



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



Barometric Pressure Sensor(SKU:TOY0058)



Contents

- [1 Introduction](#)
 - [1.1 Applications](#)
- [2 Specification](#)
- [3 Pin Definition](#)
- [4 Documents](#)
- [5 Characteristic Curve](#)
- [6 Sample](#)
 - [6.1 Sample Code](#)

Introduction

Nowadays, there are more and more navigation devices with GPS function available in the electronic market. But along with high precision micro pressure sensors came, a problem was solved that we cannot know the current altitude, even we can use GPS to determine the location of the equipment. Bosch Sensortec developed a smart, ultra-small size, high precision digital sensor BMP180 to fit high precision measurement and data acquisition for intelligence with high accuracy output of pressure (or height) and temperature measurement data.

Applications

- Temperature Monitoring
- Pressure Monitoring
- Altitude Monitoring

- 3D navigating in the complex indoor spaces(cooperate with accelerometer)

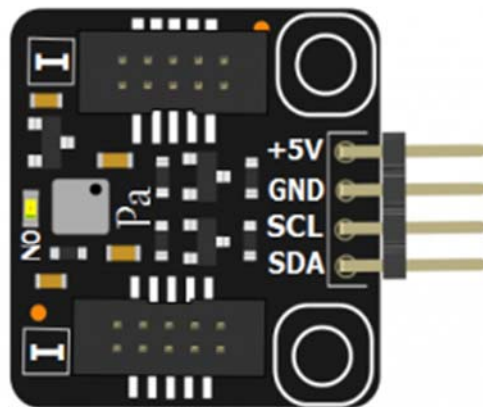
Specification

- Input voltage : 5V
- Operating temperature : -40~+85 °C
- measure accuracy : 0.12hPa/m
- Interface : IIC Digital Output x1

Pin Definition

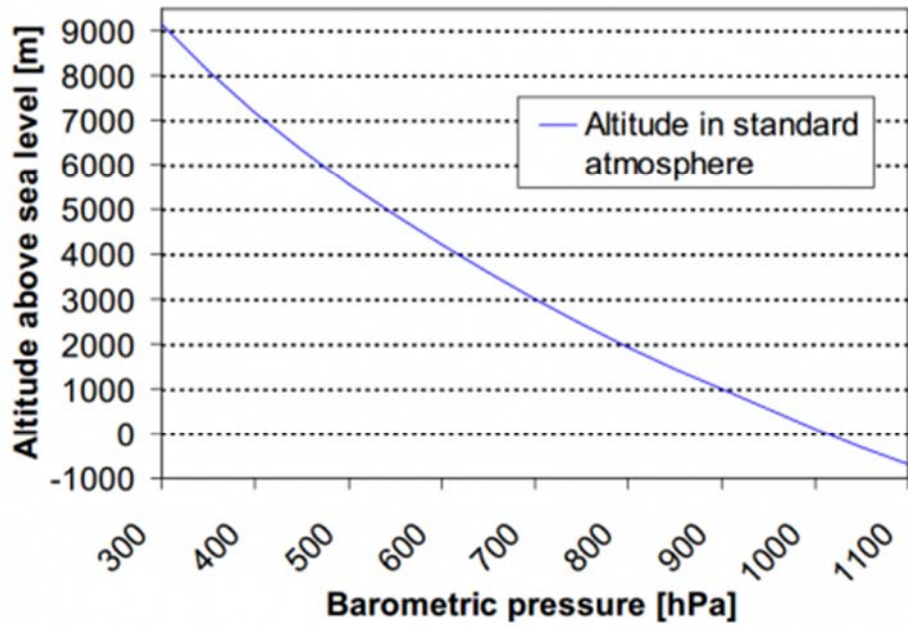
The definition of Barometric Pressure Sensor sensor pin is

1. +5V
2. GND
3. SCL
4. SDA



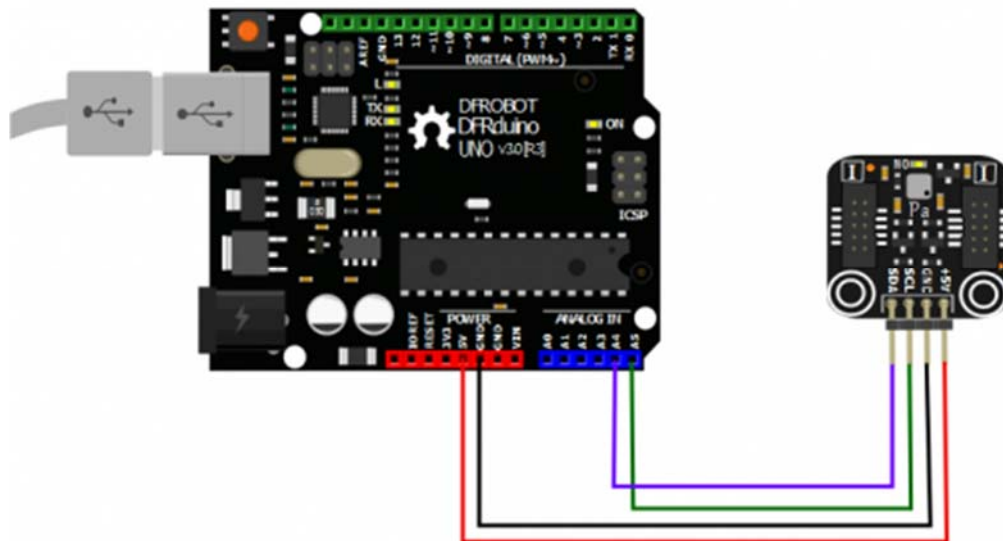
Pin Definition

Characteristic Curve

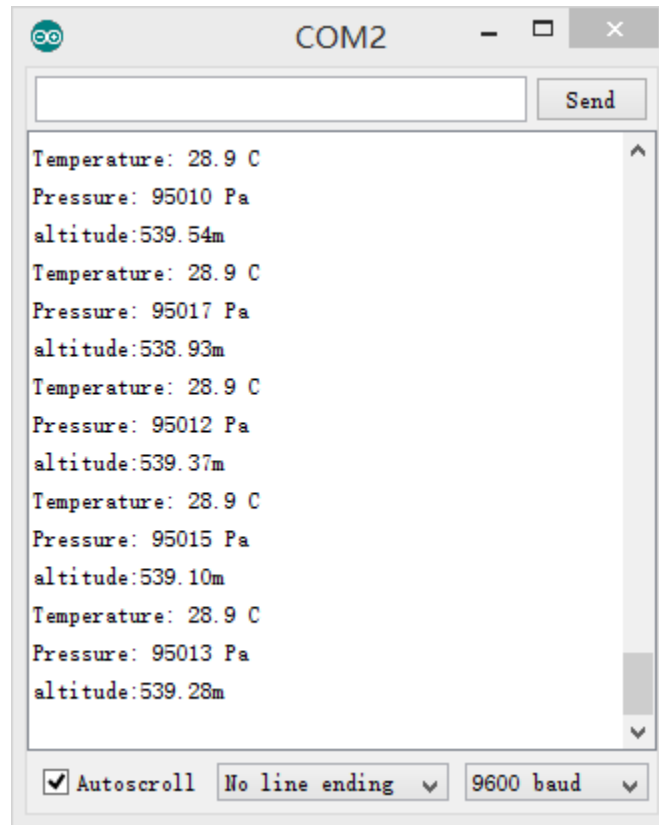


Transfer function: Altitude over sea level – Barometric pressure

Sample



TOY0058 diagram



The result in Serial Monitor

Sample Code

```
/*  
*****  
******/  
  
/******DFRobot.com******/  
  
/**write by Tom Riddler      Jun.16.14***/  
  
/**if you got any progream,please contact me terminatorfxy@hotmail.com***/  
  
#include <Wire.h>  
#define BMP180ADD 0xEE>>1 // I2C address of BMP180  
                          //write is (0xEE)      read is (0xEF)  
  
unsigned char OSS;  
  
/******MSB      LSB******/  
int acl;          // 0xAA      0xAB
```

```

int ac2;          // 0xAC    0xAD
int ac3;          // 0xAE    0xAE
unsigned int ac4; // 0xB0    0xB1
unsigned int ac5; // 0xB2    0xB3
unsigned int ac6; // 0xB4    0xB5
int b1;           // 0xB6    0xB7
int b2;           // 0xB8    0xB9
int mb;           // 0xBA    0xBB
int mc;           // 0xBC    0xBD
int md;           // 0xBE    0xBF

float temperature;
double pressure;
double pressure2;
long b5;
double altitude;

void setup()
{
  Serial.begin(9600);
  Wire.begin();

  OSS = 2; // Oversampling Setting           0: single    1: 2 times    2: 4
times    3: 8 times
  BMP180start();
}

void loop()
{
  calculate();
  show();
  delay(1000);
}

```

```

/** calculate centure **/
void calculate()
{
    temperature = bmp180GetTemperature(bmp180ReadUT());
    temperature = temperature*0.1;
    pressure = bmp180GetPressure(bmp180ReadUP());
    pressure2 = pressure/101325;
    pressure2 = pow(pressure2,0.19029496);
    altitude = 44330*(1-pressure2); //altitude = 443
    30*(1-(pressure/101325)^0.19029496);
}

/** print reslut **/
void show()
{
    Serial.print("Temperature: ");
    Serial.print(temperature, 1); //10 hexadecimal
    Serial.println(" C");
    Serial.print("Pressure: ");
    Serial.print(pressure, 0); //10 hexadecimal
    Serial.println(" Pa");
    Serial.print("altitude:");
    Serial.print(altitude);
    Serial.println("m");
}

/**BMP180 satrt program**/
void BMP180start()
{
    //MSB*/
    ac1 = bmp180ReadDate(0xAA); //get full data
    ac2 = bmp180ReadDate(0xAC);
    ac3 = bmp180ReadDate(0xAE);
    ac4 = bmp180ReadDate(0xB0);
    ac5 = bmp180ReadDate(0xB2);
}

```

```

    ac6 = bmp180ReadDate(0xB4);
    b1  = bmp180ReadDate(0xB6);
    b2  = bmp180ReadDate(0xB8);
    mb  = bmp180ReadDate(0xBA);
    mc  = bmp180ReadDate(0xBC);
    md  = bmp180ReadDate(0xBE);
}

/**BMP180 temperature Calculate***/
short bmp180GetTemperature(unsigned int ut)
{
    long x1, x2;
    x1 = ((long)ut - (long)ac6)*(long)ac5 >> 15; //x1=((ut-ac6)*ac5)/(2^15)
    x2 = ((long)mc << 11)/(x1 + md); //x2=(mc*2^11)/(x1+md)
    b5 = x1 + x2; //b5=x1+x2
    return ((b5 + 8)>>4); //t=(b5+8)/(2^4)
}

/**BMP180 pressure Calculate***/

long bmp180GetPressure(unsigned long up)
{
    long x1, x2, x3, b3, b6, p;
    unsigned long b4, b7;

    b6 = b5 - 4000;

    x1 = (b2 * (b6 * b6)>>12)>>11;
    x2 = (ac2 * b6)>>11;
    x3 = x1 + x2;
    b3 = (((((long)ac1)*4 + x3)<<OSS) + 2)>>2;

    x1 = (ac3 * b6)>>13;
    x2 = (b1 * ((b6 * b6)>>12))>>16;

```



```

x3 = ((x1 + x2) + 2)>>2;
b4 = (ac4 * (unsigned long)(x3 + 32768))>>15;

b7 = ((unsigned long)(up - b3) * (50000>>OSS));
if (b7 < 0x80000000)
    p = (b7<<1)/b4;
else
    p = (b7/b4)<<1;

x1 = (p>>8) * (p>>8);
x1 = (x1 * 3038)>>16;
x2 = (-7357 * p)>>16;
p += (x1 + x2 + 3791)>>4;

return p;
}

```

```

/**** Read 1 bytes from the BMP180 ****/

```

```

int bmp180Read(unsigned char address)
{
    unsigned char data;

    Wire.beginTransaction(BMP180ADD);
    Wire.write(address);
    Wire.endTransmission();

    Wire.requestFrom(BMP180ADD, 1);
    while(!Wire.available());

    return Wire.read();
}

```

```

/**** Read 2 bytes from the BMP180 ****/
int bmp180ReadDate(unsigned char address)
{
    unsigned char msb, lsb;
    Wire.beginTransaction(BMP180ADD);
    Wire.write(address);
    Wire.endTransmission();
    Wire.requestFrom(BMP180ADD, 2);
    while(Wire.available()<2);
    msb = Wire.read();
    lsb = Wire.read();
    return (int) msb<<8 | lsb;
}

/**** read uncompensated temperature value ****/
unsigned int bmp180ReadUT()
{
    unsigned int ut;
    Wire.beginTransaction(BMP180ADD);
    Wire.write(0xF4); // Write 0x2E into Register 0xF4
    Wire.write(0x2E); // This requests a temperature reading
    Wire.endTransmission();
    delay(5); // Wait at least 4.5ms
    ut = bmp180ReadDate(0xF6); // read MSB from 0xF6 read LSB from (16 bit)
    return ut;
}

/**** Read uncompensated pressure value from BMP180 ****/
unsigned long bmp180ReadUP()
{
    unsigned char msb, lsb, xlsb;
    unsigned long up = 0;

```

```

Wire.beginTransmission(BMP180ADD);
Wire.write(0xF4); // Write 0x34+(OSS<<6) into register 0xF4
Wire.write(0x34 + (OSS<<6)); // 0x34+oss*64
Wire.endTransmission();
delay(2 + (3<<OSS)); // Wait for conversion, delay time dependent on OSS

Wire.beginTransmission(BMP180ADD);
Wire.write(0xF6); // Read register 0xF6 (MSB), 0xF7 (LSB), and 0xF8 (XLSB)
Wire.endTransmission();

Wire.requestFrom(BMP180ADD, 3);
while(Wire.available() < 3); // Wait for data to become available
msb = Wire.read();
lsb = Wire.read();
xlsb = Wire.read();
up = (((unsigned long) msb << 16) | ((unsigned long) lsb << 8) | (unsigned long) xlsb) >> (8-OSS); //16 to 19 bit
return up;
}

```