin and PowerPin
  * @copyright   Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
  * @licence     The MIT License (MIT)
  * @author      ZhixinLiu(zhixin.liu@dfrobot.com)
  * @version     V1.1
  * @date        2021-04-19
  * @get         from https://www.dfrobot.com
  * @url         https://github.com/dfrobot/DFRobot_MICS
  */
#include "DFRobot_MICS.h"
// When using the Breakout version, use the following program to construct an object from
  adcPin is A0~A5
  powerPin is General IO
*/#define ADC_PIN   A0#define POWER_PIN 10
DFRobot_MICS_ADC mics(/*adcPin*/ADC_PIN, /*powerPin*/POWER_PIN);
DFRobot_MICS_ADC mics(/*adcPin*/ADC_PIN, /*powerPin*/POWER_PIN);
void setup() {
  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");
  }}
void loop() {
  int16_t ox_data  = 0;
  int16_t red_data = 0;
  /**!
  /**!
    MICS-5524 Only OX_MODE ADC data can be obtained
    MICS-2714 Only RED_MODE ADC data can be obtained
    MICS-4514 Gravity can obtain AllMode ADC data
  */

  ox_data  = mics.getADCData(OX_MODE);
  //red_data = mics.getADCData(RED_MODE);
  Serial.print("ox  data = ");
  Serial.println(ox_data);
  //Serial.print("red data = ");
  //Serial.println(red_data);
  delay(1000);}
```

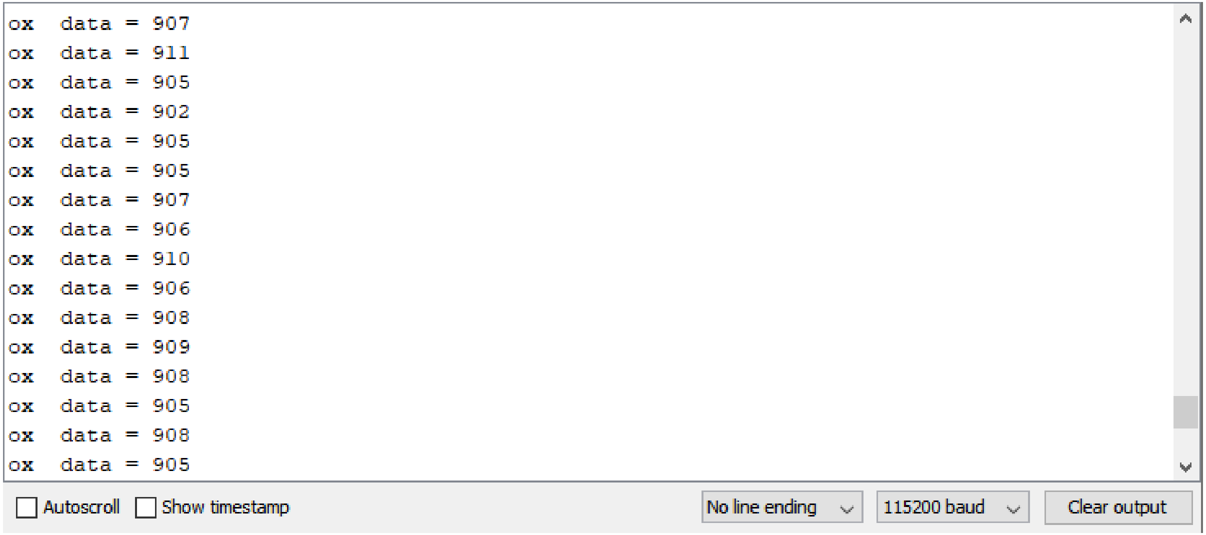
Results

Open the serial port monitor and warm up for about 3 minutes, you can obtain the MiCS-5524 raw data of MEMS chip.

Notes:

- **The red detection unit is integrated in the MiCS-5524 to calculate the concentration data of the different gas. For specific curves, please refer to the sensor data curve in the wiki.**

- **The sensor takes 3 minutes to warm up**



```
ox  data = 907
ox  data = 911
ox  data = 905
ox  data = 902
ox  data = 905
ox  data = 905
ox  data = 907
ox  data = 906
ox  data = 910
ox  data = 906
ox  data = 908
ox  data = 909
ox  data = 908
ox  data = 905
ox  data = 908
ox  data = 905
```

□ Autoscroll □ Show timestamp No line ending 115200 baud

()

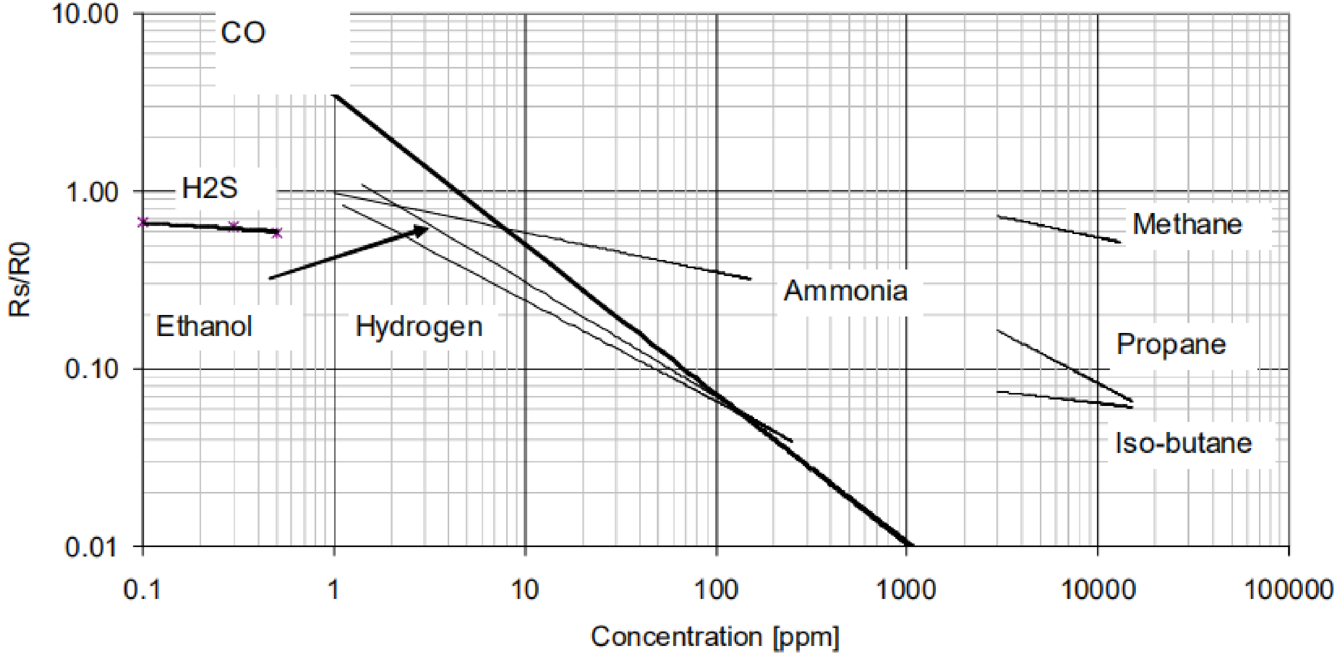
Precautions for use

- Do not expose the sensor to high concentrations of organic solvents, silicone vapor or cigarette smoke to avoid poisoning the sensitive layer cigarette smoke to avoid poisoning the sensitive layer.

- For stable performance, preheat the module for about 3 minutes before testing.
- Sensors shall be placed in a filtered enclosure to protect them from water and dust.

Sensor Data Curve

The MiCS-5524 contains a CO detection unit for detecting the gas concentration data or gas presence of CO, CH4, C2H5OH, C3H8, C4H10, H2, H2S and NH3, with the reference curve as shown in the figure below.



Continuous power ON, 25°C, 50% RH

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

- MiCS-5524 Datasheet (https://img.dfrobot.com.cn/wiki/5b973267c87e6f19943ab3ad/86011e31a53db4f1bb6a3978de97a135.pdf)
- Schematics Diagram (https://dfimg.dfrobot.com/nobody/wiki/bc7b994d37fdaffc6e43624bda91cbde.pdf)
- Dimension and Component Layout (https://dfimg.dfrobot.com/nobody/wiki/065f0240e12ab11e527c4c6cf91799ba.pdf)

DFshopping_car1.png Get **Fermion Mems Gas Sensor (https://www.dfrobot.com/product-2419.html)** from **DFRobot Store or DFRobot Distributor. (https://www.dfrobot.com/new-products?gclid=EAIaIQobChMImb_MxbrL8gIVgagWCh0-9gy3EAAYASAAEgIYIfD_BwE)** products?gclid=EAIaIQobChMImb_MxbrL8gIVgaqWCh0-9gy3EAAYASAAEgIYifD_BwE)