Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Gravity: I2C BME680 Environmental Sensor (VOC, Temperature, Humidity, Barometer)

SKU: SEN0248

Introduction

DFRobot BME680 Environmental Sensor is a low power gas, pressure, temperature & humidity sensor based on BOSCH BME680 sensor. It is a 4-in-1 multi-functional MEMS environmental sensor which integrates VOC (Volatile Organic Compounds) sensor, temperature sensor, humidity sensor and barometer.

With DFRobot Gravity BME680 Environmental Sensor, you can monitor 4 environmental parameters simultaneously at the most. It is designed for air quality monitor, and due to the MEMS technology, BME680 has a small size and low power consumption. It can be widely used in environmental monitoring, home automation and control, Internet of Things (IoT) wearable device, GPS enhancement, etc.

DFRobot Gravity BME680 environmental sensor provides a Gravity I2C connector, plug & play, easy to connect. With onboard voltage regulator IC and level translator IC, DFRobot Gravity BME680 environmental sensor shows good compatibility. It can be directly connected to 3.3V and 5V systems. Moreover, there is also a SPI connector in reserve for further expansion projects.

With the development of industrialization, air pollution is getting worse and worse. Toxic chemical odors are even common with new furniture. These invisible killers are destroying your health day by day. You do need to concern about your health as soon as possible and DFRobot Gravity BME680 environmental sensor can help make an air quality monitor. DFRobot Gravity BME680, take care of your health!

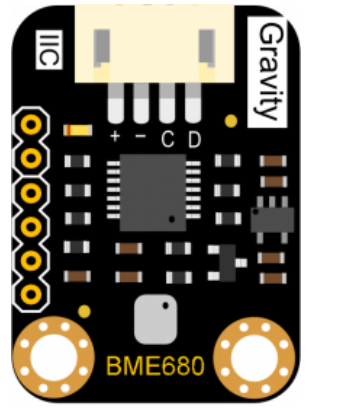
Specification

- Input Voltage: 3.3V~5.0V
- Operating Current: 5mA (25mA in VOC)
- Wire Connector: Gravity I2C
- Connector in Reserve: SPI
- Temperature Measurement Range: -40℃~+85℃
- Temperature Measurement Precision: ±1.0℃(0~65℃)
- Humidity Measurement Range: 0-100%r.H
- Humidity Measurement Precision: ±3%r.H.(20-80% r.H.,25℃)
- Atmospheric Pressure Measurement Range: 300-1100hPa
- Atmospheric Pressure Measurement Precision: ±0.6hPa(300-1100hPa,0~65℃)
- IAQ (Indoor Air Quality) Range: 0-500 (the larger the worse)
- Module Size: 30 × 22(mm) / 1.18 x0.87(inches)

IAQ (Indoor Air Quality) Sheet

IAQ Index	Air Quality
0 – 50	good[10]
51 – 100	average
101 – 150	little bad
151 – 200	bad
201 – 300	worse[2]
301 – 500	very bad

Board Overview



Num	Label	Description
+	VCC	Power Input(3.3~5.0V)
-	GND	Power Ground(0V)
C	SCL	I2C Clock Signal
D	SDA	I2C Data Signal

Tutorial

This tutorial will demonstrate how to use this sensor.
At present, only FireBeetle ESP8266 IOT Microcontroller can read IAQ, other controllers can not support.

Requirements

- **Hardware**
  DFRduino UNO R3 (or similar) x 1
  DFRobot Gravity BME680 environmental sensor x1
  Gravity 4Pin Sensor wire (or Dupont wires) x1

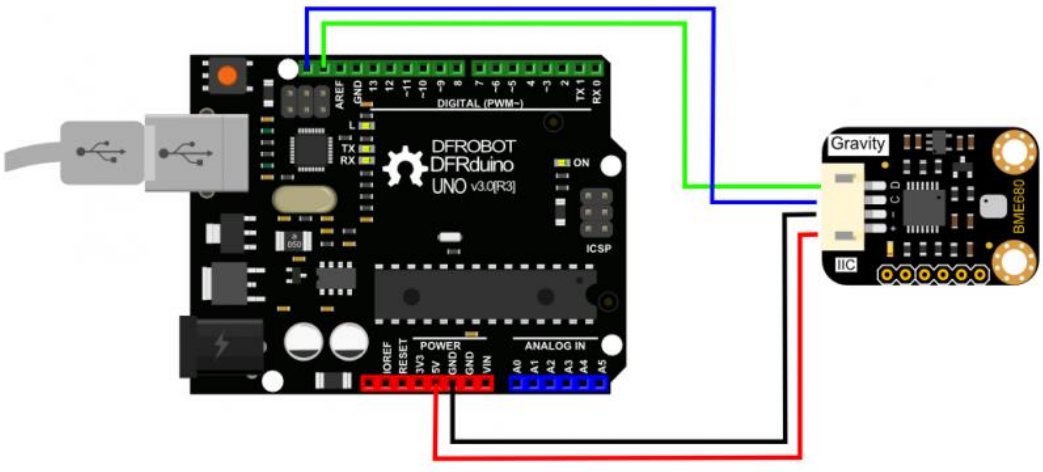
- **Software**
  Arduino IDE (V1.0.x or V1.8.x), Click to Download Arduino IDE from Arduino®
  Download and install the **BME680 Library**. How to install the library?

Connection Diagram

This product supports both IIC and SPI wiring connector. Please select suitable connector according to the wiring. Below is the connection diagram for your reference.
IIC wiring connector is recommended, plug & play, easy to use.

IIC Connection Diagram

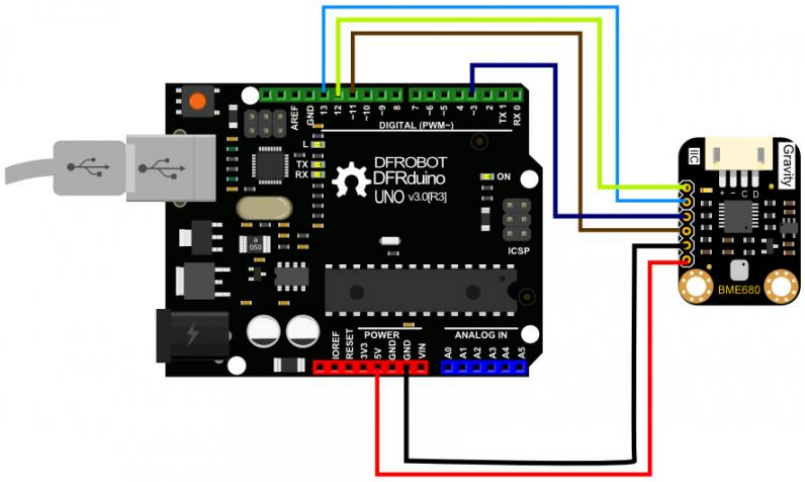
```
You must pay attention to the wiring order, VCC to Power Supply, GND to Gr
ound.
```



IIC Connection Diagram

SPI Connection Diagram

```
You must pay attention to the wiring order, VCC to Power Supply, GND to Gr
ound.
```



SPI Connection Diagram

Sample Code

Download and install the **BME680 Library**. How to install Libraries in Arduino IDE
This sample code is based on IIC connector. Please check file: DFRobot_BME680_SPI.ino for sample code for SPI connector in the library file. Because SPI sample code realizes the same function, it will not be shown at here.
Calibration is needed to monitor altitude value exactly. So that please fill in the sample code with correct local altitude value:

```
seaLevel = bme.readSeaLevel (your correct local altitude value)
```

Without IAQ

Program Function: read data from BME680 sensor and serial printing(without IAQ).

```
#include "DFRobot_BME680_I2C.h"
#include "Wire.h"
```

```
#include "SPI.h"


/*use an accurate altitude to calibrate sea level air pressure*/
#define CALIBRATE_PRESSURE


DFRobot_BME680_I2C bme(0x77);  //0x77 I2C address


float seaLevel;
void setup()
{
  Serial.begin(115200);
  while(!Serial);
  delay(1000);
  Serial.println();
  Serial.print(bme.begin());
  #ifdef CALIBRATE_PRESSURE
  bme.startConvert();
  delay(1000);
  bme.update();
  /*You can use an accurate altitude to calibrate sea level air pressure.
   *And then use this calibrated sea level pressure as a reference to obta
in the calibrated altitude.
   *In this case,525.0m is chendu accurate altitude.
   */
  seaLevel = bme.readSeaLevel(525.0);
  Serial.print("seaLevel :");
  Serial.println(seaLevel);
  #endif
}


void loop()
{
  bme.startConvert();
  delay(1000);
  bme.update();
```

```
  Serial.println();
  Serial.print("temperature(C) :");
  Serial.println(bme.readTemperature(), 2);
  Serial.print("pressure(Pa) :");
  Serial.println(bme.readPressure());
  Serial.print("humidity(%rh) :");
  Serial.println(bme.readHumidity(), 2);
  Serial.print("gas resistance(ohm) :");
  Serial.println(bme.readGasResistance());
  Serial.print("altitude(m) :");
  Serial.println(bme.readAltitude());
  #ifdef CALIBRATE_PRESSURE
  Serial.print("calibrated altitude(m) :");
  Serial.println(bme.readCalibratedAltitude(seaLevel));
  #endif
}
```

With IAQ

Program Function: read data from BME680 sensor and serial printing(with IAQ).
At present, only FireBeetle ESP8266 IOT Microcontroller can read IAQ, other controllers can not support. For FireBeetle ESP8266 IOT Microcontroller,please use Arduino IDE 1.8.x. Then update the SDK to 2.3.1 or above. Refer to section 4.2 of the FireBeetle ESP8266 Wiki for the tutorial.

```
#include "DFRobot_BME680_I2C.h"
#include "Wire.h"

/*use an accurate altitude to calibrate sea level air pressure*/
#define CALIBRATE_PRESSURE

DFRobot_BME680_I2C bme(0x77);  //0x77 I2C address



float seaLevel;
```

```
void setup()
{
  uint8_t       rslt = 1;
  Serial.begin(115200);
  while(!Serial);
  delay(1000);
  Serial.println();
  while(rslt != 0) {
    rslt = bme.begin();
    if(rslt != 0) {
      Serial.println("bme begin faild");
      delay(2000);
    }
  }
  Serial.println("bme begin successful");
  bme.supportIAQ();
}


void loop()
{
  static uint8_t       firstCalibrate = 0;


  #ifdef CALIBRATE_PRESSURE
  if(firstCalibrate == 0) {
    if(bme.iaqUpdate() == 0) {
      /*You can use an accurate altitude to calibrate sea level air pressu
re.
       *And then use this calibrated sea level pressure as a reference to
obtain the calibrated altitude.
       *In this case,525.0m is chendu accurate altitude.
       */
      seaLevel = bme.readSeaLevel(525.0);
      Serial.print("seaLevel :");
      Serial.println(seaLevel);
      firstCalibrate = 1;
```

```
    }
  }
  #else
    firstCalibrate = 1;
  #endif

  if(firstCalibrate) {
    uint8_t rslt = bme.iaqUpdate();
    if(rslt == 0) {
      Serial.println();
      Serial.print("time(ms) :");
      Serial.println(millis());
      Serial.print("temperature(C) :");
      Serial.println(bme.readTemperature(), 2);
      Serial.print("pressure(Pa) :");
      Serial.println(bme.readPressure());
      Serial.print("humidity(%rh) :");
      Serial.println(bme.readHumidity(), 2);
      Serial.print("altitude(m) :");
      Serial.println(bme.readAltitude());
#ifdef CALIBRATE_PRESSURE
      Serial.print("calibrated altitude(m) :");
      Serial.println(bme.readCalibratedAltitude(seaLevel));
#endif
      Serial.print("gas resistance :");
      Serial.println(bme.readGasResistance());
      if(bme.isIAQReady()) {
        Serial.print("IAQ :");
        float iaq = bme.readIAQ();
        Serial.print(iaq);
        if(iaq < 50) Serial.println(" good");
        else if(iaq < 100) Serial.println(" average");
        else if(iaq < 150) Serial.println(" little bad");
        else if(iaq < 200) Serial.println(" bad");
```

```
          else if(iaq < 300) Serial.println(" worse");
          else Serial.println(" very bad");
        } else Serial.println("IAQ not ready, please wait about 5 minutes");
      }
    }
}
```

Expected Results

Without IAQ

```
temperature(C) :22.00
pressure(Pa) :95945.00
humidity(%rh) :70.45
gas resistance(ohm) :13259.00
altitude(m) :470.61
calibrated altitude(m) :523.07

temperature(C) :22.01
pressure(Pa) :95946.00
humidity(%rh) :70.34
gas resistance(ohm) :13567.00
altitude(m) :470.52
calibrated altitude(m) :522.98

temperature(C) :22.03
pressure(Pa) :95943.00
humidity(%rh) :70.24
gas resistance(ohm) :13853.00
altitude(m) :470.78
calibrated altitude(m) :523.24

temperature(C) :22.04
pressure(Pa) :95944.00
humidity(%rh) :70.15
gas resistance(ohm) :14048.00
altitude(m) :470.69
calibrated altitude(m) :523.16
```

With IAQ

```
time(ms) :300003
temperature(C) :21.00
pressure(Pa) :96320.00
humidity(%rh) :37.00
altitude(m) :438.15
calibrated altitude(m) :521.95
gas resistance :8700.00
IAQ not ready, please wait about 5 minutes

time(ms) :302816
temperature(C) :21.00
pressure(Pa) :96320.00
humidity(%rh) :37.00
altitude(m) :438.15
calibrated altitude(m) :521.95
gas resistance :8629.00
IAQ :25.00 good

time(ms) :305630
temperature(C) :21.00
pressure(Pa) :96322.00
humidity(%rh) :37.00
altitude(m) :437.98
calibrated altitude(m) :521.78
gas resistance :8665.00
IAQ :21.00 good
```

Arduino Library Functions List

- Create a bme object and write to IIC address.

```
DFRobot_BME680_I2C bme(0x77);
```

- Initialize BME680 and library

```
begin();
```

- Start data converting

```
startConvert();
```

- Read converted data

```
update();
```

- support reading the IAQ

```
supportIAQ();
```

- Start data converting with IAQ

```
iaqUpdate();
```

- Query whether the IAQ conversion is complete. If Completed return 1, else return 0.

```
isIAQReady();
```

- Get temperature data, unit of ℃, precision of 0.01℃

```
readTemperature();
```

- Get atmosphere pressure, unit of pa, precision of 0.01pa

```
readPressure();
```

- Get humidity, unit of %rh, precision of 0.01%rh

```
readHumidity();
```

- Get resistance value of gas-resistance, unit of ohm, precision of 0.01ohm

```
readGasResistance();
```

- Get altitude, unit of m, precision of 0.01m

```
readAltitude();
```

- Get sea level atmosphere pressure reference value and send altitude data

```
readSeaLevel(float altitude);
```

- Get calibrated altitude value and send atmosphere pressure reference value

```
readCalibratedAltitude(float seaLevel);
```

Compatibility Test

MCU	Pass	Fail	Untested	Note
FireBeetle-Board328P	√			Do Not Support IAQ
FireBeetle-ESP32	√			Do Not Support IAQ
FireBeetle-ESP8266	√			Support IAQ
Leonardo	√			Do Not Support IAQ

FAQ

For any questions, advice or cool ideas to share, please visit the DFRobot Forum.

More Documents

- Schematic
- Layout
- BME680 Datasheet
- 74HC125PW Datasheet