

Measuring ambient pressure and temperature with the BMP280 sensor and Arduino Mega 2560

R. Armendariz
QCC physics, Sep 2023

Install the BMP280 library from the Arduino library manager

Wire 5V and GND to BMP280

Connect jumper wires from Arduino PWM pins 10, 11, 12, 13 to BMP280 pins CS, SDI, SDO, and SCK respectively

Open a new Arduino sketch

In the IDE go to FILE, EXAMPLES, Adafruit BMP280 Library, open up bmp280test.ino (this code runs in the Arduino IDE 2.0.3)

Towards the top of the code comment out "Adafruit_BMP280 bmp;" and uncomment "Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);"

i.e. run

```
//Adafruit_BMP280 bmp; // I2C
```

```
//Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
```

```
Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);
```

HERE IS THE CODE:

```
/******
```

```
This is a library for the BMP280 humidity, temperature & pressure sensor
```

```
Designed specifically to work with the Adafruit BMP280 Breakout
```

```
----> http://www.adafruit.com/products/2651
```

```
These sensors use I2C or SPI to communicate, 2 or 4 pins are required  
to interface.
```

```
Adafruit invests time and resources providing this open source code,  
please support Adafruit and open-source hardware by purchasing products  
from Adafruit!
```

```
Written by Limor Fried & Kevin Townsend for Adafruit Industries.
```

```
BSD license, all text above must be included in any redistribution
```

```
*****/
```

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_BMP280.h>
#define BMP_SCK (13)
#define BMP_MISO (12)
#define BMP_MOSI (11)
#define BMP_CS (10)
//Adafruit_BMP280 bmp; // I2C
//Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);

void setup() {
  Serial.begin(9600);
  while ( !Serial ) delay(100); // wait for native usb
  Serial.println(F("BMP280 test"));
  unsigned status;
  //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
  status = bmp.begin();
  if (!status) {
    Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
      "try a different address!"));
    Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);
    Serial.print("  ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");
    Serial.print("  ID of 0x56-0x58 represents a BMP 280,\n");
    Serial.print("  ID of 0x60 represents a BME 280.\n");
  }
}
```

```
Serial.print("    ID of 0x61 represents a BME 680.\n");
    while (1) delay(10);
}
/* Default settings from datasheet. */
bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */
    Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */
    Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */
    Adafruit_BMP280::FILTER_X16, /* Filtering. */
    Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */
}
void loop() {
    Serial.print(F("Temperature = "));
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");
    Serial.print(F("Pressure = "));
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");
    Serial.print(F("Approx altitude = "));
    Serial.print(bmp.readAltitude(1013.25)); /* Adjusted to local forecast! */
    Serial.println(" m");
    Serial.println();
    delay(2000);
}
```