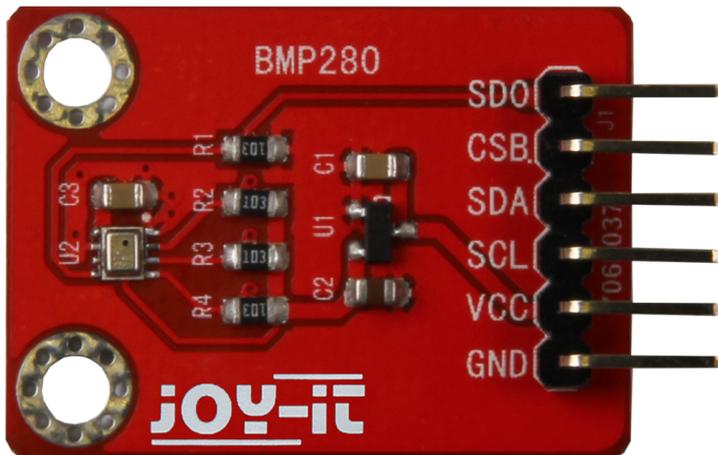


KY-052 Pressure-sensor / Temperature-sensor (BMP280)

Contents

1 Picture	1
2 Technical data / Short description	1
3 Pinout	2
4 Software example Arduino	2
5 Software example Raspberry Pi	3

Picture

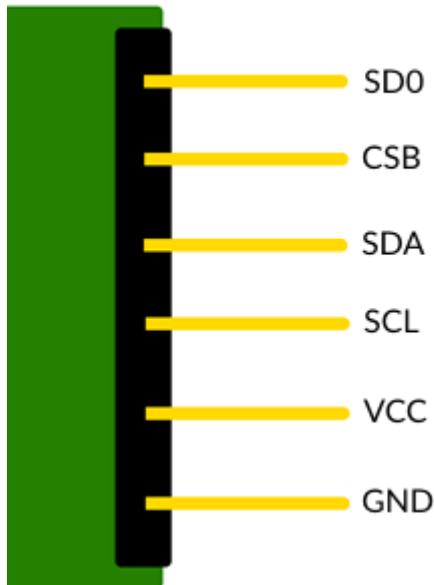


Technical data / Short description

This pressure sensor measures the air pressure at the sensor output and sends the result encoded to the I2C-bus.

You will need additional software for this sensor

Pinout



- *You can connect this sensor to 5V systems and to 3,3V systems. **Please pay attention that only one of the power supply pins is connected to the desired voltage** - you can get additional informations at the example below: [Connection Arduino \(5V\)](#) and [Connection Raspberry-Pi \(3,3V\)](#).

Software example Arduino

An additional Arduino library is needed to use this sensor. This library, for the BMP280, was written by Kevin Townsend (KTOWN) and was produced for Adafruit. It is available here: [Adafruit BMP280 Library](#)

The example below is using this library - we advise to download the library from Github, decompress it and copy it into the Arduino-library folder which can be found under the path (C:\user\[username]\documents\arduino\libraries).

```
//Initialise Libraries
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
//Define PIN-Wiring
#define BMP_SCK 13
#define BMP_MISO 12
#define BMP_MOSI 11
#define BMP_CS 10

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

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

KY-052 Pressure-sensor / Temperature-sensor (BMP280)

```
if (!bmp.begin()) {  
  Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));  
  while (1);  
}  
}  
  
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.println();  
  delay(2000);  
}
```

Example program download:[KY-052_BMP280_ARD](#)**Connections Arduino:**

VCC	=	[Pin 5V]
GND	=	[Pin GND]
SCL	=	[Pin 13]
SDA	=	[Pin 11]
SD0	=	[Pin 12]
CSB	=	[Pin 10]

Software example Raspberry Pi

To run our example code, you need to activate some modules and install an additional library first.

The library was written by Adafruit and updated for BMP280 support by "bastienwartz". This library is published under the MIT license and is available here: [BMP280 Library for Raspberry Pi](#)

It has to be installed first. But at the very beginning, you need to activate I2C in your settings.

You can do this by entering this in your command line:

```
sudo raspi-config
```

Simply navigate to no. 5 (Interfacing Options) and enable I2C. Restart your Raspberry after doing this.

Second, you have to install the github software on your Raspberry-Pi

```
sudo apt-get install git
```

For this you have to make sure that the Raspberry Pi has a connection to the internet. You can download and decompress the latest version of the Adafruit_BMP280 library with the command...

KY-052 Pressure-sensor / Temperature-sensor (BMP280)

```
git clone https://github.com/bastienwartz/Adafruit_Python_BMP
```

After that we can jump into the downloaded folder with the command...

```
cd Adafruit_Python_BMP/
```

... and install the library with

```
sudo python setup.py install
```

Now you are ready to use the library.

In order that the Raspberry Pi and the sensor can communicate via I2C-bus, you have to start the I2C-function of the Raspberry Pi first.

To do that you have to add the following line to the end of the file `"/boot/config.txt`:

```
dtoverlay=i2c-arms-on
```

You can edit the file with the following command:

```
sudo nano /boot/config.txt
```

You can save and close the file with the key sequence `[ctrl + x -> y -> enter]`.

Also you need additional libraries if you want to use I2C in with python. To install it, you have to enter the following command to the console:

```
sudo apt-get install python-smbus i2c-tools -y
```

After that you can use the following python code example. The program starts the measurement for air pressure, temperature and the altitude above sea level.

```
import Adafruit_BMP.BMP280 as BMP280
sensor = BMP280.BMP280(address=0x76)

print 'Temp = {0:0.2f} *C'.format(sensor.read_temperature())
print 'Pressure = {0:0.2f} Pa'.format(sensor.read_pressure())
print 'Altitude = {0:0.2f} m'.format(sensor.read_altitude())
print 'Sealevel Pressure = {0:0.2f} Pa'.format(sensor.read_sealevel_pressure())
```

Connections Raspberry Pi:

VCC	= -	[N.C]
GND	= GND	[Pin 06]
SCL	= GPIO03 / SCL	[Pin 05]

KY-052 Pressure-sensor / Temperature-sensor (BMP280)

```
SDA = GPIO02 / SDA [Pin 03]
3.3 = 3,3V [Pin 01]
```

Example program download[KY-052_BMP280_RPi](#)

To start, enter the command:

```
sudo python KY-052_BMP280_RPi.py
```

If your example is not working, you might need to change the I2C-Address in the example-code.

In order to do this, after wiring the sensor, check the I2C-Address by entering:

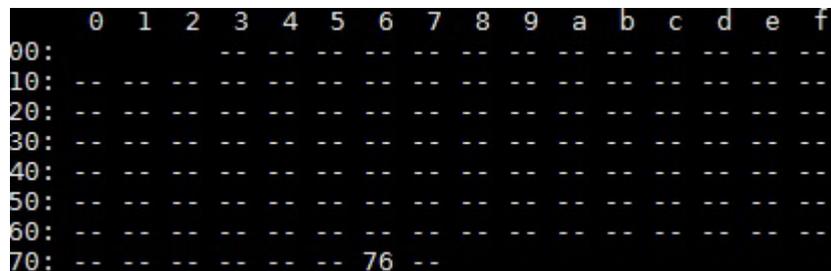
```
sudo i2cdetect -y 1
```

for newer Raspberry Pis or

```
sudo i2cdetect -y 1
```

for older Raspberry Pis.

The result should look like this:



```
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  76  --  --  --  --  --  --  --  --
```

The displayed number (in this case: 76) is your I2C-Address. The I2C-Address 76 is configured as default in our example. If your address is different, enter the example code by entering

```
sudo nano KY-052_BMP280_RPi.py
```

and update the following line to your address.