Gravity: Analog Infrared CO2 Sensor For Arduino
SKU: SEN0219

Introduction

DFRobot released its latest high-precision analog infrared CO2 sensor. The effectively measuring range is from 0 to 5000ppm. This sensor is based on non-dispersive infrared (NDIR) technology and has good selectivity and oxygen-free dependency. Besides, its service life could up to 5 years! It integrates temperature compensation and support DAC output. Most importantly, the product is easy to use; it is compatible with all types of microcontrollers with ADC function.

In addition, this product is a high-performance sensor that combines technology of mature infrared absorption gas detection with precision optical circuit design as well as sophisticated circuit design. It has characteristics such as high sensitivity, high resolution, low power consumption, fast response, anti-water vapor interference, no poisoning, high stability and long life.

This sensor is able to directly compatible with the DFRobot Arduino IO expansion board thanks to its external DFRobot Gravity interface. This character simplify the use of the sensor as it is plug and play and no need additional wiring.

This product could be widely used in HVAC, indoor air quality monitoring, industrial process and security protection monitoring, agriculture and animal husbandry production process monitoring.

Feature

- Waterproof and anti-corrosion
- High sensitivity
- Low power consumption
- Excellent stability
- Temperature compensation
- Excellent linear output
- High cycle life
- Anti-water vapor interference
- No poisoning

Specification

- Gas Detection: Carbon Dioxide (CO_2)
- Operating Voltage: 4.5 ~ 5.5V DC
- Average Current: <60mA @ 5V
- Peak Current: 150mA @ 5V
- Output Signal: Analog output (0.4 ~ 2V)
- Measuring Range: 0 ~ 5000ppm
- Accuracy: ± (50ppm + 3% reading)
- Preheating Time: 3min
- Response Time: 120s
- Operating Temperature: 0 ~ 50 ℃
- Operating Humidity: 0 ~ 95% RH (no condensation)
- Service Life: >5 years
- Size: 37mm * 69mm

Board Overview

SEN0219 CO2 Sensor

| Num	Label	Description
1	Signal	Analog Output (0.4~2V)
2	VCC	VCC (4.5~5.5V)
3	GND	GND

Tutorial

In this tutorial, we'll show you how to use DFRobot Infrared Sensor within 5 minutes.

Requirements

- **Hardware**
  DFRduino UNO x 1
  DFRobot Gravity: Infrared CO2 Sensor
  Gravity: IO Expansion Shield for Arduino V7.1

- **Software**
  Arduino IDE Click to Download Arduino IDE from Arduino®
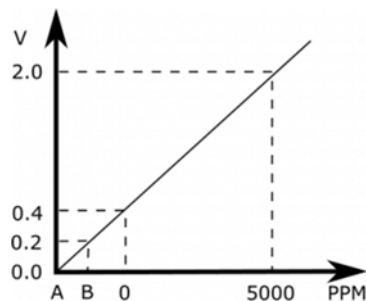 https://www.arduino.cc/en/Main/Software%7C

Connection Diagram
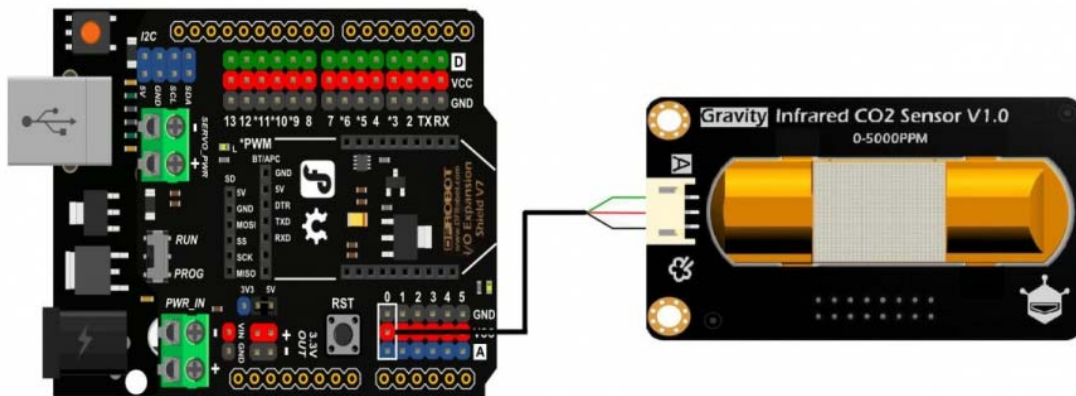
SEN0219_Connection

Data Analysis

When the IR CO2 sensor get 5V power supply, it will output 0.4~2V analog value, corresponding to 0~5000 ppm; And when the sensor finds faults during the self-checking process, it will output 0V.

A: Fault
B: Preheat

Sample Code

```
/**************************************************
 * Infrared CO2 Sensor0-5000ppm
 * **************************************************
 * This example The sensors detect CO2
 *
 * @author lg.gang(lg.gang@qq.com)
 * @version  V1.0
 * @date   2016-7-6
 *
 * GNU Lesser General Public License.
 * See <http://www.gnu.org/licenses/> for details.
 * All above must be included in any redistribution
 * **************************************************/
int sensorIn = A0;

void setup(){
  Serial.begin(9600);
  // Set the default voltage of the reference voltage
  analogReference(DEFAULT);
}

void loop(){
  //Read voltage
  int sensorValue = analogRead(sensorIn);

  // The analog signal is converted to a voltage
  float voltage = sensorValue*(5000/1024.0);
  if(voltage == 0)
  {
    Serial.println("Fault");
  }
  else if(voltage < 400)
  {
```

```
    Serial.println("preheating");

  }

  else

  {

    int voltage_diference=voltage-400;

    float concentration=voltage_diference*50.0/16.0;

    // Print Voltage

    Serial.print("voltage:");

    Serial.print(voltage);

    Serial.println("mv");

    //Print CO2 concentration

    Serial.print(concentration);

    Serial.println("ppm");

  }

  delay(100);

}
```

Expected Results

Open your IDE serial monitor and wait for 3 minutes (preheat process), then you'll see the finial data.
(Indoor Temperature: 30℃)