SEN-000007

MG-811 CO2 Sensor Module

Description

This sensor module has an MG-811 onboard as the sensor component. There is an onboard signal conditioning circuit for amplifying output signal and an onboard heating circuit for heating the sensor. The MG-811 is highly sensitive to CO2 and less sensitive to alcohol and CO. It could be used in air quality control, ferment process, in-door air monitoring application. The output voltage of the module falls as the concentration of the CO2 increases.

Features

- Analog and digital output
- Onboard signal conditioning circuit
- Onboard heating circuit
- Sensor jack eliminates soldering the sensor and allows plug-and-play.
- 4-pin interlock connectors onboard
- 4-pin interlock cables included in the package
- Compact size

MG-811 Specifications

Symbol	Parameter	Value	Remarks
V_H	Heating Voltage	6.0±0.1V	AC or DC
R_H	Heating Resistor	~30.0 Ohm	At room temperature
I_H	Heating Current	~200mA	
P_H	Heating Power	~1200mW	
Tao	Operating Temperature	-20 - 50℃	
Tas	Storage Temperature	-20 - 70℃	
EMF	Output	100-600mV	400-10000ppm CO2

SEN-000007 Pinout

Name	Description	Remarks
VCC	5V power supply for signal conditioning	<5.5V
VOUT	Analog voltage signal output	
BOOL	Comparator output	Open drain
HEAT	Heating power supply	7.5-12V*
VSET*	Heating voltage select	0-5V
GND	Common ground	

*Please note that the heating voltage should be 7.5-12V instead of 6-24V as marked around the barrel connector on the PCB.

Typical Application Schematics

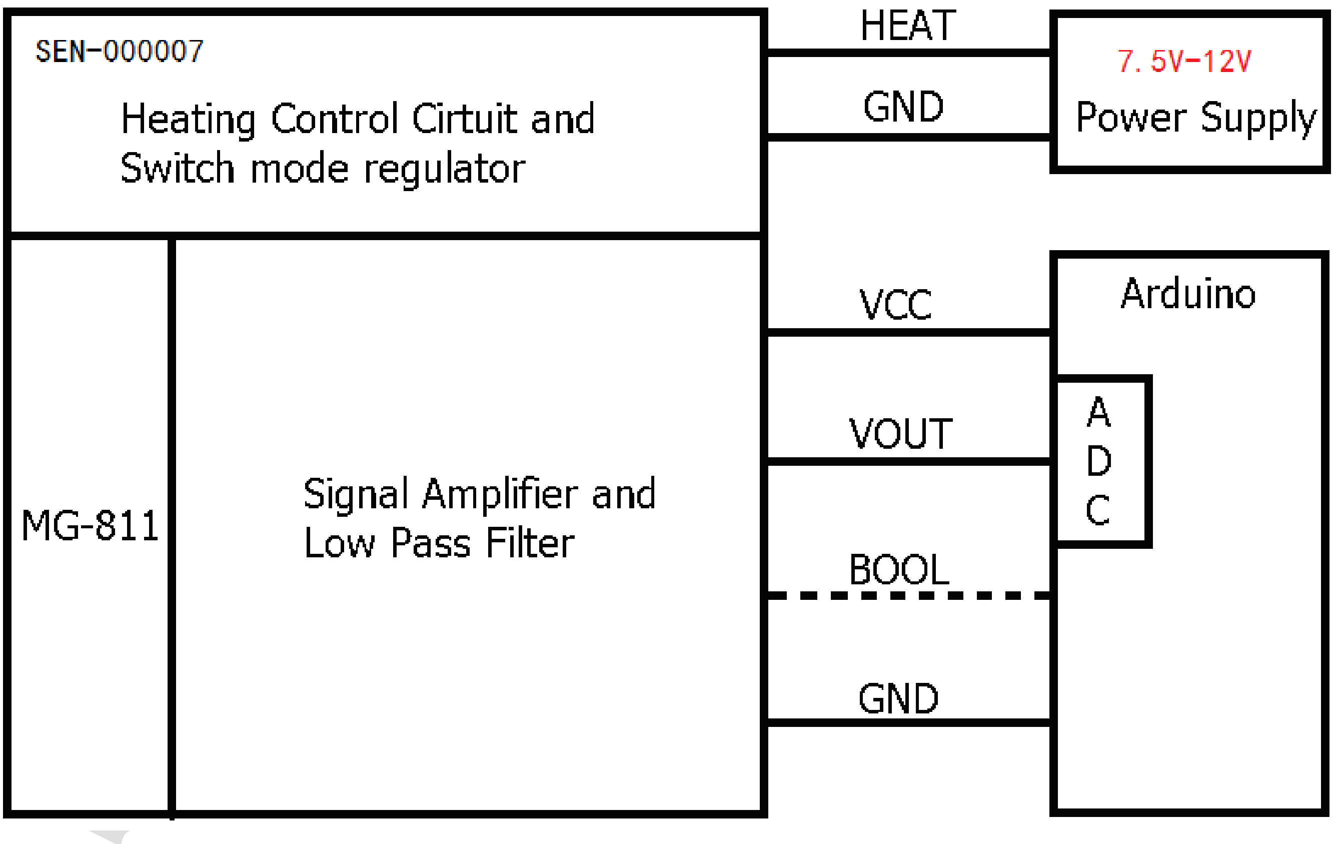

Figure 1-1, Typical Application Schematics

Operation

The MG-811 sensor is basically a cell which gives an output in the range of 100-600mV (400—10000ppm CO_2). The current sourcing capability of the cell is quite limited. The amplitude of the signal is so low and the output impedance of the cell is so high that a signal conditioning circuit is required between the sensor and microcontroller's ADC input. The output voltage of the sensor in clean air (typically 400ppm CO2) is in the range of 200mV-600mV, this output voltage is defined as Zero Point Voltage (V_0) which is the baseline voltage. The output voltage will decrease as the CO2 concentration increases. When the concentration of CO2 is greater than 400ppm, the output voltage (Vs) is linear to the common logarithm of the CO2 concentration (C_{CO2}):

$$Vs = V_0 + \Delta Vs / (\log_{10}400 - \log_{10}1000) * (\log_{10}C_{CO2} - \log_{10}400)$$

Where ΔVs = sensor output@400ppm – sensor output@1000ppm

Reaction Voltage(ΔVs) is the voltage drop from CO2 concentration of 400ppm to CO2 concentration of 1000ppm, which may differ from sensor to sensor. The typical value for ΔVs is 30mV-90mV. In order to get an accurate CO2 concentration result, proper calibration is required.

The DC gain of the signal conditioning circuit is 8.5. So the range of VOUT is 0.85-5.0V, which is a reasonable range for a 5V microcontroller or standalone ADC.

The threshold of the comparator open drain output pin BOOL can be set by on-board trimmer R11. When VOUT is lower than the threshold voltage the BOOL is at ground potential. When VOUT is greater than the preset value, the BOOL is open circuit. User should connect a pull-up resistor to the BOOL pin in order to have a valid "high" state.

Circuit Description

a. Signal Conditioning

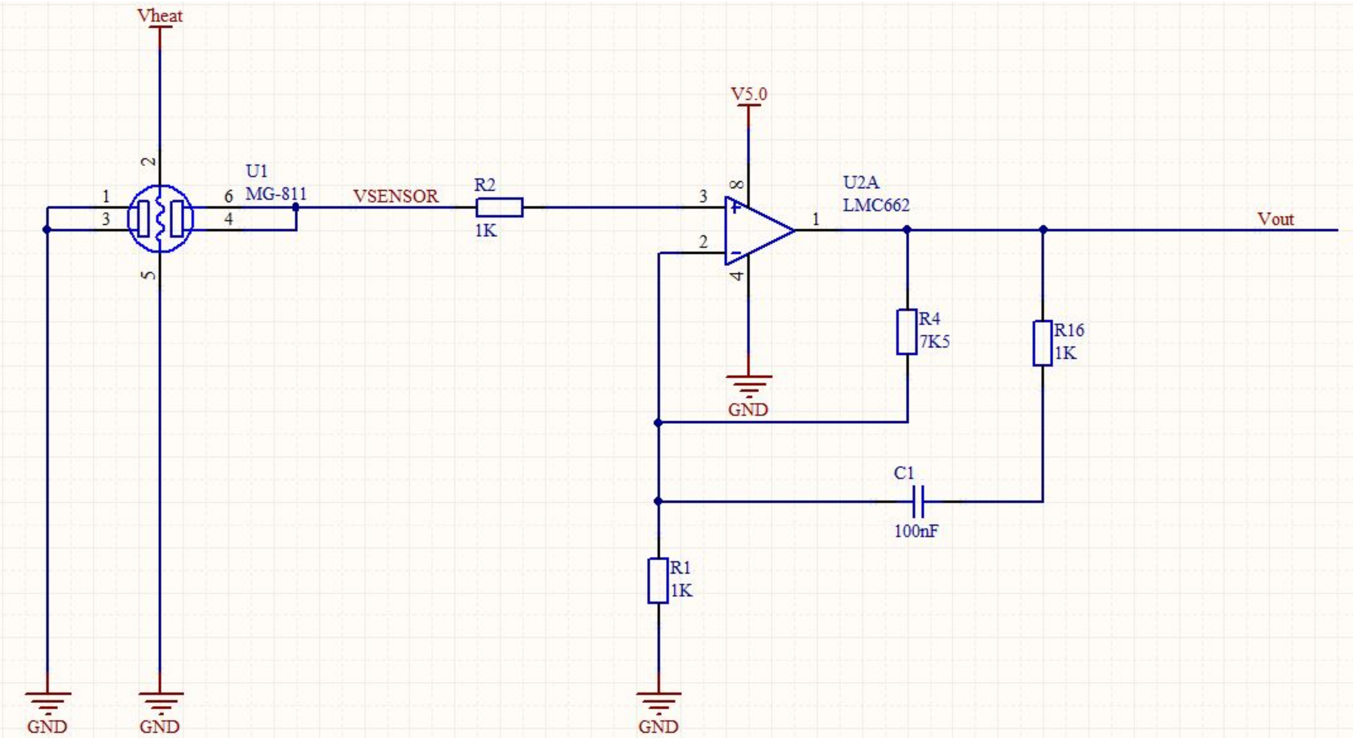

Figure 1-2, Signal Conditioning Circuit

The LMC662 is used as the amplifier because of its ultra high input impedance. According to the datasheet of MG-811, this sensor require an input impedance of 100-1000Gohm, the LMC662 has an input resistance above 1Tohm, which meets this requirement. The typical input offset voltage of this OPA is about 3mV, which is insignificant for this application. The DC gain is set by R4 and R1, with the formula

$$Vout = Vin * (1 + R4/R1)$$

In this specific application, Vout = 8.5*Vin.

R16 and C1 form a Low Pass Filter which gives a cleaner output by filtering out the high frequency noise.

b. Comparator

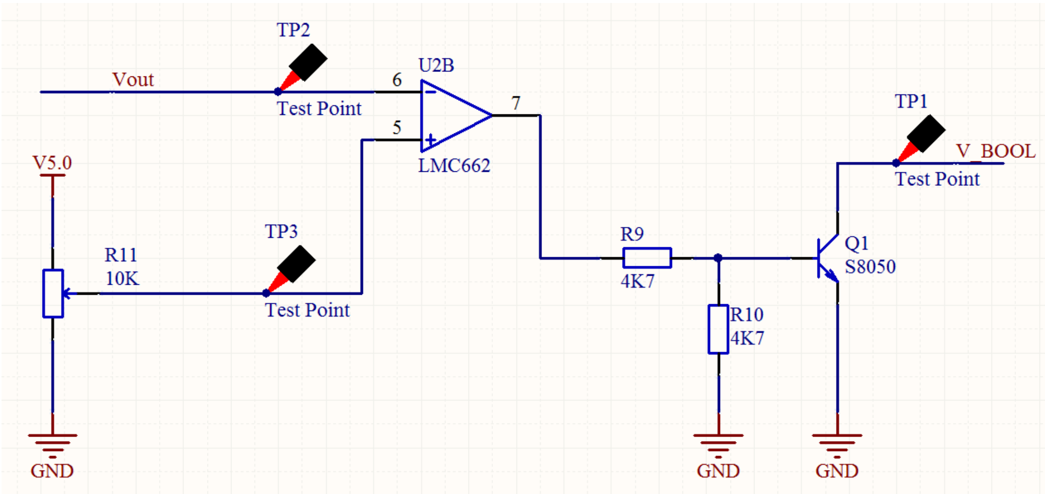

Figure 1-3, Open collector digital output circuit

The LMC662 is used as a comparator here. The R11 set the threshold of the comparator. If VOUT goes below the threshold, V_BOOL is at ground potential. If VOUT goes greater than the threshold, V_BOOL is floating. A pull-up resistor is needed to pull the BOOL pin up in order to have a valid "high" state when V_BOOL is floating.

c. Switch Mode Heating Voltage Regulator

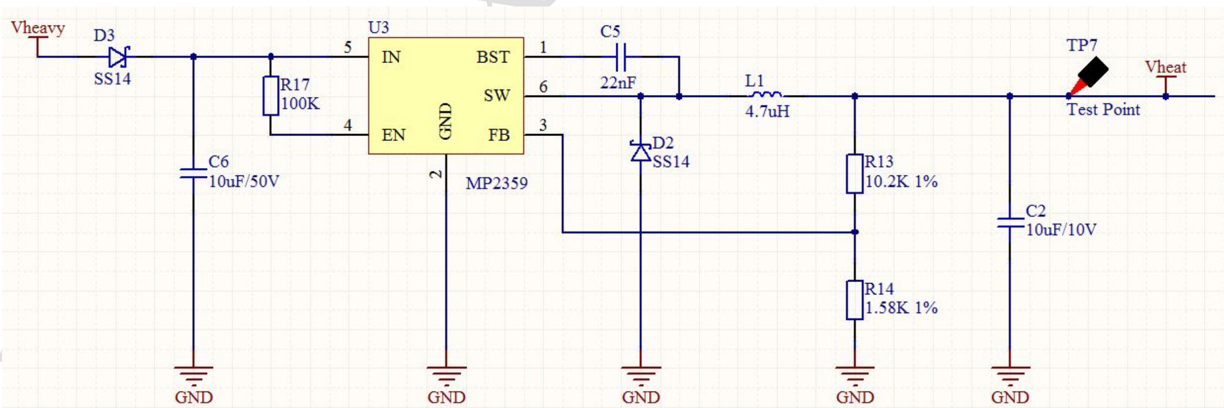

Figure 1-4, Sensor Heating Circuit

This is a typical step-down SMPS, the feedback voltage of the MP2359 is 0.81V, here is the relationship between VIN and VOUT of this circuit. This is not a low power device, so please don't use a 9V battery as the power source of the heating circuit. The battery will die very soon if you apply it to this circuit.

VOUT = 0.81 * (1 + R13/R14)

In this specific application, VOUT = 0.81 * (1 + 10.2K/1.58K) = 6.0V

Test Point Description

There are six test points on board. They are VE, AN, BL, TH, +V and GND.

VE – the regulated heating voltage, typical values are 6.0V
AN – analog output, the voltage should drop down when you puffing air to the sensor
BL – digital output, see "b. Comparator"
TH – comparator threshold voltage, you can set it to any value between 0 and +V
+V – signal conditioning circuit power supply, which is 5V

Assembly Drawing

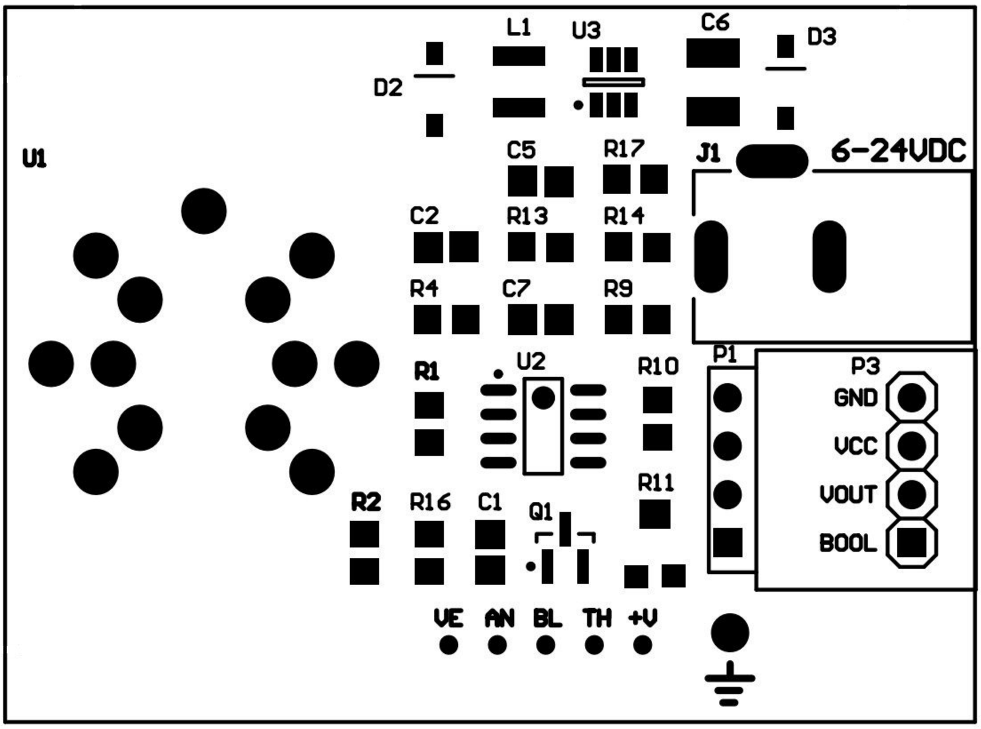

Figure 1-5, Assembly Drawing

Dimensions

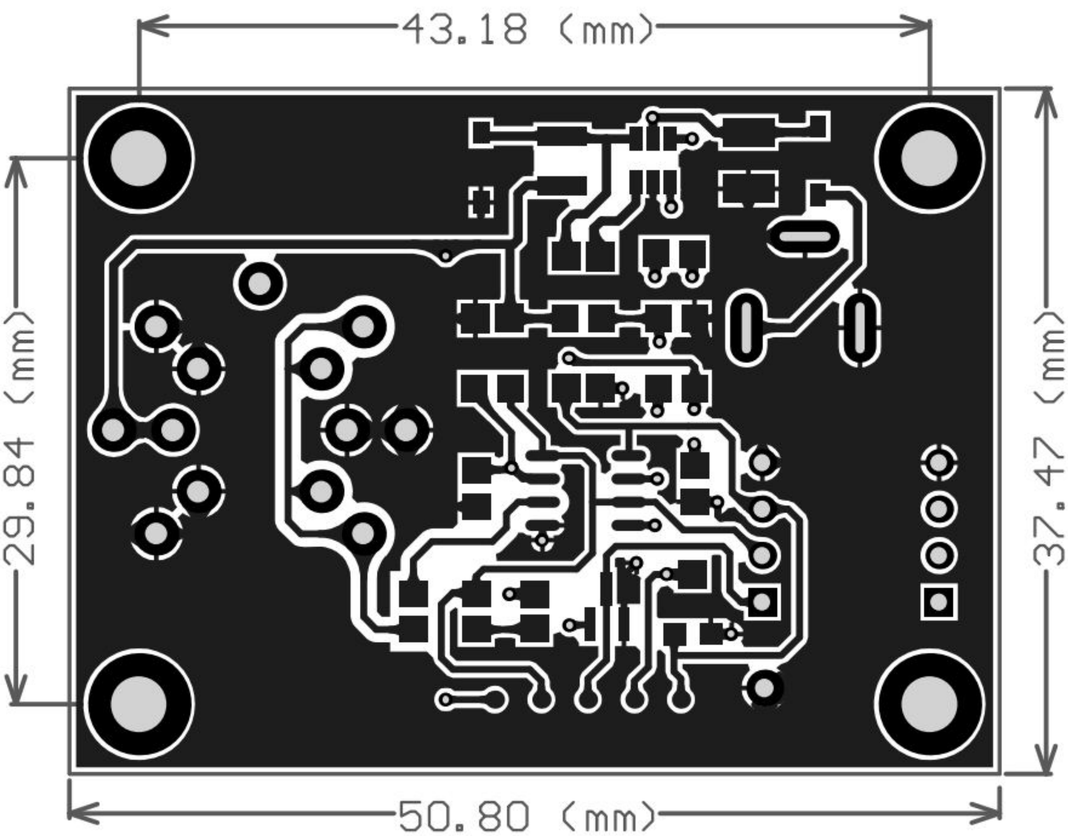

Figure 1-6, Outline Dimension

Sample Code for Arduino

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

Lisence: 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 demostration ONLY. More
       sophisticated calibration is required for industrial field application.

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

/************************Hardware Related Macros************************************/
#define         MG_PIN                       (0)     //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_VOLTGAE             (0.020) //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_VOLTGAE/(2.602-3))};
                                                     //two points are taken from the curve.
                                                     //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 Demostration\n");
}

void loop()
{
    int percentage;
    float volts;


    volts = MGRead(MG_PIN);
    Serial.print( "SEN-00007:" );
    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(200);
}
```

```
/*****************************  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  MGGetPercentage(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]);
  }
}
```

Demo Output

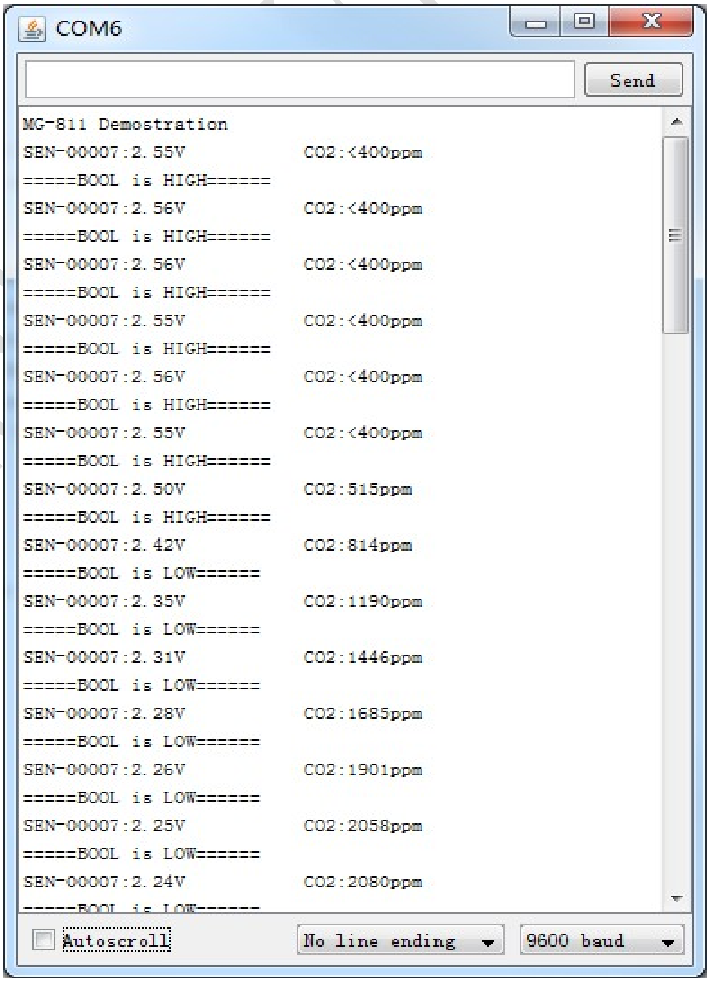

Figure 1-7 Demo output as puffing a small amount of breath to the sensor