Using BMP180 With Raspberry Pi

HCMUTE

SVTH:

Lê Anh Cường – 22146279

Nguyễn Quang Dũng – 22146288

Hoàng Minh Đức – 22146295

Contents

1. Overview:

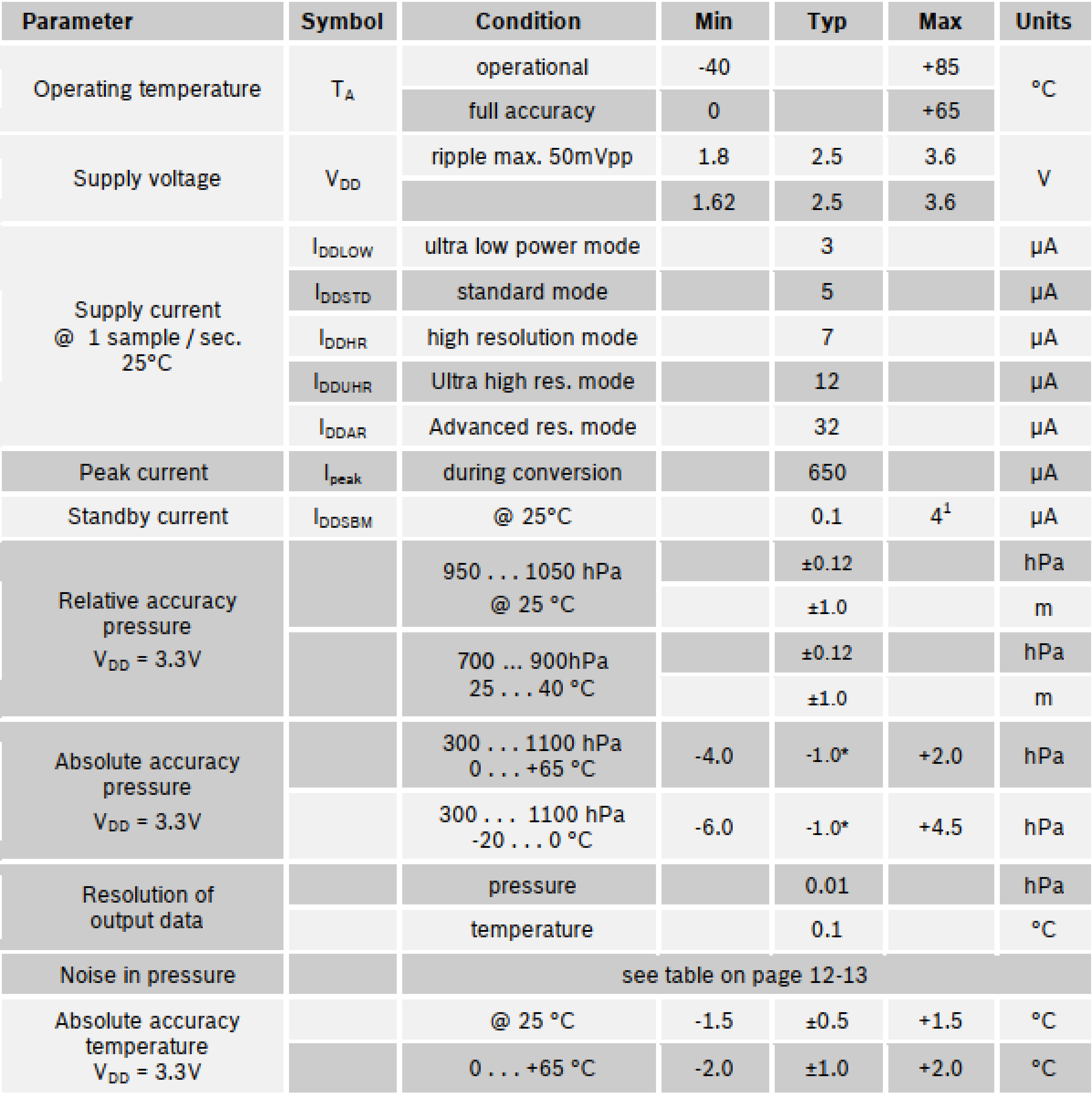
The BMP180 is a high-precision degital barometric pressure and temperature sensor develop by Bosch. It communicate using the I2C protocol, making it ideal for integration with microcontrollers and singles-board computers like the Raspberry Pi.

In this project, you will learn how to connect the BMP180 sensor to a raspberry Pi and use a custom driver to read real-time temperature and atmospheric pressure data. This guide will walk you through the necessary setup steps, wiring instructions, using the driver library, and troubleshooting any issues you may encounter.

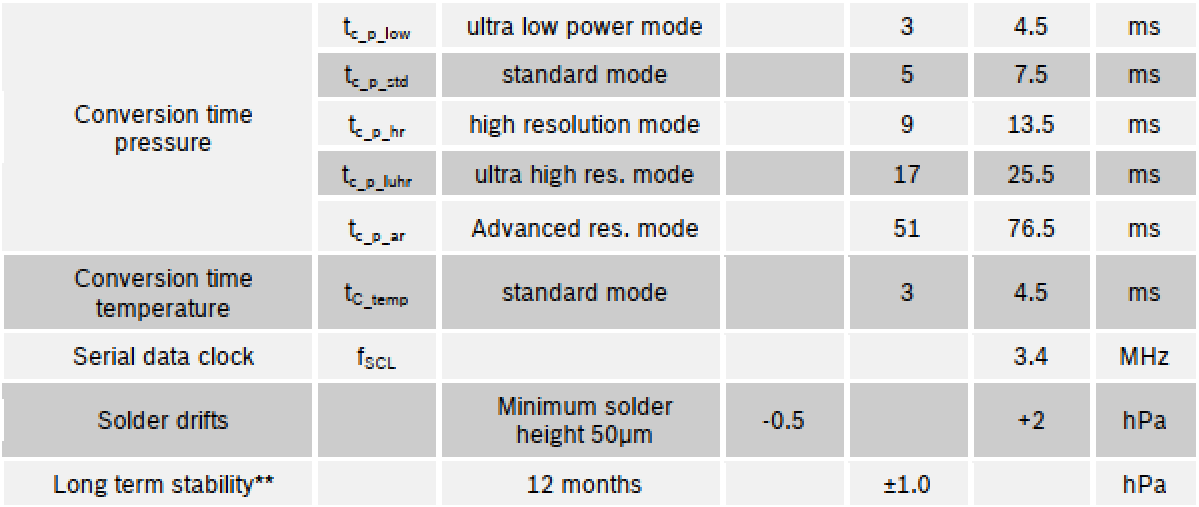
Whether you're building a weather station , a drone altimeter or a smart IoT device, understanding how to interface with BMP180 is a great place to start.

Sensor BMP180

2. Electrical Characteristics:



Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature	T_A	operational	-40		+85	°C
		full accuracy	0		+65	
Supply voltage	V_{DD}	ripple max. 50mVpp	1.8	2.5	3.6	V
			1.62	2.5	3.6	
Supply current @ 1 sample / sec. 25°C	I_{DDLOW}	ultra low power mode		3		µA
	I_{DDSTD}	standard mode		5		µA
	I_{DDHR}	high resolution mode		7		µA
	I_{DDUHR}	Ultra high res. mode		12		µA
	I_{DDAR}	Advanced res. mode		32		µA
Peak current	I_{peak}	during conversion		650		µA
Standby current	I_{DDSBM}	@ 25°C		0.1	4[1]	µA
Relative accuracy pressure V_{DD} = 3.3V		950 . . . 1050 hPa @ 25 °C		±0.12		hPa
				±1.0		m
		700 ... 900hPa 25 . . . 40 °C		±0.12		hPa
				±1.0		m
Absolute accuracy pressure V_{DD} = 3.3V		300 . . . 1100 hPa 0 . . . +65 °C	-4.0	-1.0*	+2.0	hPa
		300 . . . 1100 hPa -20 . . . 0 °C	-6.0	-1.0*	+4.5	hPa
Resolution of output data		pressure		0.01		hPa
		temperature		0.1		°C
Noise in pressure		see table on page 12-13				
Absolute accuracy temperature V_{DD} = 3.3V		@ 25 °C	-1.5	±0.5	+1.5	°C
		0 . . . +65 °C	-2.0	±1.0	+2.0	°C



Conversion time pressure	$t_{c\_p\_low}$	ultra low power mode		3	4.5	ms
	$t_{c\_p\_std}$	standard mode		5	7.5	ms
	$t_{c\_p\_hr}$	high resolution mode		9	13.5	ms
	$t_{c\_p\_luhr}$	ultra high res. mode		17	25.5	ms
	$t_{c\_p\_ar}$	Advanced res. mode		51	76.5	ms
Conversion time temperature	t_{C_temp}	standard mode		3	4.5	ms
Serial data clock	f_{SCL}				3.4	MHz
Solder drifts		Minimum solder height 50µm	-0.5		+2	hPa
Long term stability**		12 months		±1.0		hPa

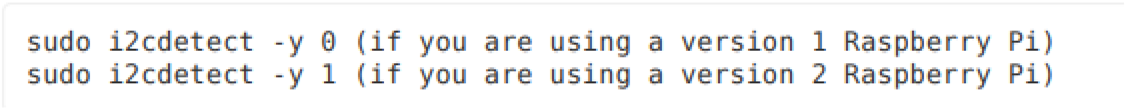
3. Configuring for I2C:

Before you can get started with I2C on the Pi you'll need to run through quick steps from the console.

Check out this tutorial for more details and follow it completely:

https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c

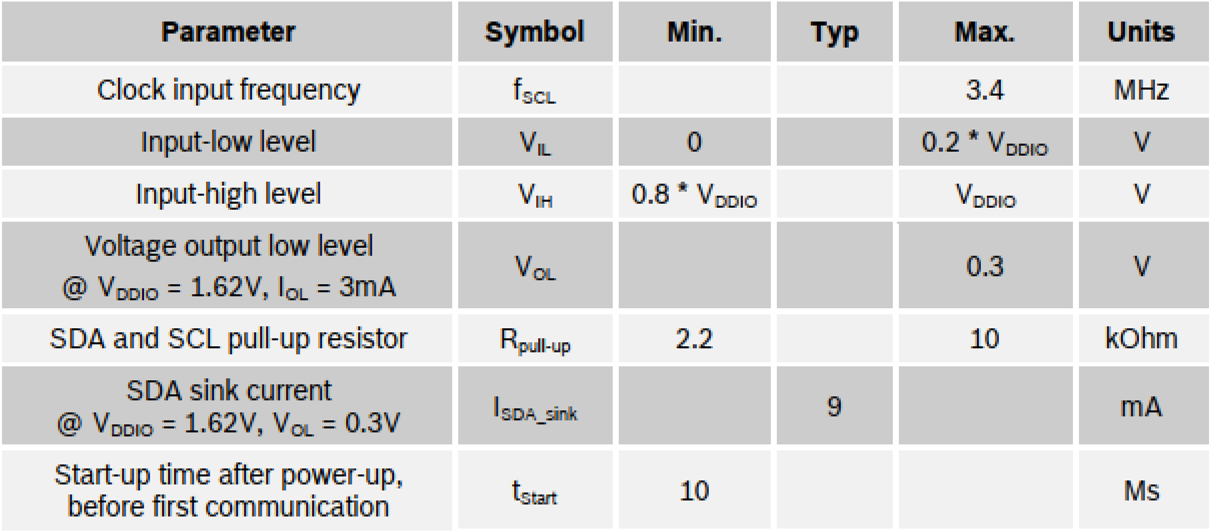
When you're done, run:



```
sudo i2cdetect -y 0 (if you are using a version 1 Raspberry Pi)
sudo i2cdetect -y 1 (if you are using a version 2 Raspberry Pi)
```

This will search /dev/i2c-1 for all address.

- **I2C Specification:**



Parameter	Symbol	Min.	Typ	Max.	Units
Clock input frequency	f_{SCL}			3.4	MHz
Input-low level	V_{IL}	0		$0.2 * V_{DDIO}$	V
Input-high level	V_{IH}	$0.8 * V_{DDIO}$		V_{DDIO}	V
Voltage output low level @ $V_{DDIO} = 1.62V$, $I_{OL} = 3mA$	V_{OL}			0.3	V
SDA and SCL pull-up resistor	$R_{pull-up}$	2.2		10	kOhm
SDA sink current @ $V_{DDIO} = 1.62V$, $V_{OL} = 0.3V$	I_{SDA_sink}		9		mA
Start-up time after power-up, before first communication	t_{Start}	10			Ms

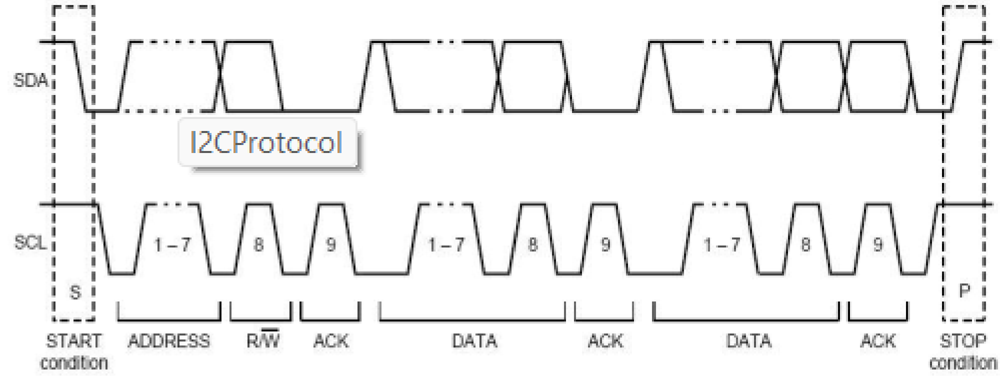
- **I2C communication:**

The I2C bus uses two signals: SDA(data) and SCL(clock). A Start condition(S) occurs when SDA falls while SCL is high and A Stop condition(P) occurs when SDA rises while SCL is high.

After the master sends 7 address bits + R/W bit:

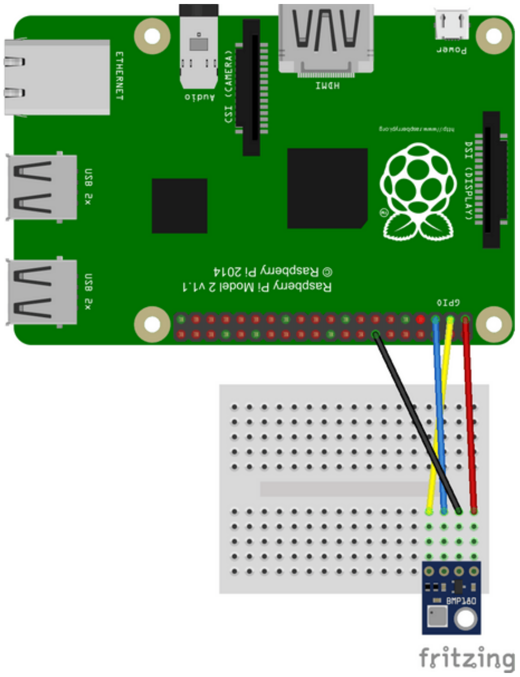
The BMP180 acknowledges by pull SDA low on the 9th clock cycle (ACK)

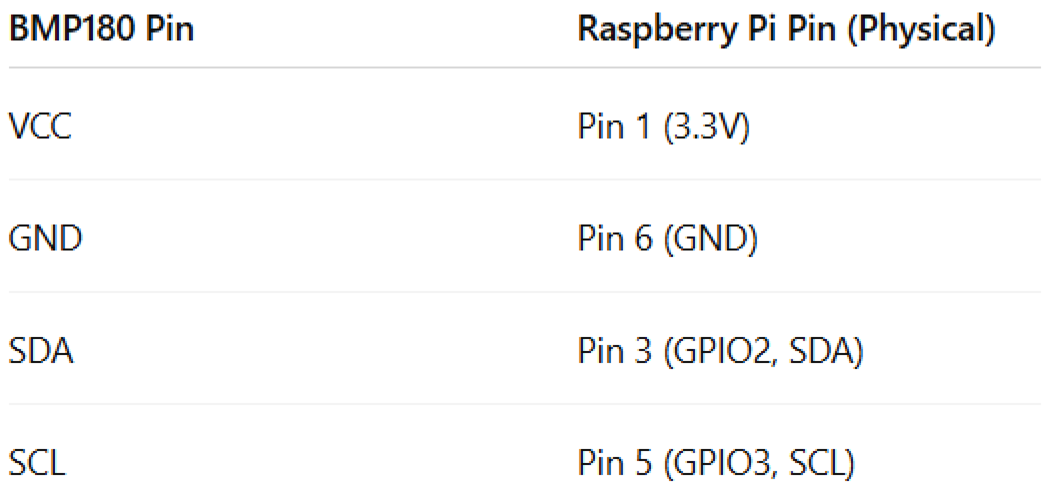
Data must remain stable while SCL is high, changes are allowed only when SCL is low.


SDA
I2CProtocol
SCL
1 – 7
8
9
1 – 7
8
9
1 – 7
8
9
S
P
START condition
ADDRESS
R/$\overline{W}$
ACK
DATA
ACK
DATA
ACK
STOP condition

4. Hooking up the BMP180:

Provide wiring instructions and pin mapping between the BMP180 and Raspberry Pi.





BMP180 Pin	Raspberry Pi Pin (Physical)
VCC	Pin 1 (3.3V)
GND	Pin 6 (GND)
SDA	Pin 3 (GPIO2, SDA)
SCL	Pin 5 (GPIO3, SCL)

5. How to Build and Use the BMP180 Driver:

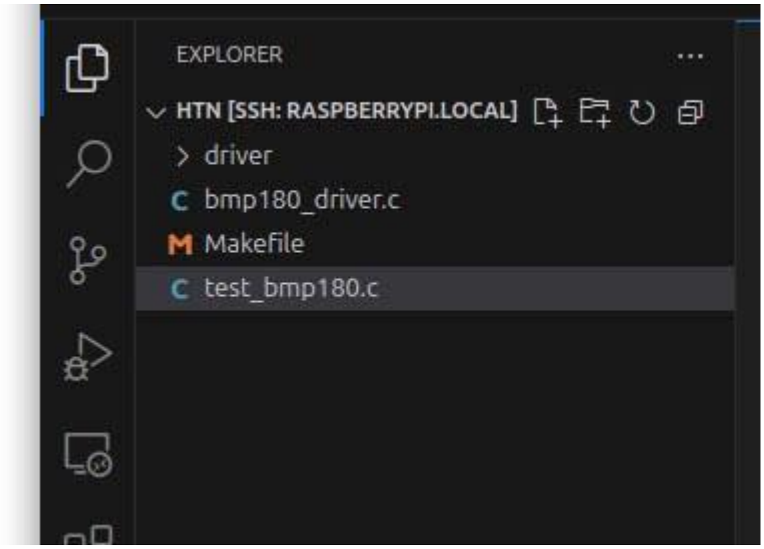
- **Check BMP180 Sensor Connection**

Verify the sensor connection:

```
i2cdetect -y 1
```

You should see the address '0x77' in the table if the sensor is properly connected.

- **Directory Structure:**



This folder contains the source file, test file, and build instructions.

Example Makefile:

```
all: gcc -o test_bmp180 test_bmp180.c bmp180_driver.c -lwiringPi
```

To build the project, run:

```
make
```

To run the program:

```
gcc test_bmp180.c -o test_bmp180.c

sudo ./test
```

- **Kernel Module Loading with insmod:**

The following command is used to load a kernel module (.ko file) into the Linux kernel:

-
  ```
  sudo insmod bmp180_driver.ko
  ```