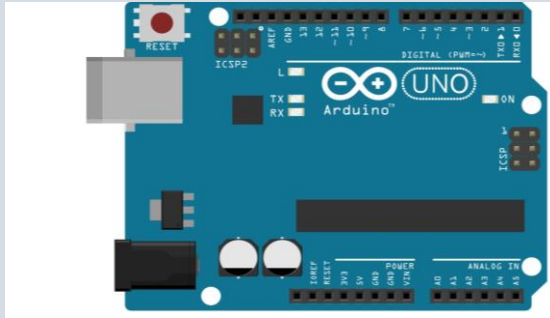


Mon Arduino veut un Capteur de CO2 SEN0159

✓ Présentation du matériel.



Introduction

« L'effet de serre » fait fondre les icebergs chaque minute. En connaissant la concentration exacte de CO₂, nous pouvons faire quelque chose pour réduire le niveau de CO₂ de l'atmosphère et protéger notre terre. Pour cette raison, DFRobot engineering a conçu un capteur de CO₂ de haute qualité. Il s'agit du premier **capteur de CO₂** sur le marché du matériel opensource. La tension de sortie du module diminue à mesure que la concentration de CO₂ augmente. Le potentiomètre embarqué est conçu pour régler le seuil de tension. Une fois que la concentration de CO₂ est suffisamment élevée (la tension est inférieure au seuil), un signal numérique (ON/OFF) sera libéré.

- Il a un capteur de gaz MG-811 à bord qui est très sensible au CO₂ et moins sensible à l'alcool et au CO, faible dépendance à l'humidité et à la température. Tous les composants ont une qualité industrielle qui assure stabilité et reproductibilité.
- Le circuit de chauffage embarqué apporte la meilleure température pour que le capteur fonctionne. L'entrée d'alimentation 5V sera augmentée à 6V pour le chauffage.
- Ce capteur dispose d'un circuit de conditionnement embarqué pour amplifier le signal de sortie.



Soit le programme suivant :

Introduction



- Une alimentation externe (7 ~ 12V) est nécessaire pour alimenter la carte du microcontrôleur lorsque vous utilisez ce module de capteur de CO₂.
- Ce module est un capteur électrochimique, vous devez l'étalonner avant la mesure réelle.

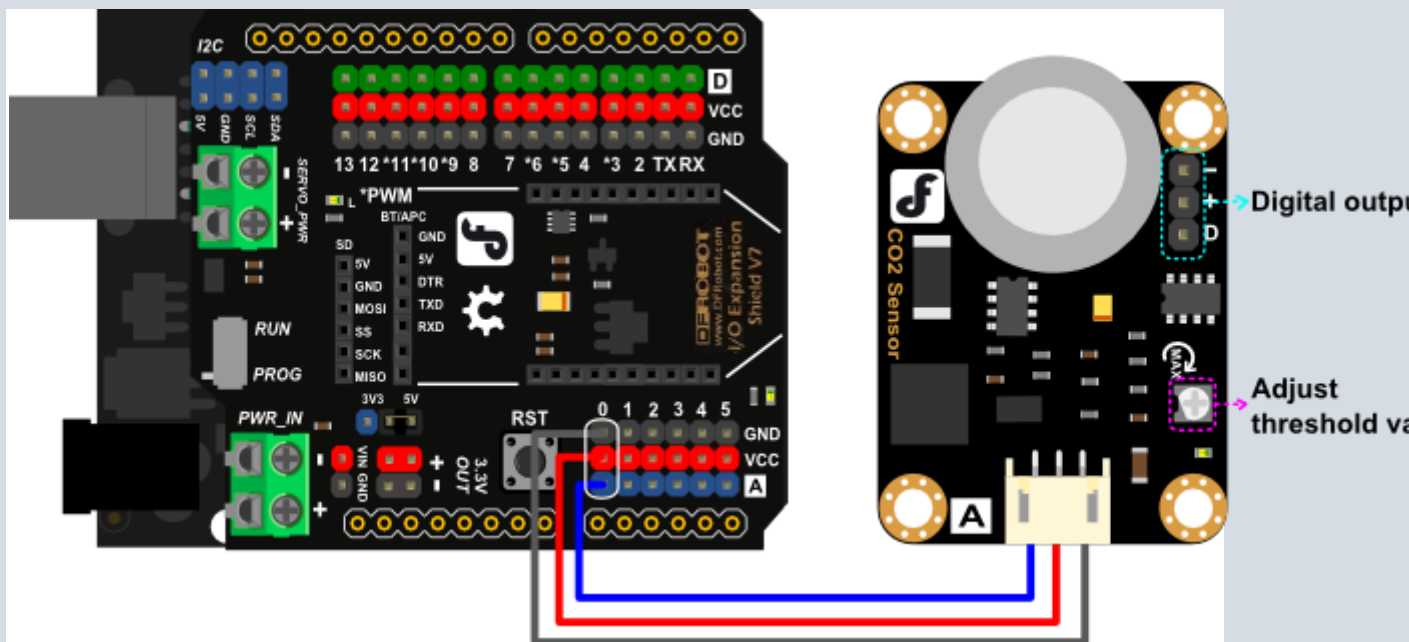
Spécification

- Tension de fonctionnement: 5V
- Interface: analogique (compatible avec la gravité)
- Une sortie numérique
- Connecteur de haute qualité
- Surface d'or par immersion
- Circuit de chauffage embarqué
- Taille: 32x42mm (1.26x1.65 »)



Arduino

Diagramme de connexion



Tutoriel

Comment utiliser ce module ? C'est très facile. Vous devez régler le potentiomètre embarqué sur la valeur seuil. Il suffit de faire éteindre la led rouge. Avec la concentration de CO2 est suffisamment élevée pour que la tension de sortie du capteur soit supérieure à la valeur seuil, la led sera allumée. Si vous connectez un buzzer au module (côté droit), vous entendrez l'alarme.

Étalonnage

Ce module est un capteur d'électrochimie, vous devez l'étalonner avant la mesure réelle. Vous devez fournir une alimentation stable à ce module et le capteur chauffera pendant le fonctionnement. Veuillez placer ce module dans une zone où l'air est propre. Après un travail continu pendant environ 48 heures, vous pouvez mesurer la tension de sortie de ce module. Modifiez ensuite la définition dans le code avec la valeur de tension (unité:V) divisée par 8,5.

```
#define ZERO_POINT_VOLTAGE (voltage/8.5)
```

Copier

Par exemple, la tension que vous avez mesurée à partir du module est de 2,4 V, puis de $2,4/8,5 = 0,282$. Modifiez donc la définition comme ci-dessous:



```
#define
```

```
ZERO_POINT_VOLTAGE
```

```
(0.282)
```

Copier

Après la modification, téléchargez l'exemple de code sur votre carte Arduino.

Exemple de code

```
/******Demo for MG-811 Gas Sensor Module
V1.1*****
Author: Tiequan Shao: tiequan.shao@sandboxelectronics.com
Peng Wei: peng.wei@sandboxelectronics.com

Licence: Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA
3.0)

Note: This piece of source code is supposed to be used as a demonstration
ONLY. More
sophisticated calibration is required for industrial field
application.

Sandbox Electronics
2012-05-31
*****/

/******Hardware Related
Macros*****/
#define MG_PIN (A0) //define which analog
input channel you are going to use
#define BOOL_PIN (2)
#define DC_GAIN (8.5) //define the DC gain
of amplifier

/******Software Related
Macros*****/
#define READ_SAMPLE_INTERVAL (50) //define how many
samples you are going to take in normal operation
#define READ_SAMPLE_TIMES (5) //define the time
interval(in milisecond) between each samples in
//normal operation

/******Application Related
Macros*****/
//These two values differ from sensor to sensor. user should derermine this
value.
#define ZERO_POINT_VOLTAGE (0.220) //define the output of
the sensor in volts when the concentration of CO2 is 400PPM
#define REACTION_VOLTGAEE (0.030) //define the voltage
drop of the sensor when move the sensor from air into 1000ppm CO2

/******Globals*****/
float CO2Curve[3] =
{2.602,ZERO_POINT_VOLTAGE,(REACTION_VOLTGAEE/(2.602-3))};
//two points are taken
from the curve.
```



Arduino

```

//with these two
points, a line is formed which is
//"approximately
equivalent" to the original curve.
//data format:{ x, y,
slope}; point1: (lg400, 0.324), point2: (lg4000, 0.280)
//slope = ( reaction
voltage ) / (log400 -log1000)

void setup()
{
    Serial.begin(9600); //UART setup, baudrate
= 9600bps
    pinMode(BOOL_PIN, INPUT); //set pin to input
    digitalWrite(BOOL_PIN, HIGH); //turn on pullup
resistors

    Serial.print("MG-811 Demonstration\n");
}

void loop()
{
    int percentage;
    float volts;

    volts = MGRead(MG_PIN);
    Serial.print( "SEN0159:" );
    Serial.print(volts);
    Serial.print( "V          " );

    percentage = MGGetPercentage(volts,CO2Curve);
    Serial.print("CO2:");
    if (percentage == -1) {
        Serial.print( "<400" );
    } else {
        Serial.print(percentage);
    }

    Serial.print( "ppm" );
    Serial.print("\n");

    if (digitalRead(BOOL_PIN) ){
        Serial.print( "====BOOL is HIGH====" );
    } else {
        Serial.print( "====BOOL is LOW====" );
    }

    Serial.print("\n");

    delay(500);
}

/***** MGRead *****/
*****
Input:  mg_pin - analog channel
Output: output of SEN-000007
Remarks: This function reads the output of SEN-000007
*****/
float MGRead(int mg_pin)
{

```



```

int i;
float v=0;

for (i=0;i<READ_SAMPLE_TIMES;i++) {
    v += analogRead(mg_pin);
    delay(READ_SAMPLE_INTERVAL);
}
v = (v/READ_SAMPLE_TIMES) *5/1024 ;
return v;
}

/***** MQGetPercentage *****/
*****
Input:  volts    - SEN-000007 output measured in volts
        pcurve   - pointer to the curve of the target gas
Output: ppm of the target gas
Remarks: By using the slope and a point of the line. The x(logarithmic
value of ppm)
          of the line could be derived if y(MG-811 output) is provided. As
it is a
          logarithmic coordinate, power of 10 is used to convert the result
to non-logarithmic
          value.
*****
*****/
int  MQGetPercentage(float volts, float *pcurve)
{
    if ((volts/DC_GAIN )>=ZERO_POINT_VOLTAGE) {
        return -1;
    } else {
        return pow(10, ((volts/DC_GAIN)-pcurve[1])/pcurve[2]+pcurve[0]);
    }
}

```

Compléter le tableau suivant en utilisant le TFA Dostmann AirCo2ntrol COACH Appareil de mesure du dioxyde de carbone.

	Capteur de CO2 IR en volt	Capteur de CO2 IR en ppm	Mesure du TFA en ppm	Erreur en %
1 ^{er} mesure				
2 ^{ème} mesure				
3 ^{ème} mesure				
4 ^{ème} mesure				

Tracer la fonction de transfert du capteur de CO2 IR, de la valeur de CO2 en ppm en fonction des volts.



Conclure.



Arduino