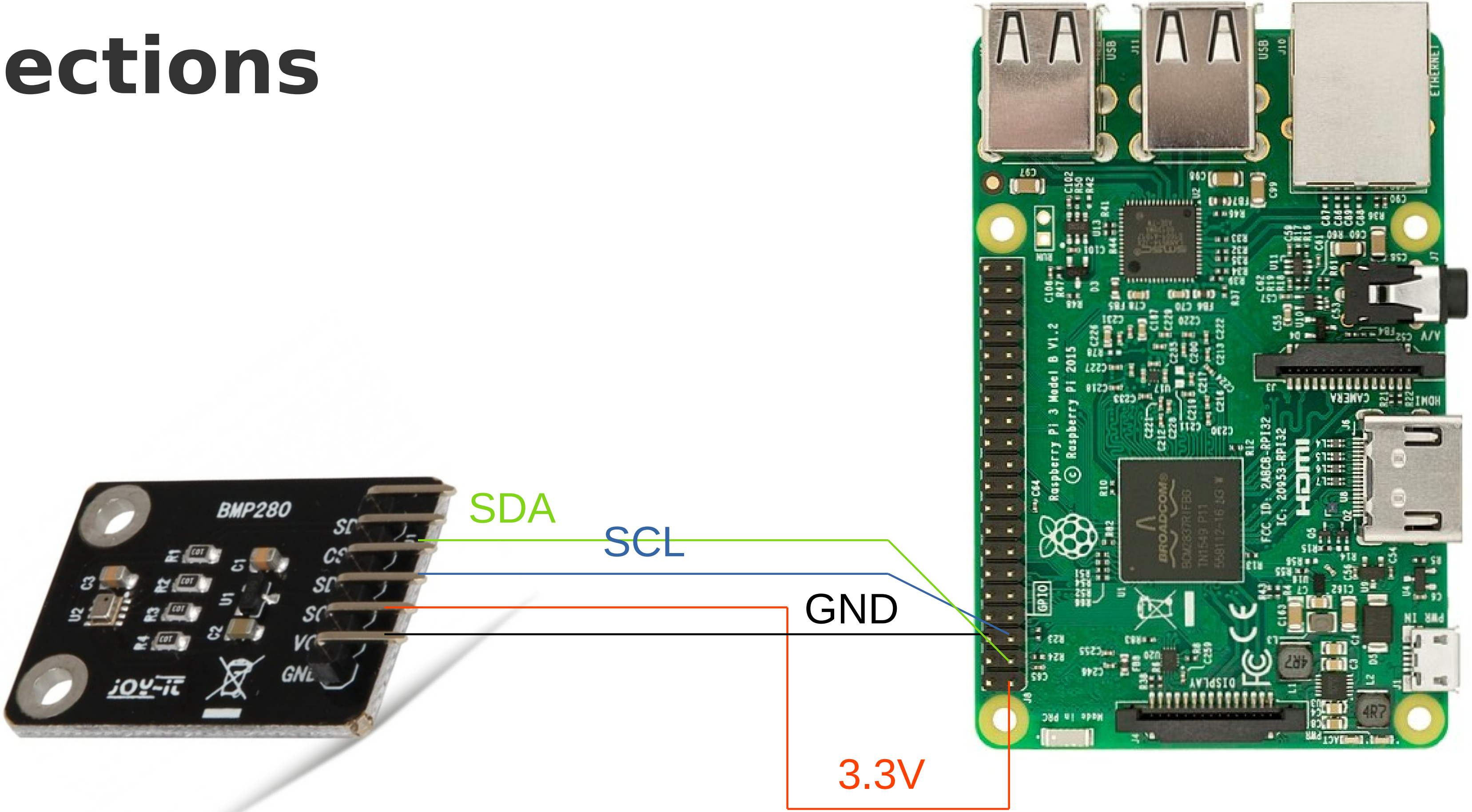
BMP280 Pressure & Temperature Sensor with Python

Controll over I²C

Agenda

1. How to connect the Sensor?
2. What’s behind the memory map?
3. Writing a simple Python Script to read the temperature
4. Demonstration of the script
5. What’s about the pressure?

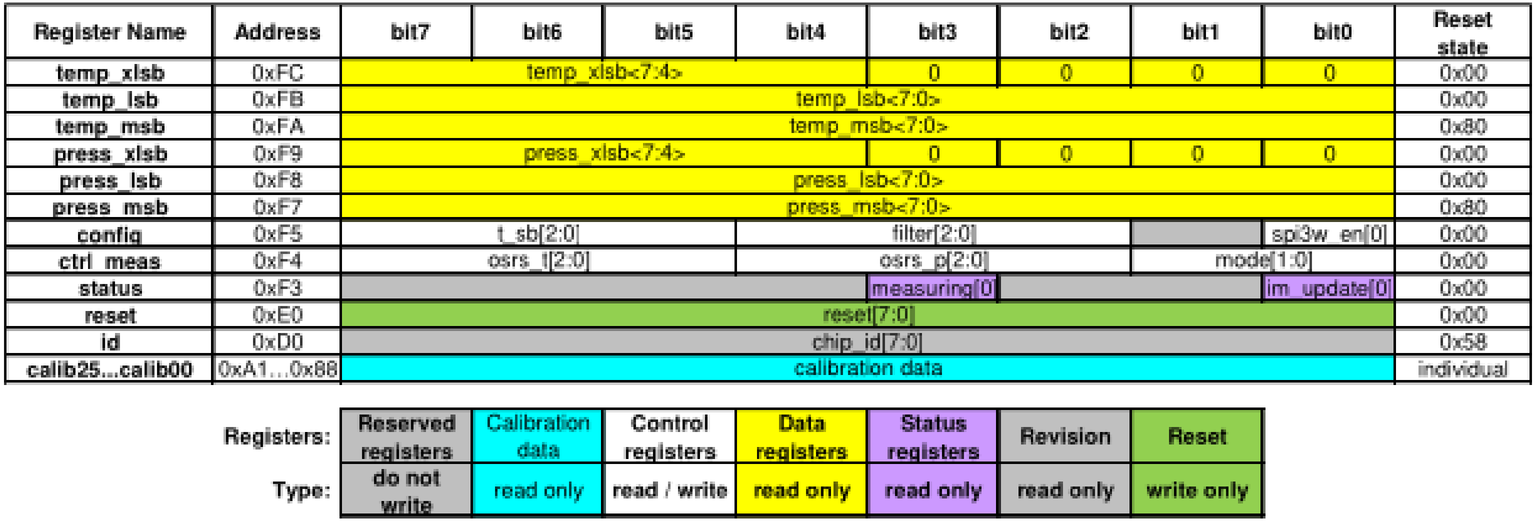
Connections



Source BMP280: https://www.pollin.de/p/joy-it-bmp-280-druck-und-temperatursensor-810918
Source Raspberry Pi: https://commons.wikimedia.org/wiki/File:Raspberry-Pi-3-Flat-Top.jpg

BMP280 Memory Map

Table 18: Memory map



Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00
press_msb	0xF7	press_msb<7:0>								0x80
config	0xF5	t_sb[2:0]			filter[2:0]				spi3w_en[0]	0x00
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00
status	0xF3					measuring[0]			im_update[0]	0x00
reset	0xE0	reset[7:0]								0x00
id	0xD0	chip_id[7:0]								0x58
calib25...calib00	0xA1...0x88	calibration data								individual

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Revision	Reset
Type:	do not write	read only	read / write	read only	read only	read only	write only

Settings in config register 0xF5

Register 0xF5	Name	Description
Bit 0	spi3_wen	Enables 3-wire SPI interface when set to ‘1’.

→ We use I²C: Value: 0

Settings in config register 0xF5

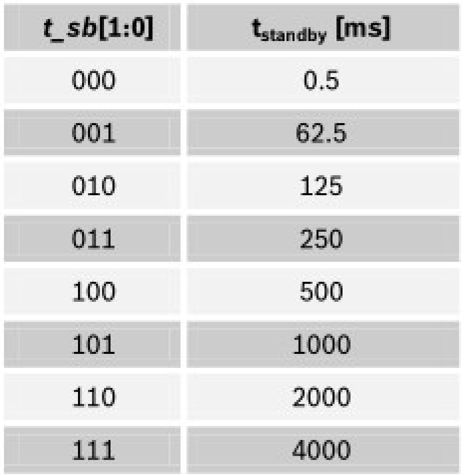
Register 0xF5	Name	Description
Bit 4:2	filter	Controls the time constant of the IIR filter.

→ We don’t use a filter: Value: 0

Settings in config register 0xF5

Register 0xF5	Name	Description
Bit 7:5	t_sb	Controls inactive duration t standby in normal mode.

Table 11: *t_sb* settings



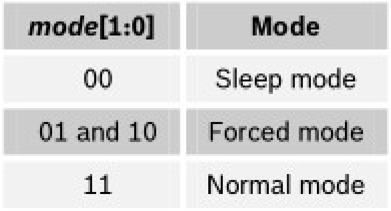
t_sb[1:0]	$t_{standby}$ [ms]
000	0.5
001	62.5
010	125
011	250
100	500
101	1000
110	2000
111	4000

→ One sample every 1s: Value: 5

Settings in meas_ctrl register 0xF4

Register 0xF4	Name	Description
Bit 1:0	mode	Controls the power mode of the device.

Table 10: *mode* settings



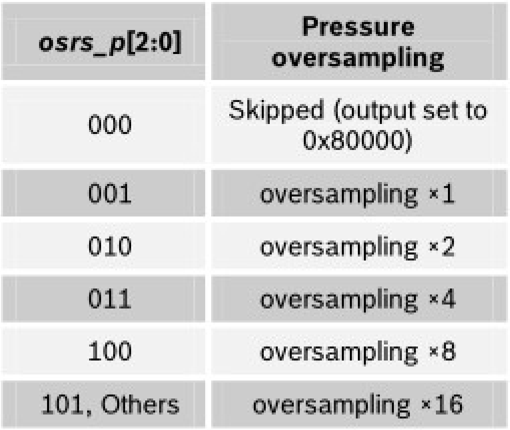
***mode*[1:0]**	**Mode**
00	Sleep mode
01 and 10	Forced mode
11	Normal mode

→ We want normal mode: Value 0x3

Settings in meas_ctrl register 0xF4

Register 0xF4	Name	Description
Bit 4:2	osrs_p	Controls oversampling of pressure data.

Table 21: register settings *osrs_p*



***osrs_p*[2:0]**	**Pressure oversampling**
000	Skipped (output set to 0x80000)
001	oversampling ×1
010	oversampling ×2
011	oversampling ×4
100	oversampling ×8
101, Others	oversampling ×16

→ We will only measure the temperature: Value: 0x0

Settings in meas_ctrl register 0xF4

Register 0xF4	Name	Description
Bit 7:5	osrs_t	Controls oversampling of temperature data.

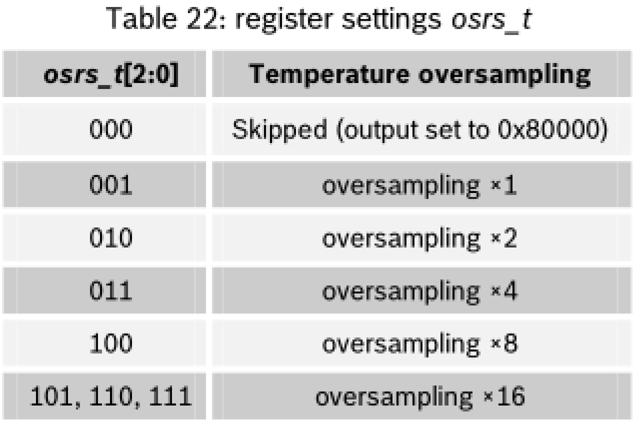


Table 22: register settings *osrs_t*

osrs_t[2:0]	Temperature oversampling
000	Skipped (output set to 0x80000)
001	oversampling ×1
010	oversampling ×2
011	oversampling ×4
100	oversampling ×8
101, 110, 111	oversampling ×16

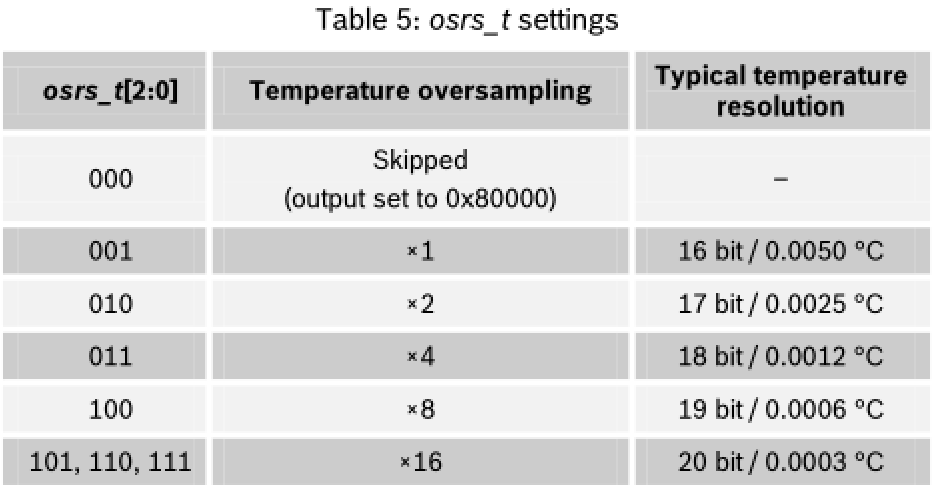
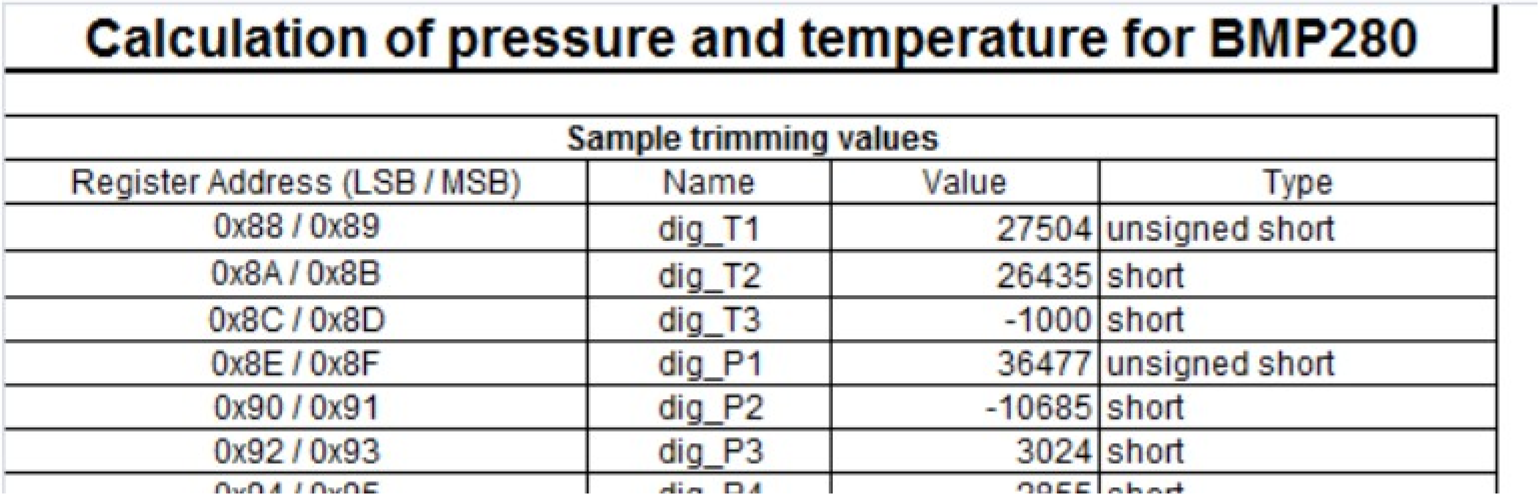


Table 5: *osrs_t* settings

osrs_t[2:0]	Temperature oversampling	Typical temperature resolution
000	Skipped (output set to 0x80000)	–
001	×1	16 bit / 0.0050 °C
010	×2	17 bit / 0.0025 °C
011	×4	18 bit / 0.0012 °C
100	×8	19 bit / 0.0006 °C
101, 110, 111	×16	20 bit / 0.0003 °C

→ Let’s take the highest resolution: Value: 0x5

Read calibration of temperature



Calculation of pressure and temperature for BMP280

Sample trimming values			
Register Address (LSB / MSB)	Name	Value	Type
0x88 / 0x89	dig_T1	27504	unsigned short
0x8A / 0x8B	dig_T2	26435	short
0x8C / 0x8D	dig_T3	-1000	short
0x8E / 0x8F	dig_P1	36477	unsigned short
0x90 / 0x91	dig_P2	-10685	short
0x92 / 0x93	dig_P3	3024	short
0x94 / 0x95	dig_P4	2855	short

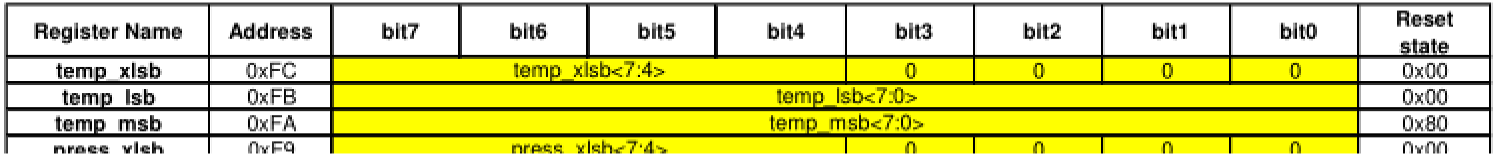
→ Read words at addresses 0x88, 0x8A and 0x8C

Correct calibration values

```
if(dig_T2 > 32767):
    dig_T2 -= 65536

if(dig_T3 > 32767):
    dig_T3 -= 65536
```

Read raw temperature



Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00

→ Read bytes from addresses 0xFA, 0xFB and 0xFC

Calculate Temperature

```
// Returns temperature in DegC, resolution is 0.01 DegC. Output value of
“5123” equals 51.23 DegC.
// t_fine carries fine temperature as global value
BMP280_S32_t t_fine;
BMP280_S32_t bmp280_compensate_T_int32(BMP280_S32_t adc_T)
{
    BMP280_S32_t var1, var2, T;
    var1 = ((((adc_T>>3) - ((BMP280_S32_t)dig_T1<<1))) *
((BMP280_S32_t)dig_T2)) >> 11;
    var2 = (((((adc_T>>4) - ((BMP280_S32_t)dig_T1)) * ((adc_T>>4) -
((BMP280_S32_t)dig_T1))) >> 12) * ((BMP280_S32_t)dig_T3)) >> 14;
    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}
```

What’s about the pressure?

Datasheet:

https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf