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

Data sheet

BMP280
Digital Pressure Sensor

Bosch Sensortec

BMP280: Data sheet

Document revision	1.19
Document release date	January 8th, 2018
Document number	BST-BMP280-DS001-19
Technical reference code(s)	0273 300 436
Notes	Data in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance.

BMP280

DIGITAL PRESSURE SENSOR

Key parameters

• Pressure range	300 ... 1100 hPa (equiv. to +9000...-500 m above/below sea level)
• Package	8-pin LGA metal-lid Footprint : 2.0 × 2.5 mm², height: 0.95 mm
• Relative accuracy (700 ... 900hPa @25°C)	±0.12 hPa, equiv. to ±1 m
• Absolute accuracy (950 ...1050 hPa, 0 ...+40 °C)	typ. ±1 hPa
• Temperature coefficient offset (25 ... 40°C @900hPa)	1.5 Pa/K, equiv. to 12.6 cm/K
• Digital interfaces	I²C (up to 3.4 MHz) SPI (3 and 4 wire, up to 10 MHz)
• Current consumption	2.7µA @ 1 Hz sampling rate
• Temperature range	-40 ... +85 °C
• RoHS compliant, halogen-free	
• MSL 1	

Typical applications

- Enhancement of GPS navigation
  (e.g. time-to-first-fix improvement, dead-reckoning, slope detection)
- Indoor navigation (floor detection, elevator detection)
- Outdoor navigation, leisure and sports applications
- Weather forecast
- Vertical velocity indication (e.g. rise/sink speed)

Target devices

- Handsets such as mobile phones, tablet PCs, GPS devices
- Navigation systems
- Home weather stations
- Flying toys
- Watches

General Description

Robert Bosch is the world market leader for pressure sensors in automotive and consumer applications. Bosch's proprietary APSM (Advanced Porous Silicon Membrane) MEMS manufacturing process is fully CMOS compatible and allows a hermetic sealing of the cavity in an all silicon process. The BMP280 is based on Bosch's proven Piezo-resistive pressure sensor technology featuring high EMC robustness, high accuracy and linearity and long term stability.

The BMP280 is an absolute barometric pressure sensor especially designed for mobile applications. The sensor module is housed in an extremely compact 8-pin metal-lid LGA package with a footprint of only 2.0 × 2.5 mm^2 and 0.95 mm package height. Its small dimensions and its low power consumption of 2.7 µA @1Hz allow the implementation in battery driven devices such as mobile phones, GPS modules or watches.

As the successor to the widely adopted BMP180, the BMP280 delivers high performance in all applications that require precise pressure measurement. The BMP280 operates at lower noise, supports new filter modes and an SPI interface within a footprint 63% smaller than the BMP180.

The emerging applications of indoor navigation, fitness as well as GPS refinement require a high relative accuracy and a low TCO at the same time. BMP180 and BMP280 are perfectly suitable for applications like floor detection since both sensors feature excellent relative accuracy is ±0.12 hPa, which is equivalent to ±1 m difference in altitude. The very low offset temperature coefficient (TCO) of 1.5 Pa/K translates to a temperature drift of only 12.6 cm/K.
Please contact your regional Bosch Sensortec partner for more information about software packages enhancing the calculation of the altitude given by the BMP280 pressure reading.

Table 1: Comparison between BMP180 and BMP280

Parameter	**BMP180**	**BMP280**
Footprint	3.6 × 3.8 mm	2.0 × 2.5 mm
Minimum V_{DD}	1.80 V	1.71 V
Minimum V_{DDIO}	1.62 V	1.20 V
Current consumption @3 Pa RMS noise	12 µA	2.7 µA
RMS Noise	3 Pa	1.3 Pa
Pressure resolution	1 Pa	0.16 Pa
Temperature resolution	0.1°C	0.01°C
Interfaces	I²C	I²C & SPI (3 and 4 wire, mode '00' and '11')
Measurement modes	Only P or T, forced	P&T, forced or periodic
Measurement rate	up to 120 Hz	up to 157 Hz
Filter options	None	Five bandwidths

Index of Contents

1. Specification

If not stated otherwise,

- All values are valid over the full voltage range
- All minimum/maximum values are given for the full accuracy temperature range
- Minimum/maximum values of drifts, offsets and temperature coefficients are ±3σ values over lifetime
- Typical values of currents and state machine timings are determined at 25 °C
- Minimum/maximum values of currents are determined using corner lots over complete temperature range
- Minimum/maximum values of state machine timings are determined using corner lots over 0…+65 °C temperature range

The specification tables are split into pressure and temperature part of BMP280

Table 2: Parameter specification

Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature range	T_A	operational	-40	25	+85	°C
		full accuracy	0		+65	
Operating pressure range	P	full accuracy	300		1100	hPa
Sensor supply voltage	V_{DD}	ripple max. 50mVpp	1.71	1.8	3.6	V
Interface supply voltage	V_{DDIO}		1.2	1.8	3.6	V
Supply current	$I_{DD,LP}$	1 Hz forced mode, pressure and temperature, lowest power		2.8	4.2	µA
Peak current	I_{peak}	during pressure measurement		720	1120	µA
Current at temperature measurement	I_{DDT}			325		µA
Sleep current[1]	I_{DDSL}	25 °C		0.1	0.3	µA
Standby current (inactive period of normal mode) [2]	I_{DDSB}	25 °C		0.2	0.5	µA
Relative accuracy pressure V_{DD} = 3.3V	A_{rel}	700 … 900hPa 25 . . . 40 °C		±0.12		hPa
				±1.0		m

[1] Typical value at VDD = VDDIO = 1.8 V, maximal value at VDD = VDDIO = 3.6 V.
[2] Typical value at VDD = VDDIO = 1.8 V, maximal value at VDD = VDDIO = 3.6 V.

Offset temperature coefficient	TCO	900hPa 25 . . . 40 °C		±1.5		Pa/K
				12.6		cm/K
Absolute accuracy pressure	A^{P}_{ext}	300 . . 1100 hPa -20 . . . 0 °C		±1.7		hPa
	A^{P}_{full}	300 . . . 1100 hPa 0 . . . 65 °C		±1.0		hPa
Resolution of output data in ultra high resolution mode	R^{P}	Pressure		0.0016		hPa
	R^{T}	Temperature		0.01		°C
Noise in pressure	$V_{p,full}$	Full bandwidth, ultra high resolution See chapter 3.5		1.3		Pa
				11		cm
	$V_{p,filtered}$	Lowest bandwidth, ultra high resolution See chapter 3.5		0.2		Pa
				1.7		cm
Absolute accuracy temperature[3]	A^{T}	@ 25 °C		±0.5		°C
		0 . . . +65 °C		±1.0		°C
PSRR (DC)	PSRR	full V_{DD} range			±0.005	Pa/mV
Long term stability[4]	ΔP_{stab}	12 months		±1.0		hPa
Solder drifts		Minimum solder height 50 µm	-0.5		+2	hPa
Start-up time	$t_{startup}$	Time to first communication after both V_{DD} > 1.58V and V_{DDIO} > 0.65V			2	ms
Possible sampling rate	f_{sample}	*osrs_t* = *osrs_p* = 1; See chapter 3.8	157	182	tbd[5]	Hz
Standby time accuracy	$\Delta t_{standby}$			±5	±25	%

[3] Temperature measured by the internal temperature sensor. This temperature value depends on the PCB temperature, sensor element self-heating and ambient temperature and is typically above ambient temperature.
[4] Long term stability is specified in the full accuracy operating pressure range 0 … 65°C
[5] Depends on application case, please contact Application Engineer for further questions

2. Absolute maximum ratings

The absolute maximum ratings are provided in Table 3.

Table 3: Absolute maximum ratings

Parameter	Condition	Min	Max	Unit
Voltage at any supply pin	V_{DD} and V_{DDIO} Pin	-0.3	4.25	V
Voltage at any interface pin		-0.3	V_{DDIO} + 0.3	V
Storage Temperature	≤ 65% rel. H.	-45	+85	°C
Pressure		0	20 000	hPa
ESD	HBM, at any Pin		±2	kV
	CDM		±500	V
	Machine model		±200	V

3. Functional description

The BMP280 consists of a Piezo-resistive pressure sensing element and a mixed-signal ASIC. The ASIC performs A/D conversions and provides the conversion results and sensor specific compensation data through a digital interface.

BMP280 provides highest flexibility to the designer and can be adapted to the requirements regarding accuracy, measurement time and power consumption by selecting from a high number of possible combinations of the sensor settings.

BMP280 can be operated in three power modes (see chapter 3.6):

- sleep mode
- normal mode
- forced mode

In sleep mode, no measurements are performed. Normal mode comprises an automated perpetual cycling between an active measurement period and an inactive standby period. In forced mode, a single measurement is performed. When the measurement is finished, the sensor returns to sleep mode.

A set of oversampling settings is available ranging from ultra low power to ultra high resolution setting in order to adapt the sensor to the target application. The settings are predefined combinations of pressure measurement oversampling and temperature measurement oversampling. Pressure and temperature measurement oversampling can be selected independently from 0 to 16 times oversampling (see chapter 3.3.1 and 3.3.2):

- Temperature measurement
- Ultra low power
- Low power
- Standard resolution
- High resolution
- Ultra high resolution

BMP280 is equipped with a built-in IIR filter in order to minimize short-term disturbances in the output data caused by the slamming of a door or window. The filter coefficient ranges from 0 (off) to 16.

In order to simplify the device usage and reduce the high number of possible combinations of power modes, oversampling rates and filter settings, Bosch Sensortec provides a proven set of recommendations for common use-cases in smart-phones, mobile weather stations or flying toys (see chapter 3.4):

- Handheld device low-power (e.g. smart phones running Android)
- Handheld device dynamic (e.g. smart phones running Android)
- Weather monitoring (setting with lowest power consumption)
- Elevator / floor change detection
- Drop detection
- Indoor navigation

3.1 Block diagram

Figure 1 shows a simplified block diagram of the BMP280:

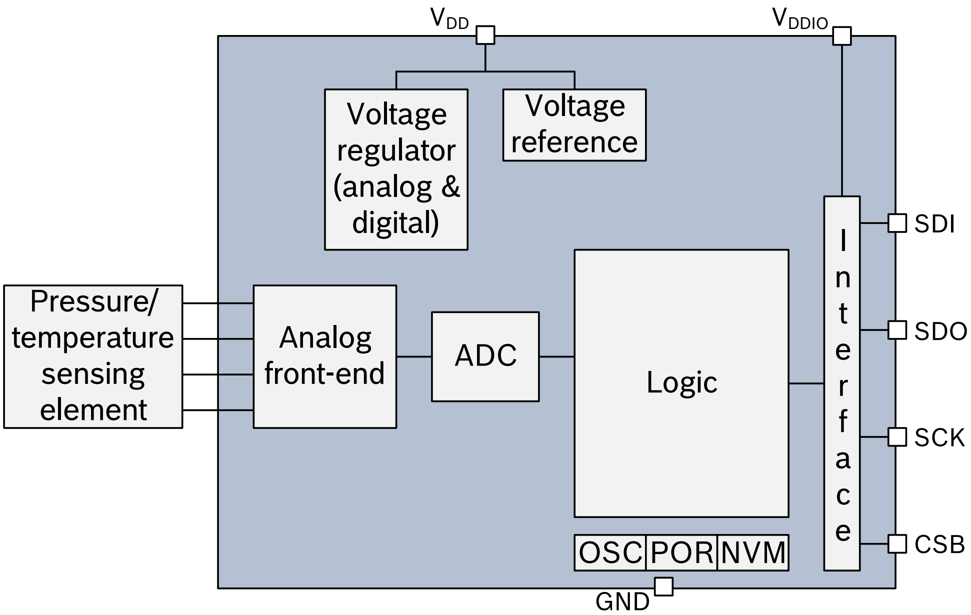

Figure 1: Block diagram of BMP280

3.2 Power management

The BMP280 has two separate power supply pins

- V_{DD} is the main power supply for all internal analog and digital functional blocks
- V_{DDIO} is a separate power supply pin, used for the supply of the digital interface

A power-on reset generator is built in which resets the logic circuitry and the register values after the power-on sequence. There are no limitations on slope and sequence of raising the V_{DD} and V_{DDIO} levels. After powering up, the sensor settles in sleep mode (see 3.6.1).

Warning. Holding any interface pin (SDI, SDO, SCK or CSB) at a logical high level when V_{DDIO} is switched off can permanently damage the device due caused by excessive current flow through the ESD protection diodes.

If V_{DDIO} is supplied, but V_{DD} is not, the interface pins are kept at a high-Z level. The bus can therefore already be used freely before the BMP280 V_{DD} supply is established.

3.3 Measurement flow

The BMP280 measurement period consists of a temperature and pressure measurement with selectable oversampling. After the measurement period, the data are passed through an optional IIR filter, which removes short-term fluctuations in pressure (e.g. caused by slamming a door). The flow is depicted in the diagram below.

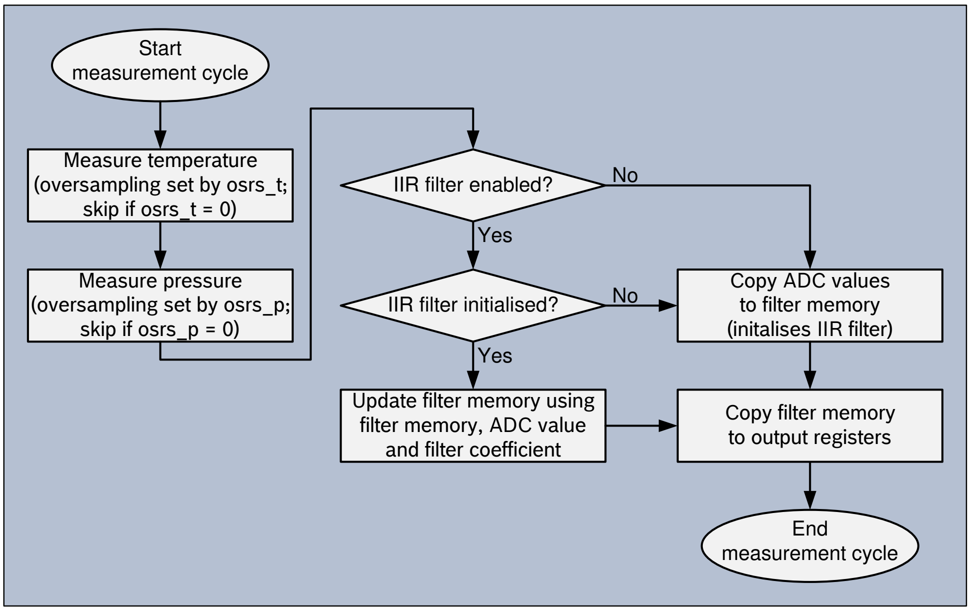

Figure 2: BMP280 measurement cycle

The individual blocks of the diagram above will be detailed in the following subchapters.

3.3.1 Pressure measurement

Pressure measurement can be enabled or skipped. Skipping the measurement could be useful if BMP280 is used as temperature sensor. When enabled, several oversampling options exist. Each oversampling step reduces noise and increases the output resolution by one bit, which is stored in the XLSB data register 0xF9. Enabling/disabling the measurement and oversampling settings are selected through the osrs_p[2:0] bits in control register 0xF4.

Table 4: *osrs_p* settings

Oversampling setting	**Pressure oversampling**	**Typical pressure resolution**	**Recommended temperature oversampling**
Pressure measurement skipped	Skipped (output set to 0x80000)	–	As needed
Ultra low power	×1	16 bit / 2.62 Pa	×1
Low power	×2	17 bit / 1.31 Pa	×1
Standard resolution	×4	18 bit / 0.66 Pa	×1
High resolution	×8	19 bit / 0.33 Pa	×1
Ultra high resolution	×16	20 bit / 0.16 Pa	×2

In order to find a suitable setting for *osrs_p*, please consult chapter 3.4.

3.3.2 Temperature measurement

Temperature measurement can be enabled or skipped. Skipping the measurement could be useful to measure pressure extremely rapidly. When enabled, several oversampling options exist. Each oversampling step reduces noise and increases the output resolution by one bit, which is stored in the XLSB data register 0xFC. Enabling/disabling the temperature measurement and oversampling setting are selected through the osrs_t[2:0] bits in control register 0xF4.

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

It is recommended to base the value of *osrs_t* on the selected value of *osrs_p* as per Table 4. Temperature oversampling above ×2 is possible, but will not significantly improve the accuracy of the pressure output any further. The reason for this is that the noise of the compensated pressure value depends more on the raw pressure than on the raw temperature noise. Following the recommended setting will result in an optimal noise-to-power ratio.

3.3.3 IIR filter

The environmental pressure is subject to many short-term changes, caused e.g. by slamming of a door or window, or wind blowing into the sensor. To suppress these disturbances in the output data without causing additional interface traffic and processor work load, the BMP280 features an internal IIR filter. It effectively reduces the bandwidth of the output signals[6]. The output of a next measurement step is filter using the following formula:

$$data\_filtered = \frac{data\_filtered\_old \cdot (filter\_coefficient - 1) + data\_ADC}{filter\_coefficient},$$

where data_filtered_old is the data coming from the previous acquisition, and data_ADC is the data coming from the ADC before IIR filtering.

The IIR filter can be configured using the filter[2:0] bits in control register 0xF5 with the following options:

[6] Since most pressure sensors do not sample continuously, filtering can suffer from signals with a frequency higher than the sampling rate of the sensor. E.g. environmental fluctuations caused by windows being opened and closed might have a frequency <5 Hz. Consequently, a sampling rate of ODR = 10 Hz is sufficient to obey the Nyquist theorem.

Table 6: *filter* settings

Filter coefficient	Samples to reach ≥75 % of step response
Filter off	1
2	2
4	5
8	11
16	22

In order to find a suitable setting for *filter*, please consult chapter 3.4.

When writing to the register *filter*, the filter is reset. The next value will pass through the filter and be the initial memory value for the filter. If temperature or pressure measurement is skipped, the corresponding filter memory will be kept unchanged even though the output registers are set to 0x80000. When the previously skipped measurement is re-enabled, the output will be filtered using the filter memory from the last time when the measurement was not skipped.

3.4 Filter selection

In order to select optimal settings, the following use cases are suggested:

Table 7: Recommended filter settings based on use cases

Use case	Mode	Over-sampling setting	osrs_p	osrs_t	IIR filter coeff. (see 3.3.3)	I_{DD} [µA] (see 3.7)	ODR [Hz] (see 3.8.2)	RMS Noise [cm] (see 3.5)
handheld device low-power (e.g. Android)	Normal	Ultra high resolution	×16	×2	4	247	10.0	4.0
handheld device dynamic (e.g. Android)	Normal	Standard resolution	×4	×1	16	577	83.3	2.4
Weather monitoring (lowest power)	Forced	Ultra low power	×1	×1	Off	0.14	1/60	26.4
Elevator / floor change detection	Normal	Standard resolution	×4	×1	4	50.9	7.3	6.4
Drop detection	Normal	Low power	×2	×1	Off	509	125	20.8
Indoor navigation	Normal	Ultra high resolution	×16	×2	16	650	26.3	1.6

3.5 Noise

Noise depends on the oversampling and filter settings selected. The stated values were determined in a controlled pressure environment and are based on the average standard deviation of 32 consecutive measurement points taken at highest sampling speed. This is needed in order to exclude long term drifts from the noise measurement.

Table 8: Noise in pressure

Typical RMS noise in pressure [Pa]					
Oversampling setting	**IIR filter coefficient**				
	off	**2**	**4**	**8**	**16**
Ultra low power	3.3	1.9	1.2	0.9	0.4
Low power	2.6	1.5	1.0	0.6	0.4
Standard resolution	2.1	1.2	0.8	0.5	0.3
High resolution	1.6	1.0	0.6	0.4	0.2
Ultra high resolution	1.3	0.8	0.5	0.4	0.2

Table 9: Noise in temperature

Typical RMS noise in temperature [°C]	
Temperature oversampling	**IIR filter off**
oversampling ×1	0.005
oversampling ×2	0.004
oversampling ×4	0.003
oversampling ×8	0.003
oversampling ×16	0.002

3.6 Power modes

The BMP280 offers three power modes: sleep mode, forced mode and normal mode. These can be selected using the mode[1:0] bits in control register 0xF4.

Table 10: *mode* settings

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

3.6.1 Sleep mode

Sleep mode is set by default after power on reset. In sleep mode, no measurements are performed and power consumption (I_{DDSM}) is at a minimum. All registers are accessible; Chip-ID and compensation coefficients can be read.

3.6.2 Forced mode

In forced mode, a single measurement is performed according to selected measurement and filter options. When the measurement is finished, the sensor returns to sleep mode and the measurement results can be obtained from the data registers. For a next measurement, forced mode needs to be selected again. This is similar to BMP180 operation. Forced mode is recommended for applications which require low sampling rate or host-based synchronization.

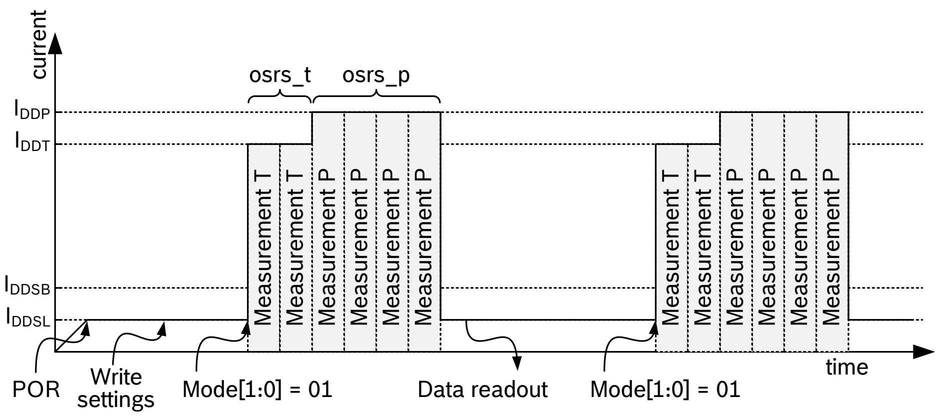

Figure 3: Forced mode timing diagram

3.6.3 Normal mode

Normal mode 16ontinuously cycles between an (active) measurement period and an (inactive) standby period, whose time is defined by $t_{standby}$. The current in the standby period (I_{DDSB}) is slightly higher than in sleep mode. After setting the mode,measurement and filter options, the last measurement results can be obtained from the data registers without the need of further write accesses. Normal mode is recommended when using the IIR filter, and useful for applications in which short-term disturbances (e.g. blowing into the sensor) should be filtered.

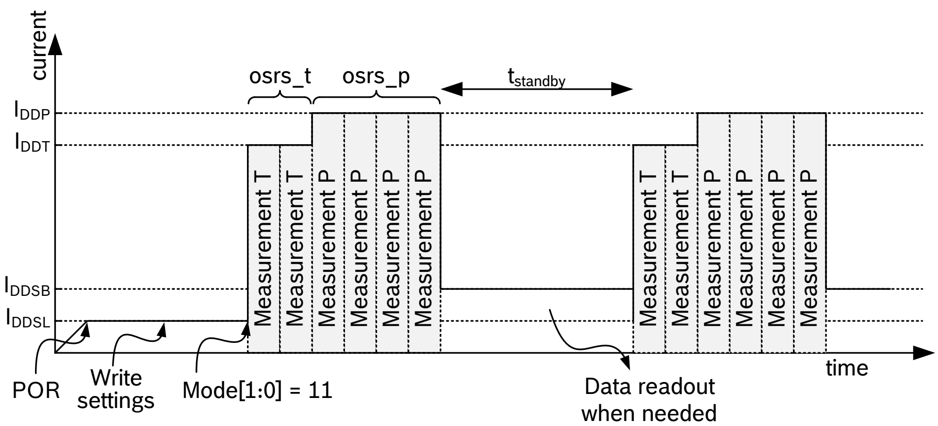

Figure 4: Normal mode timing diagram

The standby time is determined by the contents of the t_sb[2:0] bits in control register 0xF5 according to the table below:

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

3.6.4 Mode transition diagram

The supported mode transitions are displayed below. If the device is currently performing a measurement, execution of mode switching commands is delayed until the end of the currently running measurement period. Further mode change commands are ignored until the last mode change command is executed. Mode transitions other than the ones shown below are tested for stability but do not represent recommended use of the device.

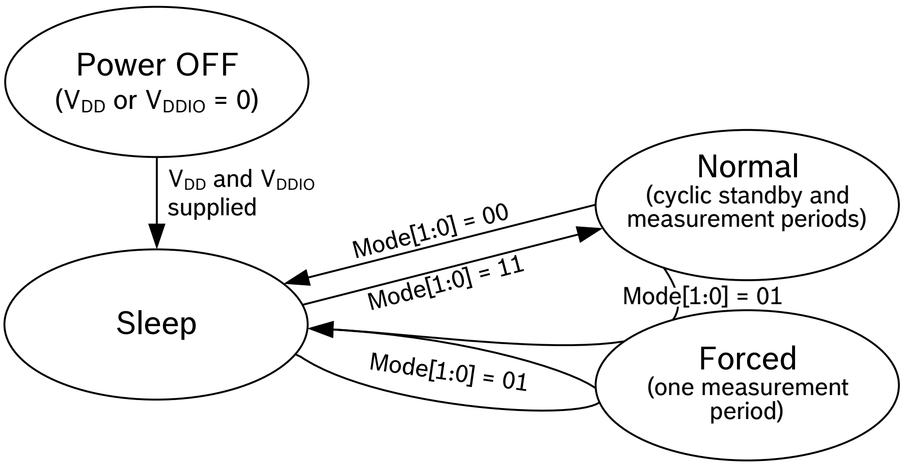

Figure 5: Mode transition diagram

3.7 Current consumption

The current consumption depends on ODR and oversampling setting. The values given below are normalized to an ODR of 1 Hz. The actual consumption at a given ODR can be calculated by multiplying the consumption in Table 12 with the ODR used. The actual ODR is defined either by the frequency at which the user sets forced measurements or by oversampling and $t_{standby}$ settings in normal mode in Table 14.

Table 12: Current consumption

Oversampling setting	Pressure oversampling	Temperature oversampling	I_{DD} [µA] @ 1 Hz forced mode	
			Typ	Max
Ultra low power	×1	×1	2.74	4.16
Low power	×2	×1	4.17	6.27
Standard resolution	×4	×1	7.02	10.50
High resolution	×8	×1	12.7	18.95
Ultra high resolution	×16	×2	24.8	36.85

3.8 Measurement timings

The rate at which measurements can be performed in forced mode depends on the oversampling settings *osrs_t* and *osrs_p*. The rate at which they are performed in normal mode depends on the oversampling setting settings *osrs_t* and *osrs_p* and the standby time $t_{standby}$. In the following table the resulting ODRs are given only for the suggested osrs combinations.

3.8.1 Measurement time

The following table explains the typical and maximum measurement time based on selected oversampling setting. The minimum achievable frequency is determined by the maximum measurement time.

Table 13: measurement time

Oversampling setting	Pressure oversampling	Temperature oversampling	Measurement time [ms]		Measurement rate [Hz]	
			Typ	Max	Typ	Min
Ultra low power	×1	×1	5.5	6.4	181.8	155.6
Low power	×2	×1	7.5	8.7	133.3	114.6
Standard resolution	×4	×1	11.5	13.3	87.0	75.0
High resolution	×8	×1	19.5	22.5	51.3	44.4
Ultra high resolution	×16	×2	37.5	43.2	26.7	23.1

3.8.2 Measurement rate in normal mode

The following table explains which measurement rates can be expected in normal mode based on oversampling setting and $t_{standby}$.

Table 14: typical output data Rate (ODR) in normal mode [Hz]

Oversampling setting	$t_{standby}$ [ms]							
	0.5	**62.5**	**125**	**250**	**500**	**1000**	**2000**	**4000**
Ultra low power	166.67	14.71	7.66	3.91	1.98	0.99	0.50	0.25
Low power	125.00	14.29	7.55	3.88	1.97	0.99	0.50	0.25
Standard resolution	83.33	13.51	7.33	3.82	1.96	0.99	0.50	0.25
High resolution	50.00	12.20	6.92	3.71	1.92	0.98	0.50	0.25
Ultra high resolution	26.32	10.00	6.15	3.48	1.86	0.96	0.49	0.25

Table 15: Sensor timing according to recommended settings (based on use cases)

Use case	Mode	Over-sampling setting	osrs_p	osrs_t	IIR filter coeff. (see 3.3.3)	Timing	ODR [Hz] (see 3.8.2)	BW [Hz] (see 3.3.3)
handheld device low-power (e.g. Android)	Normal	Ultra high resolution	×16	×2	4	$t_{standby}$ = 62.5 ms	10.0	0.92
handheld device dynamic (e.g. Android)	Normal	Standard resolution	×4	×1	16	$t_{standby}$ = 0.5 ms	83.3	1.75
Weather monitoring (lowest power)	Forced	Ultra low power	×1	×1	Off	1/min	1/60	full
Elevator / floor change detection	Normal	Standard resolution	×4	×1	4	$t_{standby}$ = 125 ms	7.3	0.67
Drop detection	Normal	Low power	×2	×1	Off	$t_{standby}$ = 0.5 ms	125	full
Indoor navigation	Normal	Ultra high resolution	×16	×2	16	$t_{standby}$ = 0.5 ms	26.3	0.55

3.9 Data readout

To read out data after a conversion, it is strongly recommended to use a burst read and not address every register individually. This will prevent a possible mix-up of bytes belonging to different measurements and reduce interface traffic. Data readout is done by starting a burst read from 0xF7 to 0xFC. The data are read out in an unsigned 20-bit format both for pressure and for temperature. It is strongly recommended to use the BMP280 API, available from Bosch

Sensortec, for readout and compensation. For details on memory map and interfaces, please consult chapters 3.12 and 5 respectively.

The timing for data readout in forced mode should be done so that the maximum measurement times (see chapter 3.8.1) are respected. In normal mode, readout can be done at a speed similar to the expected data output rate (see chapter 3.8.2). After the values of 'ut' and 'up' have been read, the actual pressure and temperature need to be calculated using the compensation parameters stored in the device. The procedure is elaborated in chapter 3.11.

3.10 Data register shadowing

In normal mode, measurement timing is not necessarily synchronized to readout. This means that new measurement results may become available while the user is reading the results from the previous measurement. In this case, shadowing is performed in order to guarantee data consistency. Shadowing will only work if all data registers are read in a single burst read. Therefore, the user must use burst reads if he does not synchronize data readout with the measurement cycle. Using several independent read commands may result in inconsistent data.

If a new measurement is finished and the data registers are still being read, the new measurement results are transferred into shadow data registers. The content of shadow registers is transferred into data registers as soon as the user ends the burst read, even if not all data registers were read. Reading across several data registers can therefore only be guaranteed to be consistent within one measurement cycle if a single burst read command is used. The end of the burst read is marked by the rising edge of CSB pin in SPI case or by the recognition of a stop condition in I2C case. After the end of the burst read, all user data registers are updated at once.

3.11 Output compensation

The BMP280 output consists of the ADC output values. However, each sensing element behaves differently, and actual pressure and temperature must be calculated using a set of calibration parameters. The recommended calculation in chapter 3.11.3 uses fixed point arithmetic. In high-level languages like Matlab™ or LabVIEW™, fixed-point code may not be well supported. In this case the floating-point code in appendix 8.1 can be used as an alternative. For 8-bit micro controllers, the variable size may be limited. In this case a simplified 32 bit integer code with reduced accuracy is given in appendix 8.2.

3.11.1 Computational requirements

The table below shows the number of clock cycles needed for compensation calculations on a 32 bit Cortex-M3 micro controller with GCC optimization level –O2. This controller does not contain a floating point unit, so all floating-point calculations are emulated. Floating point is only recommended for PC applications where an FPU is present.

Table 16: Computational requirements for compensation formulas

Compensation of	Number of clock cycles (ARM Cortex-M3)		
	32 bit integer	64 bit integer	Double precision
Temperature	~46	–	~2400 [7]
Pressure	~112 [8]	~1400	~5400 7

3.11.2 Trimming parameter readout

The trimming parameters are programmed into the devices' non-volatile memory (NVM) during production and cannot be altered by the customer. Each compensation word is a 16-bit signed or unsigned integer value stored in two's complement. As the memory is organized into 8-bit words, two words must always be combined in order to represent the compensation word. The 8-bit registers are named calib00…calib25 and are stored at memory addresses 0x88…0xA1. The corresponding compensation words are named dig_T# for temperature compensation related values and dig_P# for pressure compensation related values. The mapping is shown in Table 17.

Table 17: Compensation parameter storage, naming and data type

Register Address LSB / MSB	Register content	Data type
0x88 / 0x89	dig_T1	unsigned short
0x8A / 0x8B	dig_T2	signed short
0x8C / 0x8D	dig_T3	signed short
0x8E / 0x8F	dig_P1	unsigned short
0x90 / 0x91	dig_P2	signed short
0x92 / 0x93	dig_P3	signed short
0x94 / 0x95	dig_P4	signed short
0x96 / 0x97	dig_P5	signed short
0x98 / 0x99	dig_P6	signed short
0x9A / 0x9B	dig_P7	signed short
0x9C / 0x9D	dig_P8	signed short
0x9E / 0x9F	dig_P9	signed short
0xA0 / 0xA1	reserved	reserved

3.11.3 Compensation formula

Please note that it is strongly advised to use the API available from Bosch Sensortec to perform readout and compensation. If this is not wanted, the code below can be applied at the user's risk. Both pressure and temperature values are expected to be received in 20 bit format, positive, stored in a 32 bit signed integer.

[7] Use only recommended for high-level programming languages like Matlab™ or LabVIEW™
[8] Use only recommended for 8-bit micro controllers

The variable t_fine (signed 32 bit) carries a fine resolution temperature value over to the pressure compensation formula and could be implemented as a global variable.
The data type “BMP280_S32_t” should define a 32 bit signed integer variable type and can usually be defined as “long signed int”.

The data type “BMP280_U32_t” should define a 32 bit unsigned integer variable type and can usually be defined as “long unsigned int”.

For best possible calculation accuracy, 64 bit integer support is needed. If this is not possible on your platform, please see appendix 8.2 for a 32 bit alternative.

The data type “BMP280_S64_t” should define a 64 bit signed integer variable type, which on most supporting platforms can be defined as “long long signed int”. The revision of the code is rev.1.1.

```
// 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;
}
“”_
// Returns pressure in Pa as unsigned 32 bit integer in Q24.8 format (24 integer bits and 8 fractional bits).
// Output value of “24674867” represents 24674867/256 = 96386.2 Pa = 963.862 hPa
BMP280_U32_t bmp280_compensate_P_int64(BMP280_S32_t adc_P)
{
   BMP280_S64_t var1, var2, p;
   var1 = ((BMP280_S64_t)t_fine) – 128000;
   var2 = var1 * var1 * (BMP280_S64_t)dig_P6;
   var2 = var2 + ((var1*(BMP280_S64_t)dig_P5)<<17);
   var2 = var2 + (((BMP280_S64_t)dig_P4)<<35);
   var1 = ((var1 * var1 * (BMP280_S64_t)dig_P3)>>8) + ((var1 * (BMP280_S64_t)dig_P2)<<12);
   var1 = (((((BMP280_S64_t)1)<<47)+var1))*((BMP280_S64_t)dig_P1)>>33;
   if (var1 == 0)
   {
      return 0; // avoid exception caused by division by zero
   }
   p = 1048576-adc_P;
   p = (((p<<31)-var2)*3125)/var1;
   var1 = (((BMP280_S64_t)dig_P9) * (p>>13) * (p>>13)) >> 25;
   var2 = (((BMP280_S64_t)dig_P8) * p) >> 19;
   p = ((p + var1 + var2) >> 8) + (((BMP280_S64_t)dig_P7)<<4);
   return (BMP280_U32_t)p;
```

3.12 Calculating pressure and temperature

The following figure shows the detailed algorithm for pressure and temperature measurement.

This algorithm is available to customers as reference C source code (“BMP28x_ API”) from Bosch Sensortec and via its sales and distribution partners.

Please contact your Bosch Sensortec representative for details.

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
0x96 / 0x97	dig_P5	140	short
0x98 / 0x99	dig_P6	-7	short
0x9A / 0x9B	dig_P7	15500	short
0x9C / 0x9D	dig_P8	-14600	short
0x9E / 0x9F	dig_P9	6000	short
0xA0 / 0xA1			

Sample measurement values				
Register Address (MSB / LSB / XLSB)	Name	Value	Type	
0xF7 / 0xF8 / 0xF9[7:4]	UT [20 bit]	519888	signed long (*)	(*) Value is always positive, even though the compensation functions expect a signed integer as input
0xFA / 0xFB / 0xFC[7:4]	UP [20 bit]	415148	signed long (*)	(*) Value is always positive, even though the compensation functions expect a signed integer as input

var1 =	128793,1787		var1 = (((double)adc_T)/16384.0 - ((double)dig_T1)/1024.0) * ((double)dig_T2);
var2 =	-370,8917052		var2 = ((((double)adc_T)/131072.0 - ((double)dig_T1)/8192.0) * (((double)adc_T)/131072.0 - ((double) dig_T1)/8192.0)) * ((double)dig_T3);
tfine =	128422		t_fine = (BMP280_S32_t)(var1 + var2);
T =	**25,08**	***Temperature [°C]***	T = (var1 + var2) / 5120.0;
integer result (**):	**2508**	***Temperature [1/100 °C]***	
var1 =	211,1435029		var1 = ((double)t_fine/2.0) - 64000.0;
var2 =	-9,523652701		var2 = var1 * var1 * ((double)dig_P6) / 32768.0;
var2 =	59110,65716		var2 = var2 + var1 * ((double)dig_P5) * 2.0;
var2 =	187120057,7		var2 = (var2/4.0)+(((double)dig_P4) * 65536.0);
var1 =	-4,302618389		var1 = (((double)dig_P3) * var1 * var1 / 524288.0 + ((double)dig_P2) * var1) / 524288.0;
var1 =	36472,21037		var1 = (1.0 + var1 / 32768.0)*((double)dig_P1);
p =	633428		p = 1048576.0 - (double)adc_P;
p =	100717,8456		p = (p - (var2 / 4096.0)) * 6250.0 / var1;
var1 =	28342,24444		var1 = ((double)dig_P9) * p * p / 2147483648.0;
var2 =	-44875,50492		var2 = p * ((double)dig_P8) / 32768.0;
p =	**100653,27**	***Pressure [Pa]***	p = p + (var1 + var2 + ((double)dig_P7)) / 16.0;
int32 result (**):	**100653**	***Pressure [Pa]***	
int64 result (**):	**25767236**	***Pressure [1/256 Pa]***	

(**) The actual result of the integer calculation may deviate slightly from the values shown here due to integer calculation rounding errors

4. Global memory map and register description

4.1 General remarks

All communication with the device is performed by reading from and writing to registers. Registers have a width of 8 bits. There are several registers which are reserved; they should not be written to and no specific value is guaranteed when they are read. For details on the interface, consult chapter 5.

4.2 Memory map

The memory map is given in Table 18 below. Reserved registers are not shown.

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**

4.3 Register description

4.3.1 Register 0xD0 *"id"*

The *"id"* register contains the chip identification number chip_id[7:0], which is 0x58. This number can be read as soon as the device finished the power-on-reset.

4.3.2 Register 0xE0 *"reset"*

The *"reset"* register contains the soft reset word reset[7:0]. If the value 0xB6 is written to the register, the device is reset using the complete power-on-reset procedure. Writing other values than 0xB6 has no effect. The readout value is always 0x00.

4.3.3 Register 0xF3 *"status"*

The *"status"* register contains two bits which indicate the status of the device.

Table 19: Register 0xF3 *"status"*

Register 0xF3 *"status"*	Name	Description
Bit 3	measuring[0]	Automatically set to '1' whenever a conversion is running and back to '0' when the results have been transferred to the data registers.
Bit 0	im_update[0]	Automatically set to '1' when the NVM data are being copied to image registers and back to '0' when the copying is done. The data are copied at power-on-reset and before every conversion.

4.3.4 Register 0xF4 *"ctrl_meas"*

The *"ctrl_meas"* register sets the data acquisition options of the device.

Table 20: Register 0xF4 *"ctrl_meas"*

Register 0xF4 *"ctrl_meas"*	Name	Description
Bit 7, 6, 5	osrs_t[2:0]	Controls oversampling of temperature data. See chapter 3.3.2 for details.
Bit 4, 3, 2	osrs_p[2:0]	Controls oversampling of pressure data. See chapter 3.3.1 for details.
Bit 1, 0	mode[1:0]	Controls the power mode of the device. See chapter 3.6 for details.

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