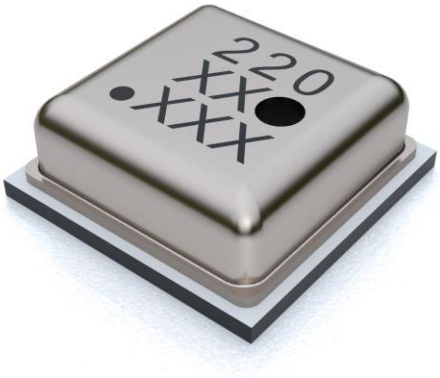


ENS220

Premium accuracy barometric pressure and temperature sensor for activity tracking and indoor navigation / localization

ENS220 datasheet

Version: 1.0

Release Date: 2023-09-19

Document Status: Production

The ENS220 is an ultra-low-power, high-accuracy barometric pressure and temperature sensor. It comes in the smallest size LGA package with digital I²C and SPI interfaces. This enables new use cases in activity tracking, indoor navigation/localization, fall- and liquid-level detection.

The capacitive pressure sensor of the ENS220 is integrated on a CMOS ASIC. This single die solution not only allows small form factor packages with excellent immunity to changes in environmental conditions, but also achieves ultra-low current consumption due to the capacitive read-out. High intrinsic pressure sensitivity combined with an ultra-low noise 24-bit ADC converter results in unprecedented low pressure noise. The integrated temperature sensor matches the performance of dedicated temperature measurement devices. It ensures a stable, temperature compensated pressure output with a fast response time. The highly accurate pressure reading, and the short conversion time make the ENS220 perfect for height measurements with a high output data rate and high bandwidth.

Key Features & Benefits

Premium **absolute accuracy:** ±0.5 hPa

Relative accuracy: ±0.025 hPa (equiv. to ±20 cm in air) allows ultra-fine differential pressure measurements for reliable **centimeter resolution positioning**

Lowest power: average supply current of 0.1 µA at idle, 0.8 µA when sampling at 1/60 Hz, ensures **long battery life** even at high sampling rates

Ultra-low noise of 0.1 Pa rms (≈ 1 cm) at 2 Hz sample rate

User-configurable sample rate up to 1 kHz (0.9 Pa rms)

Temperature accuracy of ±0.1 K with 8 mK resolution

Fully digital interface with **best-of-breed form factor**, which perfectly integrates into **space-constrained designs**, e.g., mobiles, wearables, hearables

Interrupt output on multiple conditions, **including pressure thresholds** for ultra-low power applications

Applications

- Mobile/Wearables: activity tracking, indoor localization/navigation, fall detection
- Gaming, AR/VR, Drones: height tracking
- Appliances/HVAC: filter clogging detection, building balancing
- White goods: liquid level detection
- Personal health care: blood pressure measurement[1]
- Accurate temperature meter for gasses and surfaces

Properties

- Small 2.0 x 2.0 x 0.75 mm³ LGA package
- P operating range from 300 to 1200 hPa
- T operating range from −40 to +85 °C
- Power supply range from 1.62 V to 1.98 V, interface voltage up to 3.6 V
- Standard, fast, and high speed I²C and SPI interface
- T&R packaged, reflow solderable.
- MSL1 compliant

[1] Consumer grade product, not medically qualified. Customer must ensure compliance.

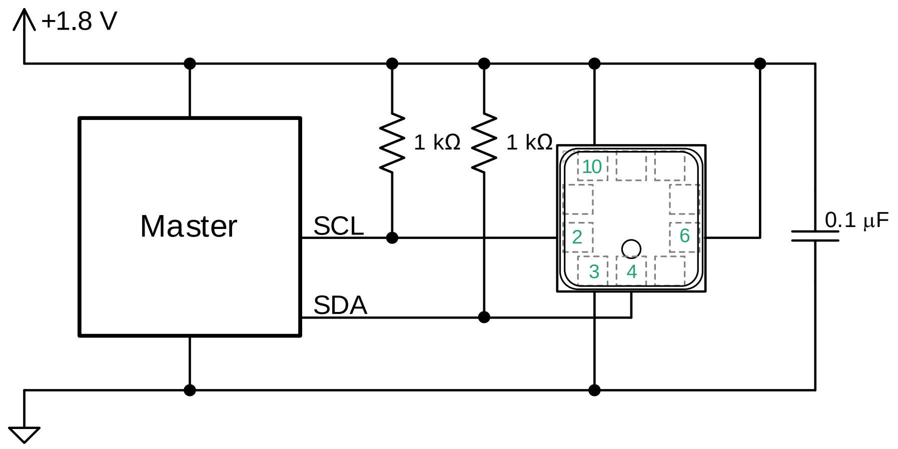
Content Guide

1 Quick Start

A typical application circuit for ENS220 is shown on the left-hand side of Figure 1. After reaching the minimal supply voltage and allowing for the maximal power-up time of 1 ms the sensor is ready for I²C communication. The quickest way to measure pressure and temperature is pseudo-coded in Figure 1. The conversion formula given in the equations in Table 5 and Table 6 are used to translate the digital signals into absolute pressure and temperature readings.

More application examples can be found in chapter 9.

Basic application circuit



Pseudo code

```
Write(0x06,0x93)    //Set HP, direct path, measure, P enabled, T enabled
Repeat forever
       I=Read(0x14)          //read DATA_STAT
       //make sure to use 32 bit integers
       if (I & 0x1)!=0     //check TR flag
              T=Read(0x1A) + 256*Read(0x1B)
              Print(T/128.0 - 273.15, "C")
       if (I & 0x2)!=0     //check PR flag
              P=Read(0x17) + 256*Read(0x18) + 256*256*Read(0x19)
              Print(P/64.0, "Pa")
```

Figure 1: Typical application circuit (top) and pseudo code (bottom) for easy starting. For details on the sensor configuration see section 6.

Find code resources and drivers on: https://github.com/sciosense/ens220-arduino

2 Pin assignment

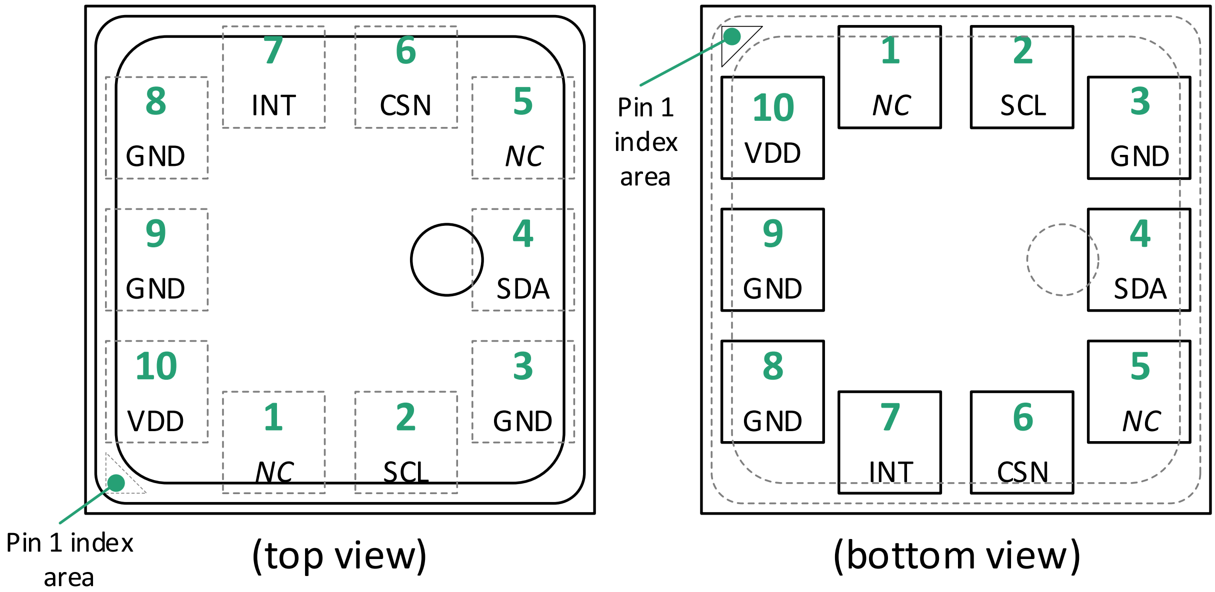


Figure 2: Pin diagram (NC = not connected)

Consult section 9 for wiring.

Table 1: Pin description

Pin	Pin Name	Pin Type	Description
1, 5	NC	unconnected	Unconnected pins. These are isolated from other pins.
2	SCL / SCLK	Input	I²C / SPI bus serial clock input
3, 8, 9	GND	Supply	Ground supply voltage
4	SDA / SDI	Input / Output	I²C bus bi-directional data / SPI data input of the sensor (MOSI)
6	CSN	Input	SPI interface select (CSN low -> SPI / CSN high -> I²C)
7	INT / SDO	Output	Interrupt output to controller / SPI data output of the sensor (MISO)
10	VDD	Supply	Supply voltage

3 Pressure and temperature specifications

Default conditions apply to values in Table 2, unless otherwise stated: 25 °C, 50 % RH, no MSL1 pre-conditioning, default periodic measurement.

Table 2: Pressure and temperature specifications

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
	Pressure					
P_{RANGE}	Pressure operating range		300		1200	hPa
P_{ABS}	Absolute pressure accuracy[2]	500…1100 hPa, −5…+55 °C		±0.5		hPa
		500…1100 hPa, −20…+70 °C		±1		hPa
		300…1200 hPa, −40…+85 °C		±3		hPa
P_{REL}	Relative accuracy	P step of 10 hPa, 600...1100 hPa, −5…+55 °C		2.5		Pa
P_{RES}	Pressure resolution	1 LSB		1/64		Pa
P_{SOLDER}	Absolute pressure accuracy after soldering[3]	1000 hPa, 25 °C		±1		hPa
P_{DRIFT}	Long term stability[4]			±10		Pa/year
T_{SENS}	Temperature sensitivity	500…1100 hPa, 25…45 °C		0.5		Pa/K
P_{NOISE}	RMS pressure noise	4 ms conversion, no over-sampling, 1000 hPa, 25 °C [5]		0.85		Pa
	Temperature					
T_{RANGE}	Temperature operating range		−40		+85	°C
T_{ACC}	Absolute temperature accuracy	-5…+55 °C	−0.2	±0.1	+0.2	°C
		−40…+85 °C	−0.5		+0.5	°C
T_{RES}	Temperature resolution			1/128		K

[2] After calibration at component-level, not including the influence of soldering and lifetime use.
[3] 168 h after soldering. The sensor is calibrated at factory for the temperature profile given in Figure 31.
[4] Average over lifetime, based on temperature-accelerated measurements.
[5] Results in ODR ≈ 40…250 Hz. See chapter 9.4 for recommended application settings.

4 Electrical characteristics

Table 3 details the electrical characteristics of the ENS220. The min and max parameter values are guaranteed by production tests or SQC (Statistical Quality Control) methods.

Table 3: Electrical characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Positive supply		1.62	1.8	1.98	V
V_{IO_OD}[6]	Interface I/O voltage SDA, SCL			1.8	3.6	V
T_{OP}	Operating temperature		−40		+85	°C
I_{DD_ULP}	Current consumption in idle state	HP = 0, V_{DD} = 1.8 V, 35 °C		105		nA
		HP = 0, V_{DD} = 1.98 V, 85 °C		1600		nA
I_{DD_LP}		HP = 1		40		µA
I_{CONV}	Current consumption during T or P conversion	128× oversampling, continuous mode		75		µA
ODR_{max}	Sampling rate			970		1/s
t_{PUP}	Power up time				1	ms
t_{VddTr}	V_{DD} rise time at power on				10	ms
a_{t_stby}	Stand-by time accuracy				20	%
t_{CONV_T}	Temperature conversion time	T = 25 °C, no oversampling		4	4.8	ms
t_{CONV_P}	Pressure conversion time	T = 25 °C, no oversampling, P_CONV = 0, MEAS_T = 0, STBY_T = 0		1	1.2	ms
V_{IH}	High-level input voltage		0.7 × V_{DD}			V
V_{IL}	Low-level input voltage				0.25 × V_{DD}	V
V_{OH}	High-level output voltage	INT / SDO sourcing 6 mA	V_{DD} − 0.5			V
V_{OL}	Low-level output voltage	INT / SDO sinking 6 mA			0.4	V

At power-up, the supply voltage (V_{DD}) must rise within t_{VddTr} = 10 ms to the minimum supply voltage. Otherwise, the internal reset may malfunction.

[6] See Figure 28. SCL and SDA should be set to 0 V when the device is powered off (V_{DD} = 0 V). Otherwise, current may flow from the (3.6 V) bus into SDA or SCL.

5 Absolute maximum ratings

Table 4: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units	Comments
Electrical Parameters					
V_{DD}	Supply Voltage	−0.3	1.98	V	
V_{IO_PP}[7]	Interface I/O voltage CSN, INT	−0.3	V_{DD} + 0.3	V	
V_{IO_OD}	Interface I/O voltage SDA, SCL	−0.3	3.6	V	
I_{SCR}	Input Current (latch-up immunity)	−50	+50	mA	JEDEC JESD78F
Electrostatic Discharge					
ESD_{HBM}	Electrostatic Discharge HBM	± 2000		V	AEC-Q100-012 Rev. E
ESD_{CDM}	Electrostatic Discharge CDM	± 500		V	AEC-Q100-011 Rev. D
Operating and Storage Conditions					
T_{STRG}	Storage Temperature	−40	+85	°C	
RH_{STRG}	Storage Relative Humidity	10	90	%	Non-condensing
T_{AMB}[8]	Operating Ambient Temperature	−40	+85	°C	
RH_{AMB}	Operating Ambient Rel. Humidity	5	95	%	Non-condensing
P_{MAX}	Survival pressure		5000	hPa	
$T_{lifetime}$	Sensor lifetime	10		years	Under typical operating conditions

Stresses beyond those listed in Table 4: Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated under Table 3: Electrical characteristics are not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability and lifetime.

Note: The ENS220 is not designed for use in safety-critical or life-protecting applications.

[7] V_{IO_PP} should not exceed the absolute maximum rating of V_{DD}.
[8] The ENS220 is electrically operable in this range, however its pressure sensing performance might vary.

6 Functional description

The ENS220 integrates an absolute pressure sensor, a temperature sensor and an ASIC which performs A/D conversions and provides the pressure and temperature data through a digital interface.

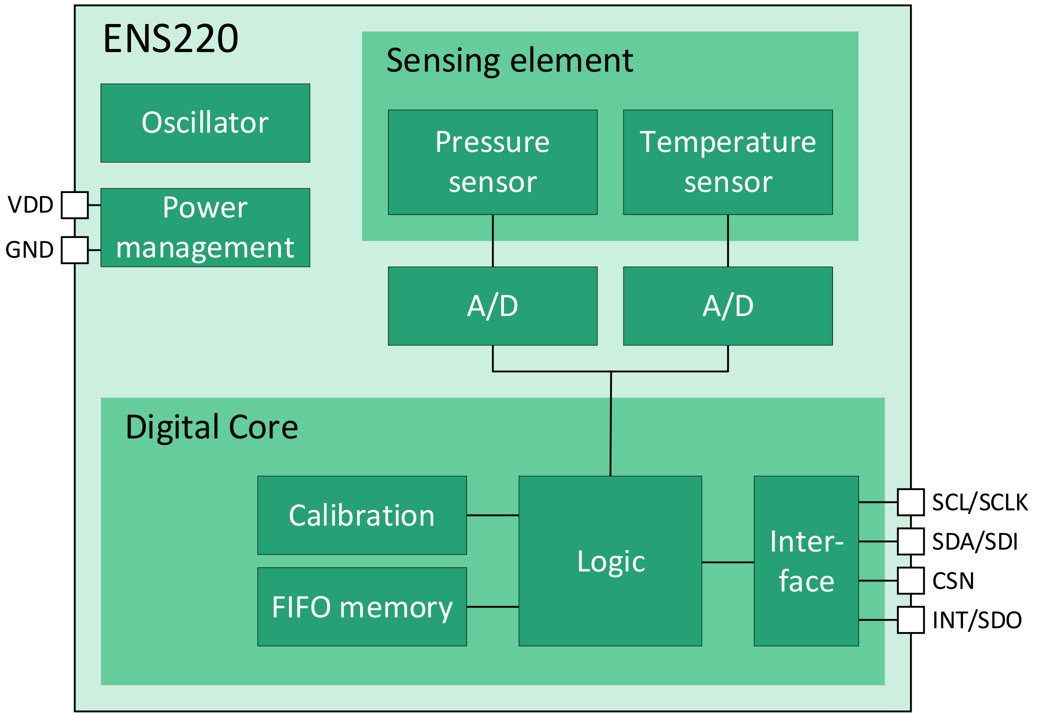


Figure 3: Block diagram of the ENS220

The ENS220 is a sensor capable of providing fast and accurate temperature measurements at low current consumption. To meet the requirements of different use cases the ENS220 offers very high configuration flexibility.

The configurable features are:

- Operating mode
  - Continuous
  - Pulsed
  - Single Shot
- Pressure measurement precision (conversion time and oversampling) and rate
- Temperature measurement precision (oversampling) and rate
- Ultra-low power mode
- FIFO
- Interrupt behavior
  - New measurement available
  - FIFO at watermark level
  - FIFO full
  - High resolution pressure threshold
  - No interrupt
- Moving average

6.1 Measurement modes

The sensor starts powering-up after reaching the power-up threshold voltage V_{DD}, Min = 1.62 V. The sensor enters the default idle mode, in which no measurements are performed, and the sensor configuration can be chosen.

By switching the START bit to 1, the sensor starts measuring pressure and temperature. The ENS220 can be operated in three measurement modes, which can be selected with the value of STBY_T (see also Figure 4):

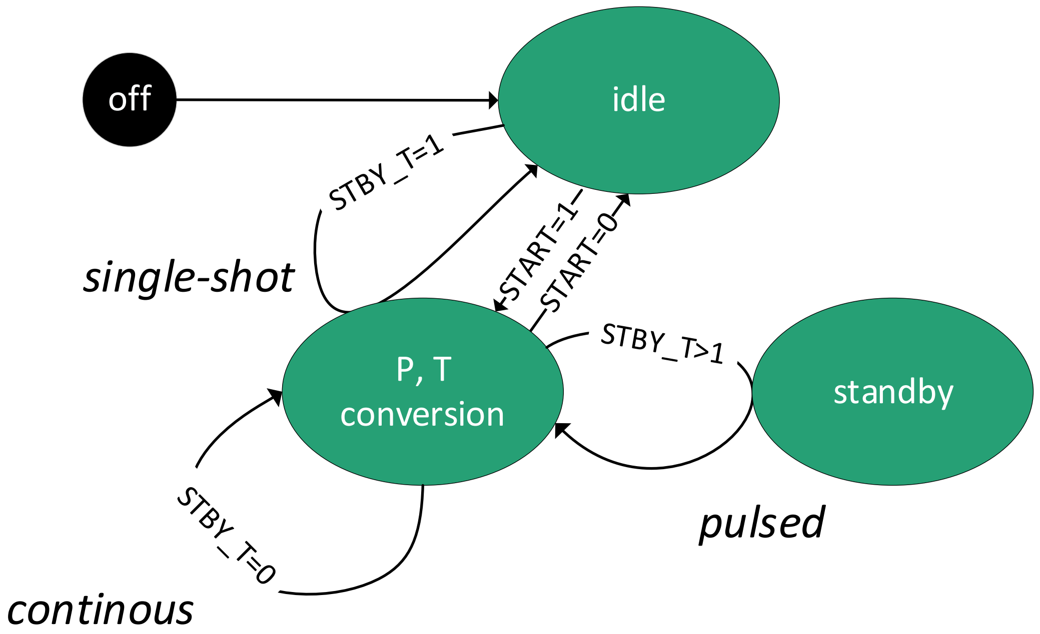


Figure 4: Simplified schematic of measurement modes

- If STBY_T = 0 the device performs **continuous** measurements (temperature and pressure).
- If STBY_T > 1 the device alternates a measurement phase and a standby phase (**pulsed** operation)
- If STBY_T = 1 the device performs a **single** measurement (temperature and pressure) and goes back into idle mode.

The length of the standby phase can be selected from 10 ms to 600 s by writing on STBY_T. A timing diagram of pulsed operation is shown below in Figure 5.

6.2 PT-Rate

In default configuration (see default values in register table), the device performs one pressure measurement followed by a temperature measurement. The ratio between pressure and temperature measurements can be altered by selecting a pressure/temperature rate by writing on PT_RATE. The value varies from 1 to 256. The device will perform a new temperature measurement once every PT_RATE pressure measurements (see Figure 5).

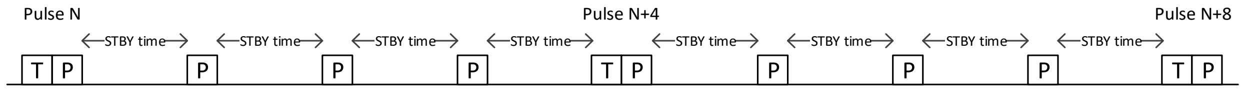


Figure 5: Effect of PT_RATE = 1 (meaning P/T rate is 4)

MEAS_P and MEAS_T determine whether pressure or temperature measurements take place. If only temperature or only pressure is enabled, PT_RATE has no effect.

The ENS220 uses the temperature value to compensate the pressure output. It is therefore advisable to take regular temperature measurements (at least once per second) to correct temperature fluctuations. For fast temperature transients the temperature measurement should be performed more often to avoid errors in the temperature compensation. The best setting for the temperature update rate depends on the use case and the physical properties of the soldering substrate. It should be evaluated by the user. See also use case examples in chapter 9.

If only pressure measurements are enabled (MEAS_P=0), the device will begin with a temperature measurement, then it will continue only measuring the pressure. See also Figure 8.

6.3 Readout

6.3.1 Absolute Pressure

The pressure value is a 24 bit unsigned integer and is available in the sensor readout registers PRESS_OUT and PRESS_OUT_F.

The PRESS_OUT value is already calibrated, linearized, and temperature compensated. It requires a division by 64 to achieve a pressure reading in Pa:

Table 5: Pressure conversion formula

Pascal
$P/[\mathrm{Pa}] = \frac{PRESS\_OUT}{64}$

6.3.2 Temperature

The temperature value is a 16-bit unsigned integer and is available in the sensor readout registers TEMP_OUT. Temperature can be calculated as:

Table 6: Temperature conversion formulas

Celsius	Kelvin
$T_C/[°\mathrm{C}] = \frac{TEMP\_OUT}{128} - 273.15$	$T_K/[\mathrm{K}] = \frac{TEMP\_OUT}{128}$

6.4 Ultra low power mode

Power consumption can be reduced at the expense of some functionality.

If the HP bit is 0 (see MODE_CFG), the power consumption in idle and standby state is low (current draw is I_{DD_ULP}) and only the following registers are accessible via SPI/I²C: MODE_CFG, PART_ID, DATA_STAT, FIFO_STAT, PRESS_OUT, TEMP_OUT, INT_STAT.

The use of this feature is recommended for every measurement mode except continuous mode.

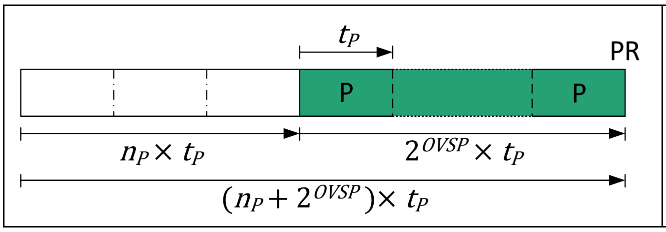
By setting HP to 1, the power consumption increases to I_{DD_LP} and all registers become available.

6.5 Precision and oversampling rate

6.5.1 Absolute Pressure

The precision of the pressure output can be optimized by adjusting the conversion time and the oversampling rate. The conversion times are multitudes of the (base) pressure conversion time t_{CONV_P} or the temperature conversion time t_{CONV_T} as specified in Table 2. The numbers below are typical values.

The ENS220 has a default conversion time t_P of 2 ms for pressure conversion. The conversion time can be selected via P_CONV to 1 ms, 2 ms or 4 ms. The first measurement after ADC power-up (i.e., after exiting standby mode or after a temperature measurement) requires 4 times the nominal duration (see Figure 6 and Figure 7).



$t_P = 1\,\text{ms}, 2\,\text{ms}$ or $4\,\text{ms}$

$n_P = 0$, if in continuous mode and previous measurement was a pressure measurement

$n_P = 3$, in all other cases

Figure 6: Timing diagram of a pressure pulse (values are typical)

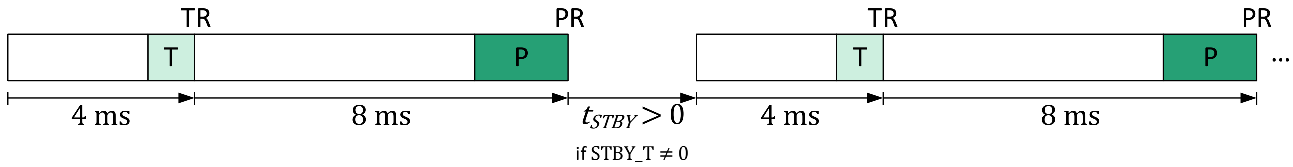


Figure 7: Typical timing for a continuous or pulsed measurement in default configuration (PT_RATE = 1, P_CONV = 1 (≈ 2 ms) and no oversampling)

Oversampling is available to reduce the measurement noise at the expense of conversion time. Between 1 and 128 measurements (see also OVS_CFG) are performed and averaged to calculate the output value. Every additional conversion for oversampling takes 1 ms, 2 ms or 4 ms, depending on the conversion time selected (Figure 8). The effect of oversampling on noise is shown in Figure 9.

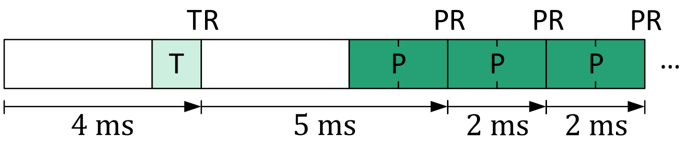


Figure 8: Typical timings for a continuous pressure-only measurement (MEAS_T = 0, P_CONV = 0) with activated 2× oversampling (OVSP = 1)

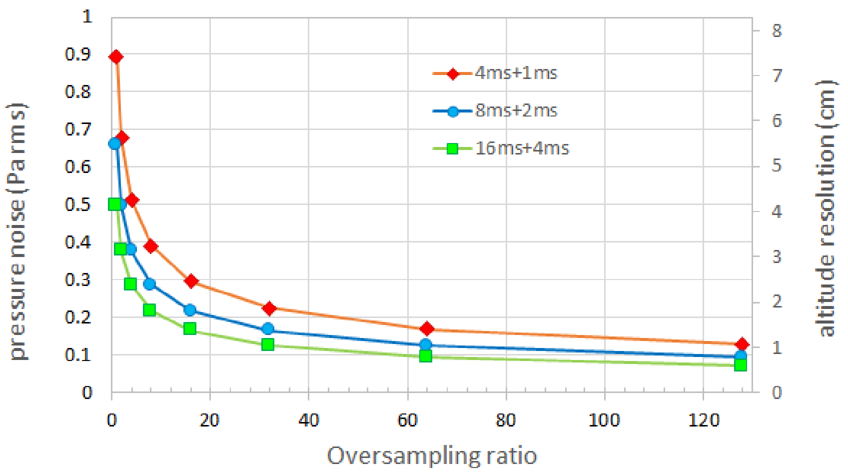
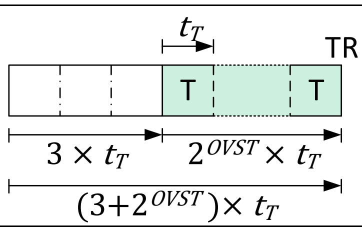


Figure 9: Pressure noise as function of oversampling ratio for three conversion time settings

The oversampled pressure value can either be transferred directly to the readout registers, can be recorded in a FIFO buffer, or can be further processed by a moving average filter. The desired path is selected by writing on MODE_CFG.FIFO_MODE (see chapter 6.6 and 6.8).

6.5.2 Temperature

The precision of the temperature output can be optimized by oversampling rate. Between 1 and 128 measurements (see also OVS_CFG) are performed and averaged to calculate the output value. The temperature conversion time is 4 ms. In case of oversampling, every additional conversion requires 1 ms (see Figure 10).



$t_T = 1$ ms

Figure 10: Timing diagram of a temperature pulse (values are typical)

Note: Each temperature measurement will take 3 ms plus the number of over-samples (Figure 11). This holds for any measurement mode, including temperature-only measurements.

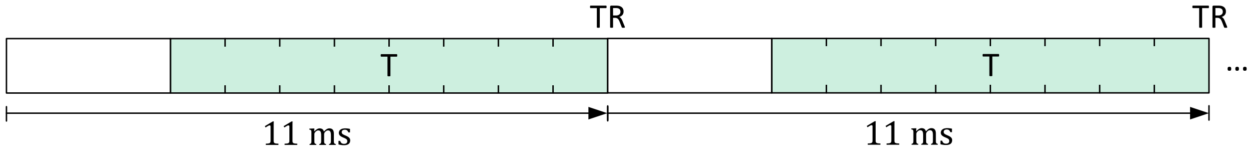


Figure 11: Typical timing of a continuous temperature-only measurement (MEAS_P=0) with 8× oversampling (OVST=3)

6.6 FIFO

A FIFO buffer is available to store 32 pressure values. It is enabled by MODE_CFG.FIFO_MODE.

The FIFO works as a circular buffer, with a write pointer to store new pressure values and a read pointer to retrieve them. The structure is shown in Figure 12. The write pointer increases whenever a new measurement is completed and the new value is written to the buffer. The pointer will increase even when the buffer is full, overwriting the oldest values recorded. Reads from PRESS_OUT_H increase the read pointer until it reaches the write pointer, subsequent reads will return 0. Both pointers wrap around (31 increments to 0).

The entire FIFO can be quickly read from PRESS_OUT_F. A continuous read past address 0x29 will cycle the memory read pointer back to 0x27, increasing at the same time the FIFO read pointer. In this way the entire FIFO buffer can be transferred without addressing overhead.

NOTE: In order to read more than one value it is necessary to set the HP bit in MODE_CFG.

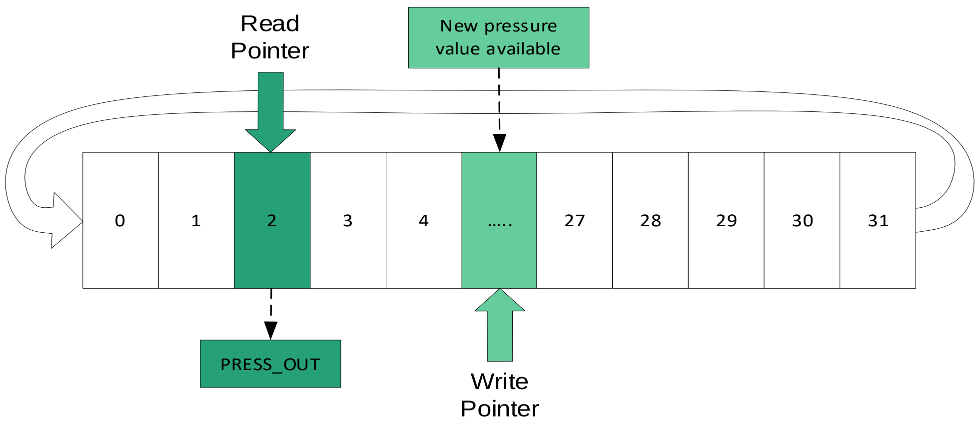


Figure 12: FIFO structure

FIFO operations are controlled by FIFO_CFG. The FIFO can be cleared by setting FP_CLEAR. The FIFO status is available in FIFO_STAT, and the current FIFO fill level (number of elements available) can be read from FP_FILL. When the FIFO is empty, FE is set. When the FIFO is full, FF is set. If a new pressure value arrives while FIFO is full, the overrun bit DATA_STAT.PO is set, and the oldest value is overwritten.

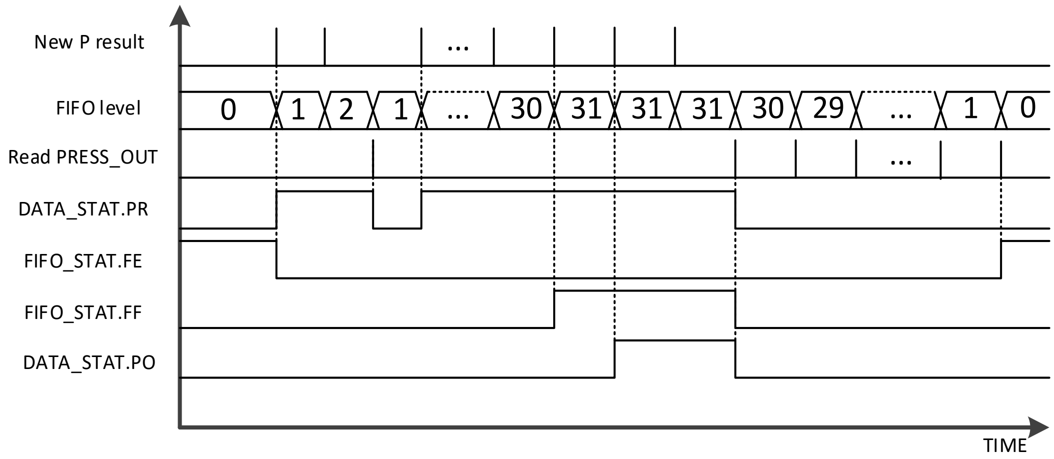


Figure 13: Sample FIFO timing diagram

Interrupts can be asserted when the FIFO is full, empty, or filled to a certain level. See chapter 6.7.

6.7 Interrupt

Interrupts can be enabled by writing on INT_CFG. Interrupts are available for the following events:

- Pressure measurement ready
- Temperature measurement ready
- FIFO level high
- FIFO full
- FIFO empty
- Pressure below PRESS_LO
- Pressure above PRESS_HI

Register INTF_CFG controls if the interrupt is signalled on the hardware pin INT. It also sets the polarity of this pin.

Once an interrupt is asserted, the interrupt source can be read in INT_STAT.

Reading INT_STAT will automatically clear all interrupt source flags, will clear IA, and will reset the INT/SDO pin value to the de-asserted state.

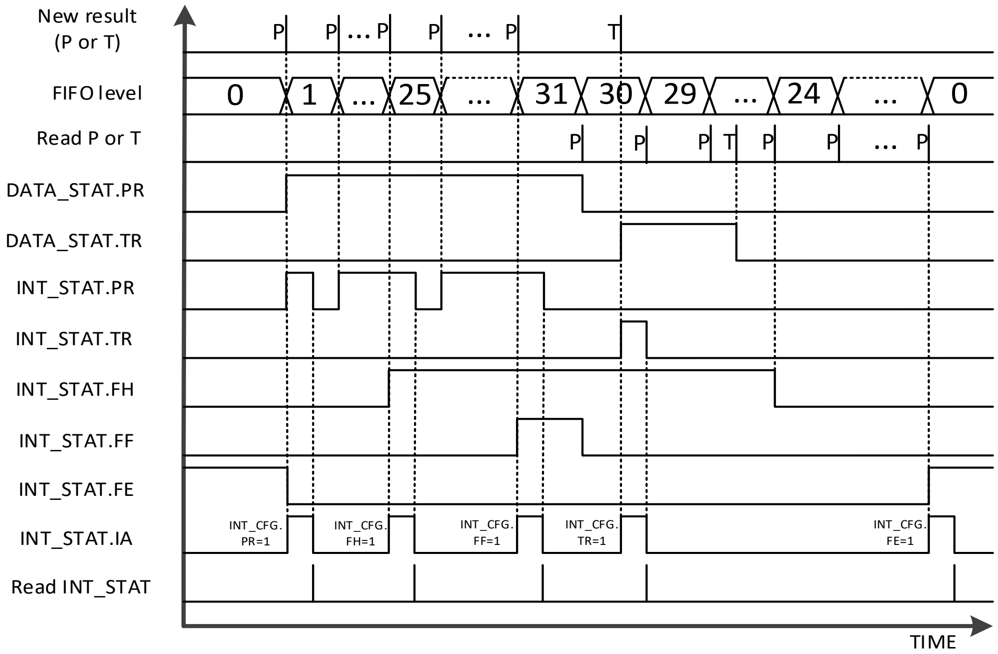


Figure 14: Sample interrupt timing diagram

6.8 Moving average

A moving average filter can be applied to pressure measurements. The filter is enabled by writing on MODE_CFG.FIFO_MODE. The FIFO buffer is used to implement this function, so FIFO is not available when the moving average filter is enabled.

The window size, i.e., the number of samples used by the moving average filter, is controlled by MAVG (1 to 32). It should only be changed while the moving average filter is disabled. After the moving average filter is enabled by setting MODE_CFG.FIFO_MODE = 2, the first pressure value available is used to initialize the window with that first pressure value.

NOTE: moving average settings are independent from pressure oversampling enabled via OVSP. The result of pressure oversampling is processed by the moving average filter.

7 Digital interface description

The ENS220 can be accessed as a slave device through either SPI 3-wire, SPI 4-wire, or I²C serial interface.

7.1 Interface selection

Selection between I²C and SPI is done through CSN.

If CSN is high, the I²C interface will be active and the SPI interface inactive. For systems where the host communicates via I²C to this device, connect CSN to VDD permanently.

A falling edge of CSN will disable the I²C interface until the next power-on cycle or software reset. The SPI host should generate a low pulse on CSN before communicating to other devices on the bus. The device may otherwise interpret SPI bus traffic as I²C commands.

7.2 I²C interface

The EN220 is an I²C slave device with a fixed 7-bit address 0x20.

The I²C interface supports fast (400 kbit/s) and high-speed (3400 kbit/s) mode. Details on I²C protocol is according to I²C-bus specifications [UM10204, I²C-bus specification and user manual, Rev. 6, 4 April 2014].

The device applies all mandatory I²C protocol features for slaves: START, STOP, Acknowledge, 7-bit slave address. None of the other optional features (10-bit slave address, general call, software reset or Device ID) are supported, nor are the master features (Synchronization, Arbitration, START byte).

The ENS220 uses a register model to interact with the host. This means that the I²C master can directly read or write values to one of the registers by first sending the single byte register address. The ENS220 implements an “auto increment” which allows reading or writing multiple bytes in a single transaction. This provides an effective way to read pressure or temperature values. See chapters 6.3, 6.6.

7.2.1 I²C write operation

The write operation is a single continuous transaction:

- The I²C master sends the START (S) condition which blocks the bus.
- The I²C Master sends the 7-bit slave address and 0 into the R bit (indicates a write transaction, the byte sent would be 0x40). The transaction will be acknowledged by the slave (ACK).
- The I²C Master then sends the address of the first register to write. The transaction will be acknowledged by the slave (ACK). (or not acknowledged (NACK) when the address is not writable)
- The I²C Master then sends one or more data bytes which are written into sequential registers (if valid) until the transaction is concluded with a STOP (P) condition.

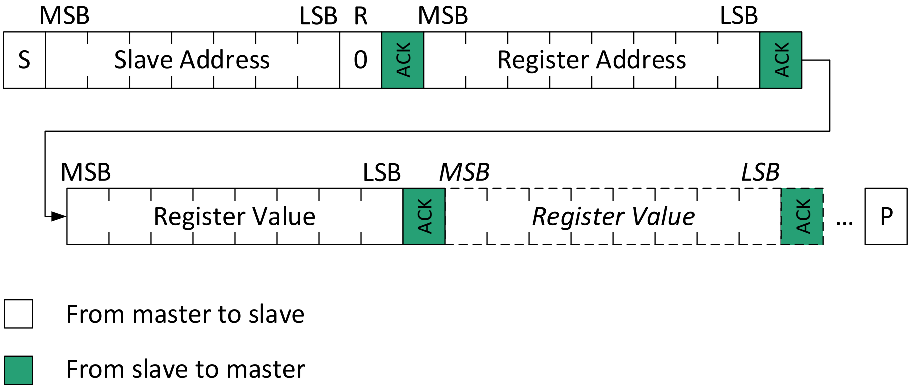


Figure 15: I²C Write operation

7.2.2 I²C read operation

A *read* transaction (see Figure 16) starts with a write (of the register address), followed by a read. Consequently, it has the following format:

The write specifies the register address:

- The I²C Master sends a START condition.
- The I²C Master sends the 7-bit slave address and 0 into the R bit (indicates a write transaction, the byte sent would be 0x40). The transaction will be acknowledged by the slave (ACK).
- The I²C Master then sends the address of the first register to read.

It is followed by the read sequence:

- The I²C Master sends again a START condition.
- The I²C Master sends the 7-bit slave address and 1 into the R bit (indicates a read transaction, the byte sent would be 0x41).
- The I²C slave then sends 1-n data bytes from sequential registers (if valid), each acknowledged by the master until the transaction is concluded with a STOP condition.

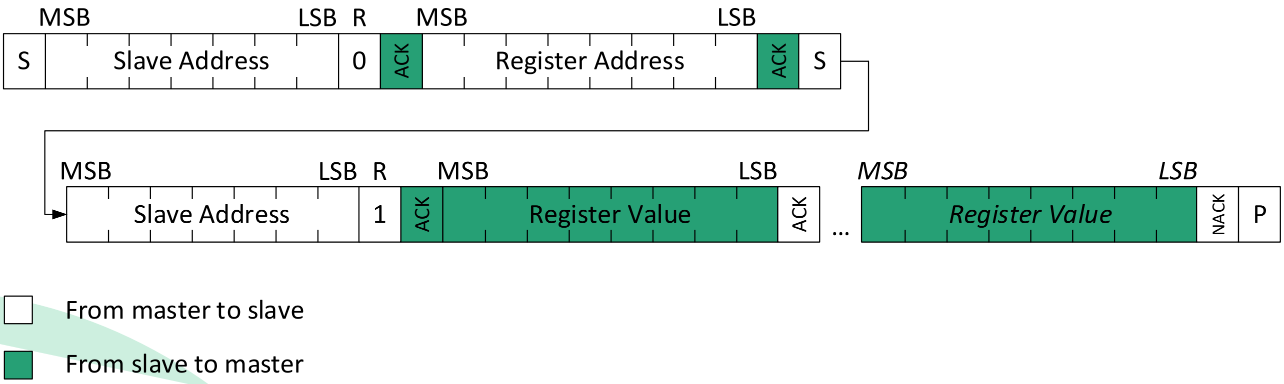


Figure 16: I²C read operation

7.2.3 High speed mode

The bus operation speed is limited to 400 kHz unless a high speed enable command (00001xxx) is issued by the master device as the first byte after START condition. This command is not acknowledged (NACK) by the slave. The high-speed operation allows data transfer frequencies up to 3.4 MHz. The input filters on the serial interface (SDA and SCL) are adapted to a higher bandwidth. After the high-speed command, the master transmits the slave address to invoke a data transfer. The bus keeps operating at the highest operating frequency until the master issues a STOP condition. Upon reception of the STOP condition by the slave, the input filters are switched to their initial time constants, which allow only up to 400 kHz transfer rates.

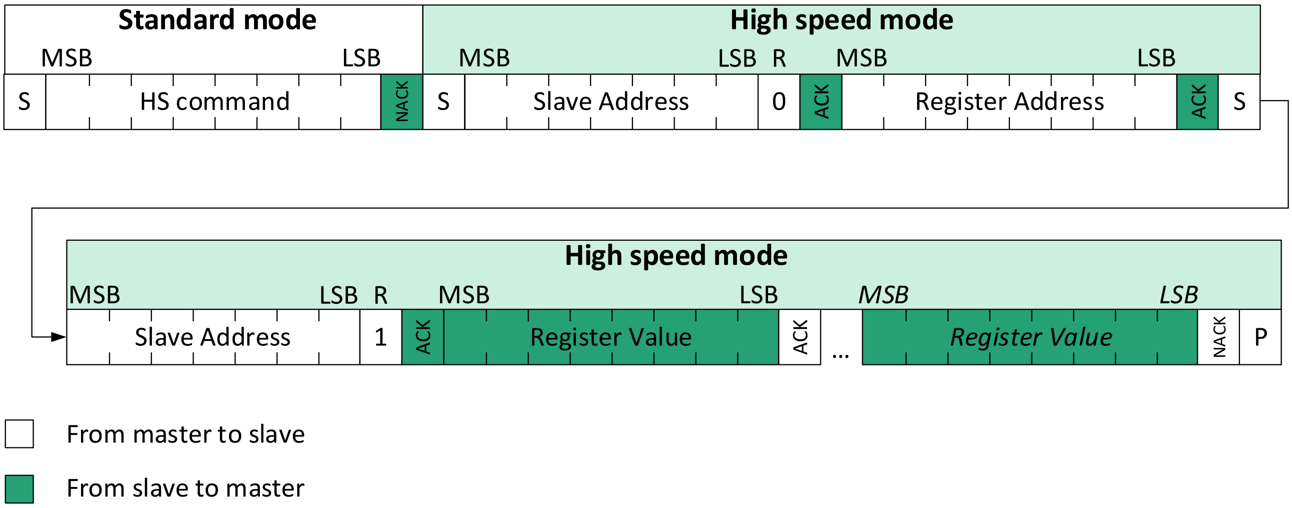


Figure 17: I²C high-speed read operation

7.2.4 Timing specifications

ENS220 is compliant to the I²C bus specifications [UM10204, I²C-bus specification and user manual, Rev. 6, 4 April 2014].

Table 7: ENS220 I²C timing parameters[9]

Parameter	Symbol	Units	Fast Mode		High-Speed Mode	
			Min	Max	Min	Max
SCL frequency	Fscl	kHz		400		3400
SCL low time	Twscll	µs	1.3		0.16	
SCL high time	Twsclh		0.6		0.06	
SDA setup time	Tsup	ns	100		10	
SDA hold time (host transmits)	Th	ns	0	-	0	-
SDA hold time (device transmits)	Tv	ns	40	900	20	160
SCL rise time	TrCL	ns		300	10	40
SCL fall time	TfCL			300	10	40

[9] All values referred to min V_{IH} and max V_{IL} levels in Table 3.

SDA rise time	TrDA			300	10	80
SDA fall time	TfDA			300	10	80
Start condition hold time	Thst	µs	0.6		0.16	
Repeated start condition setup time	Tsupst		0.6		0.16	
Stop condition setup time	Tsupsp		0.6		0.16	
Bus free time between start-stop	Twspst		1.3		0.16	
Load capacitance	CL	pF		400		100

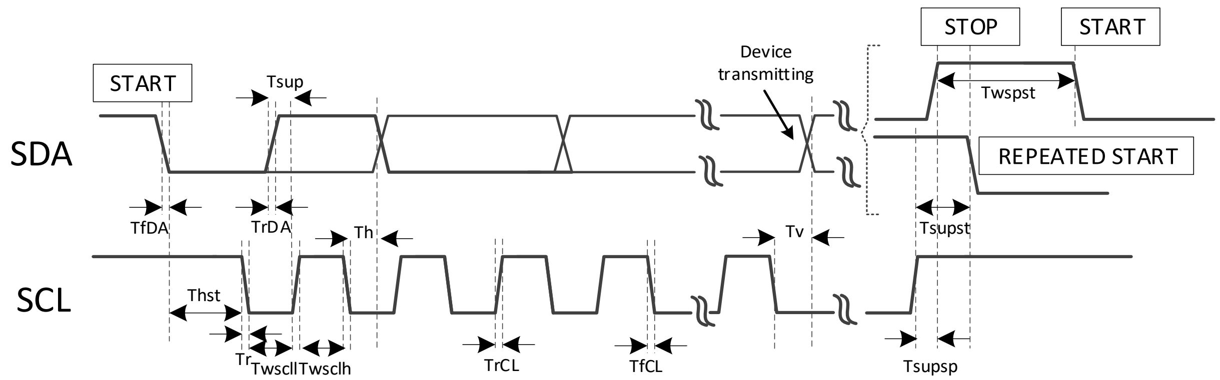


Figure 18: I^2C timing diagram

7.3 SPI interface

The ENS220 is an SPI bus slave. The SPI allows the user to write and read the registers of the device. The serial interface interacts with the outside world by four wires: CSN, SCLK, SDI and SDO. The optional three-wire mode uses the single bi-directional data line SDI instead of the two lines SDI and SDO.

Note: Use the SPI three-wire mode if multiple devices are to be used on the same SPI bus. The ENS220 pulls the SDO line low when CSN is high, preventing communication with other devices on the bus.

CSN is the serial port enable and it is controlled by the SPI master. It is driven low at the start of the SPI frame and returns high at the end.

SCLK is the serial port clock and is controlled by the SPI master. It should stay high in the absence of transmissions. SDI and SDO are respectively the serial port data input, and the output. Those lines are driven at the falling edge of SCLK and should be captured at the rising edge of SCLK.

The SDO pin is shared between SPI output data and interrupt function. When CSN is high and interrupt functionality is enabled, SDO will provide interrupt service.

If interrupts are disabled, then SDO stays low until data is requested from the device, i.e., during an SPI transaction.

The read register and write register commands are both completed in 16 clock pulses or in multiples of 8 in the case of multiple bytes read or write. The bit duration is the time between two falling edges of SCLK. The first bit (bit 0) starts at the first falling edge of SCLK after the falling edge of CSN while the last bit (bit 15, bit 23, ...) starts at the last falling edge of SCLK just before the rising edge of CSN.

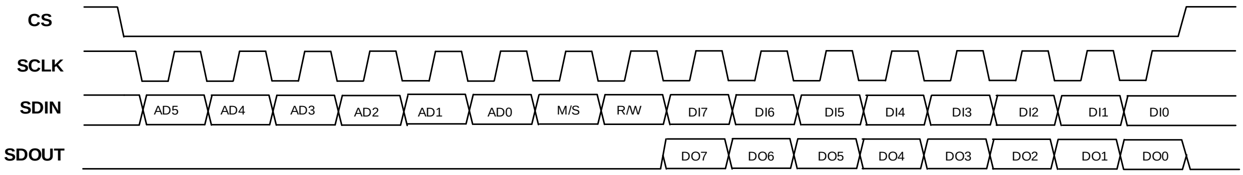


Figure 19: Minimum SPI frame

The device can also work with a low idle value of SCLK (also known as SPI mode 0), as shown below.

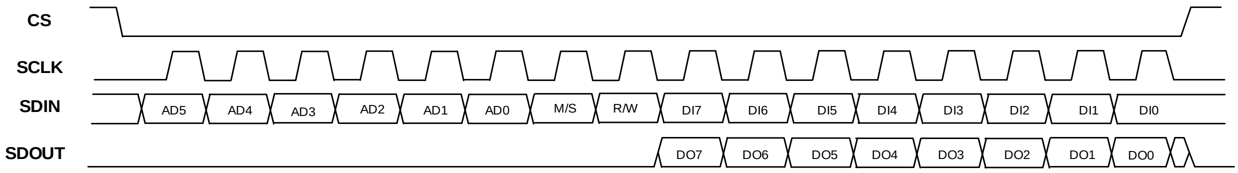


Figure 20: SPI frame using mode 0

A standard SPI frame is organized as follows in Table 8.

Table 8: SPI frames for reading and writing registers

Byte	Bit	Name	Description	
			read	write
0	7:2	AD[5:0]	Address of the indexed register	
0	1	MS	0: address will be auto incremented in multiple read/write commands. 1: no auto-incrementing of the address	
0	0	RW	1: data D*i* is read from the device. SDO becomes active at bit 8.	0: data D*i* is written to the device.
1	15:8	D1[7:0]	Data output from the device (MSB first) to pin SDO.	Data written to the device from pin SDI (MSB first)
…				
n	(8×n+7): 8×n	D*n*	*n*-th byte output to pin SDO (MSB first)	*n*-th byte input from pin SDO (MSB first)

In multiple read/write commands, further blocks of 8 clock periods are added. When the MS bit is 0, the address used to read/write data remains the same for every block. When MS bit is 1, the address used to read/write data is increased at every block. The function and the behaviour of SDI and SDO remain unchanged.

7.3.1 SPI write operation

The SPI single byte write command consists of 16 clock pulses. A multiple byte write is performed by adding blocks of 8 clock pulses to the single byte write.

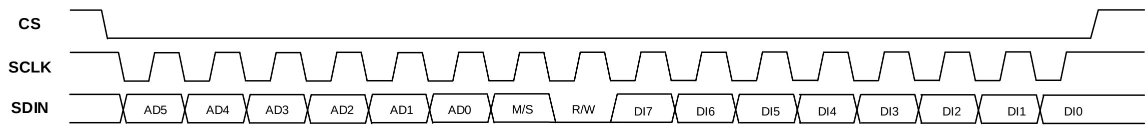


Figure 21: Single byte write

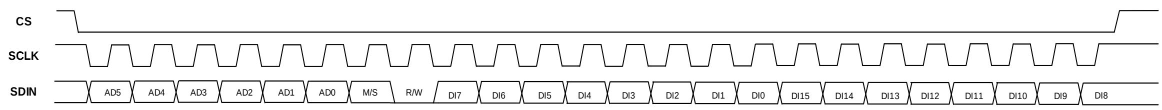


Figure 22: Multiple bytes write (2 bytes)

7.3.2 SPI read operation

The SPI single byte read command consists of 16 clock pulses. A multiple byte read is performed by adding blocks of 8 clock pulses to the single byte read. All register data is copied into an intermediate buffer at the beginning of a read transaction; multiple reads in the same read transaction will refer to the intermediate buffer.

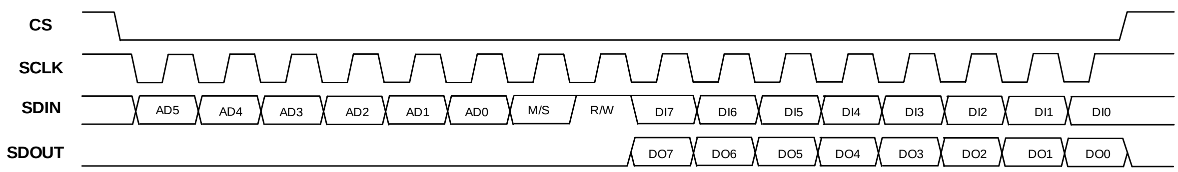


Figure 23: SPI read timing diagram

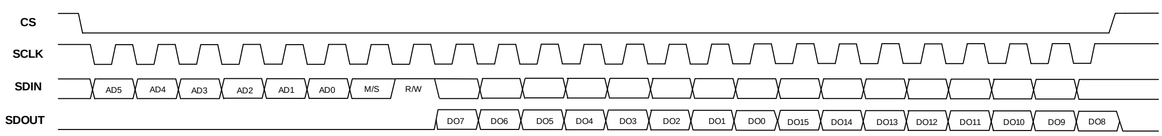


Figure 24: Multiple byte SPI read (2 bytes)

A 3-wire SPI mode can be selected by setting bit SPI3 in the INTF_CFG register. In this case, SDA acts as a bi-directional data line, and should be connected to an external pull-up resistor. Multiple byte read is also available in 3-wire mode.

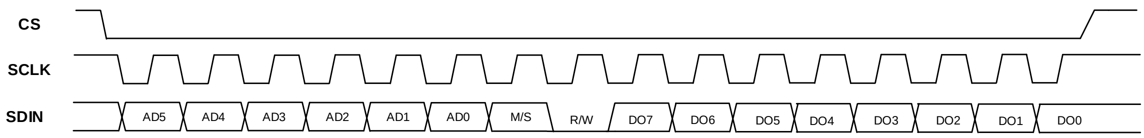


Figure 25: SPI read in 3-wire mode

7.3.3 Timing specifications

Description	Parameter	Min	Max	Units
SPI clock frequency	Fsclk		10	MHz
1/(SPI clock frequency)	Tsclk	100		ns
CSN falling edge to SCLK falling edge	Tdel	25		ns
SCLK rising edge to CSN rising edge	Tquiet	25		ns
CSN rising edge to SDO disabled	Tdis		50	ns
CSN deassertion between SPI communications	Tcsn_dis	100		ns
SCLK low pulse width	Ts	$0.4 \times t_{SCLK}$		ns
SCLK high pulse width	Tm	$0.4 \times t_{SCLK}$		ns
SDI valid before SCLK rising edge	Tsetup	25		ns
SDI valid after SCLK rising edge	Thold	25		ns
SCLK falling edge to SDO output transition	Tsdo		40	ns
SDO output low to output high transition	Tr		11.5	ns
SDO output high to output low transition	Tf		11.5	ns
Capacitance load at SDO pin	CL		25	pF

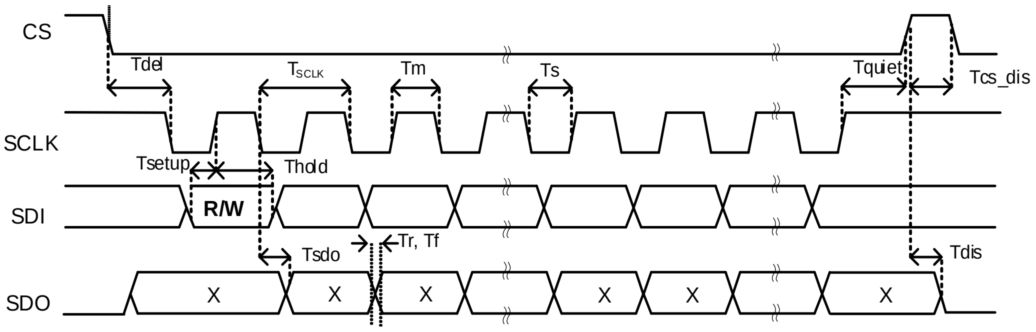


Figure 26: SPI timing diagram

8 Registers

This section describes the registers of the ENS220.

8.1 Register map

Table 9: Register map

Adr.	Name	Default	R/W	7	6	5	4	3	2	1	0	Description
0x00	PART_ID0	0x21	Rr	PART_ID0								Part ID [7:0]
0x01	PART_ID1	0x03	Rr	PART_ID1								Part ID [15:8]
0x02	UID0		R	UID0								Unit ID [7:0]
0x03	UID1		R	UID1								Unit ID [15:8]
0x04	UID2		R	UID2								Unit ID [23:16]
0x05	UID3		R	UID3								Unit ID [31:24]
0x06	MODE_CFG	0x03	RWrw	HP	FIFO_MODE		START	RESET	0	MEAS_T	MEAS_P	Device configuration
0x07	MEAS_CFG	0x08	RW	0	0		P_CONV		PT_RATE			Conversion time, P/T rate
0x08	STBY_CFG	0x00	RWw	0				STBY_T				Standby time configuration
0x09	OVS_CFG	0x00	RW	0		OVSP			OVST			Oversampling settings
0x0A	MAVG_CFG	0x00	RW	0					MAVG			Moving average config.
0x0B	INTF_CFG	0x00	RWw	0					INT_EN	INT_HL	SPI3	Interface configuration
0x0C	INT_CFG	0x7F	RWw	0	TR	FH	FF	FE	PR	PH	PL	Interrupt mask configuration
0x0D	PRESS_LO_XL	0x00	RW	PRESS_LO_XL								Low press. threshold [7:0]
0x0E	PRESS_LO_L	0x00	RW	PRESS_LO_L								Low press. threshold [15:8]
0x0F	PRESS_LO_H	0x00	RW	PRESS_LO_H								Low press. threshold [23:16]
0x10	PRESS_HI_XL	0xFF	RW	PRESS_HI_XL								High press. threshold [7:0]
0x11	PRESS_HI_L	0xFF	RW	PRESS_HI_L								High press. threshold [15:8]
0x12	PRESS_HI_H	0xFF	RW	PRESS_HI_H								High P threshold [23:16]
0x13	FIFO_CFG	0x00	RW	0		FP_CLEAR	FP_FILL_TH					FIFO configuration
0x14	DATA_STAT	0x00	Rr	0				PO	TO	PR	TR	Measurement data status
0x15	FIFO_STAT	0x02	R	FP_FILL					FF	FE	FH	FIFO status
0x16	INT_STAT	0x08	Rr	IA	TR	FH	FF	FE	PR	PH	PL	Interrupt status
0x17	PRESS_OUT_XL	0x00	Rry	PRESS_OUT_XL								Pressure value [7:0]
0x18	PRESS_OUT_L	0x00	Rry	PRESS_OUT_L								Pressure value [15:8]
0x19	PRESS_OUT_H	0x00	Rry	PRESS_OUT_H								Pressure value [23:16]
0x1A	TEMP_OUT_L	0x00	Rr	TEMP_OUT_L								Temperature value [7:0]
0x1B	TEMP_OUT_H	0x00	Rr	TEMP_OUT_H								Temperature value [15:8]
0x27	PRESS_OUT_F_XL	0x00	RFry	PRESS_OUT_F_XL								Pressure value [7:0]
0x28	PRESS_OUT_F_L	0x00	RFry	PRESS_OUT_F_L								Pressure value [15:8]
0x29	PRESS_OUT_F_H	0x00	RFry	PRESS_OUT_F_H								FIFO Pressure value [23:16]

Legend: *(the letters in the R/W column define the colours of the register fields)*

W	Write access in low power mode	w	Write access in ultra-low power mode	0	Must write 0
R	Read access in low power mode	r	Read access in ultra-low power mode	0	Reads 0
F	Address counter wraps. See text.	y	Read only the last value in ultra-low power mode		

Note that some registers are spread over multiple addresses. For example, PART_ID at address 0 is spread over 2 addresses (its "Size" is 2). Registers are stored in little endian so the LSB of PART_ID is at address 0 and the MSB of PART_ID is at address 1.

8.2 Detailed register description

8.2.1 PART_ID (Address 0x00-0x01)

The value is available when the ENS220 is initialized after power-up.

Table 10: Register PART_ID

PART_ID					
Address	**Bits**	**Field Name**	**Default**	**Access**	**Field Description**
0x00	7:0	PART_ID0	0x21	Rr	Least significant byte (LSB) of Part ID
0x01	15:8	PART_ID1	0x03	Rr	Most significant byte (MSB) of Part ID

8.2.2 UID (Address 0x02-0x05)

32 bit unique device identifier

Table 11: Register UID

UID					
Address	**Bits**	**Field Name**	**Default**	**Access**	**Field Description**
0x02	7:0	UID0		R	Least significant byte (LSB) of UID
0x03	15:8	UID1		R	Second byte of UID
0x04	23:16	UID2		R	Third byte of UID
0x05	31:24	UID3		R	Most significant byte (MSB) of UID

8.2.3 MODE_CFG (Address 0x06)

Table 12: Register MODE_CFG

Address 0x06		MODE_CFG			Default: 0x03
Bits	**Field Name**	**Default**	**Access**	**Field Description**	
7	HP	0b0	RWrw	High Power Bit 0b1: All registers are accessible via SPI/I²C; power consumption is high. A delay of 0.5 ms is required from high power enable to the next SPI/I²C access. 0b0: The following registers are accessible via SPI/I²C: MODE_CFG, PART_ID[1:0], DATA_STAT, FIFO_STAT, PRESS_OUT, TEMP_OUT, INT_STAT. Power consumption is low.	
6:5	FIFO_MODE	0b00	RWrw	Pressure data path 0b00: Direct path 0b01: FIFO 0b10: Moving average	
4	START	0b0	RWrw	Operating mode configuration 0b0: Stop measurements (idle mode) 0b1: Start measurements (measurement mode) Idle Mode is intended for configuration before running an active sensing mode. This bit is automatically reset in case of one-shot operation, after the required measurements have been performed	
3	RESET	0b0	RWrw	Device reset 0b1: The device is reset to the power-on configuration. RESET is automatically cleared. Device reset should be performed with HP bit at 0; the suggested reset sequence is: 0x08 → wait 0.5 ms → 0x80 → wait 0.5 ms → configure device	
2	X	0b0	RWrw	Reserved – must be set to default value	
1	MEAS_T	0b1	RWrw	See Table 13	
0	MEAS_P	0b1	RWrw		

Table 13: Measurement selection with MEAS_T and MEAS_P

MEAS_T	MEAS_P	
0	1	Only pressure measurements are enabled; the device will begin with a temperature measurement, then it will continue measuring only pressure.
1	0	Only temperature measurement is enabled.
1	1	Pressure and temperature measurements are enabled. PT_RATE controls the temperature interleaving timer.

8.2.4 MEAS_CFG (Address 0x07)

Table 14: Register MEAS_CFG

Address 0x07		MEAS_CFG		Default: 0x08
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:5	X	0b0	RW	Reserved – needs to be set to default value
4:3	P_CONV	0b01	RW	Pressure ADC conversion time. See Table 15.
2:0	PT_RATE	0b000	RW	Determines the ratio between P and T measurements as produced by the measurement engine. See Table 16.

Table 15: Typical pressure ADC conversion time

P_CONV[1:0]	Conversion time [ms] First conversion	Conversion time [ms] Next conversions
0	4	1
1	8	2
2	16	4

Table 16: PT_RATE

PT_RATE[2:0]	P/T rate
0	1
1	4
2	8
3	16
4	32
5	64
6	128
7	256

8.2.5 STBY_CFG (Address 0x08)

Table 17: Register STBY_CFG

Address 0x08		STBY_CFG			Default: 0x00
Bits	Field Name	Default	Access	Command	
7:4	X	0x0	RWw	Reserved, needs to be set to default value	
3:0	STBY_T	0x0	RWw	Each measurement will be followed by a standby phase (see Table 18).	

Table 18: Standby pauses in-between measurements

STBY_T[3:0]	Standby duration (ms)
0	Continuous operation
1	One-shot operation (device returns to idle after one measurement is produced by the measurement engine)
2	10
3	20
4	30
5	50
6	100
7	250
8	500
9	750
10	1000
11	2000
12	5000
13	10000
14	60000
15	600000

8.2.6 OVS_CFG (Address 0x09)

Oversampling measures the average over of a certain number of samples. Oversampling applies to all measurement modes.

Table 19: Register OVS_CFG

Address 0x09		OVS_CFG			Default: 0x00
Bits	**Field Name**	**Default**	**Access**	**Field Description**	
7:6	X	0b00	RW	Reserved, must be set to default value	
5:3	OVSP	0b000	RW	Oversampling of pressure measurements (see Table 20)	
2:0	OVST	0b000	RW	Oversampling of temperature measurements (see Table 21)	

Table 20: Oversampling of pressure measurements

OVSP[5:3]	Number of averages
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Table 21: Oversampling of temperature measurements

OVST[2:0]	Number of averages
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

8.2.7 MAVG_CFG (Address 0x0A)

Moving average configuration

Table 22: Register MAVG_CFG

Address 0x0A		MAVG_CFG			Default: 0x00
Bits	**Field Name**	**Default**	**Access**	**Field Description**	
7:3	X	0x00	RW	Reserved, must be set to default value	
2:0	MAVG	0b00	RW	Controls the number of samples used by the moving average filter. See Table 23.	

Table 23: Moving average configuration

MAVG[2:0]	Samples
0	1
1	2
2	4
3	8
4	16
5	32
6	32
7	32

8.2.8 INTF_CFG (Address 0x0B)

Host interface configuration register

Table 24: Register INTF_CFG

Address 0x0B		INTF_CFG			Default: 0x00
Bits	**Field Name**	**Default**	**Access**	**Field Description**	
7:3	X	0x0	RWw	Reserved, must be set to default value	
2	INT_EN	0b0	RWw	Interrupt enable 0b1: INT/SDO is controlled by INT_STAT.IA 0b0: INT/SDO is always low	
1	INT_HT	0b0	RWw	Interrupt polarity 0b1: INT/SDO low signals that interrupt is asserted 0b0: high signals that interrupt is asserted	
0	SPI3	0b0	RWw	SPI mode control 0b1: SPI works in 3-wire mode 0b0: SPI works in 4-wire mode	

8.2.9 INT_CFG (Address 0x0C)

Interrupt configuration register.

Table 25: Register INT_CFG

Address 0x0C		INT_CFG		Default: 0x7F
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7	X	0b0	RWw	Reserved, must be set to default value
6	TR	0b1	RWw	Temperature data is ready 0b1: interrupt is triggered when new temperature data becomes available 0b0: interrupt is disabled
5	FH	0b1	RWw	Pressure FIFO reached level high 0b1: interrupt is triggered when the filling level of the pressure FIFO becomes greater than the programmed level in FP_FILL_TH 0b0: interrupt is disabled
4	FF	0b1	RWw	Pressure FIFO is full (32 values) 0b1: interrupt is triggered when the pressure FIFO becomes full 0b0: interrupt is disabled
3	FE	0b1	RWw	Pressure FIFO is empty 0b1: interrupt is triggered when the pressure FIFO becomes empty 0b0: interrupt is disabled
2	PR	0b1	RWw	Pressure data is available 0b1: interrupt is triggered when new pressure data becomes available 0b0: interrupt is disabled
1	PH	0b1	RWw	Pressure is high 0b1: interrupt is triggered when the pressure is greater than the programmed high-pressure threshold (PRESS_HI). 0b0: interrupt is disabled
0	PL	0b1	RWw	Pressure is low 0b1: interrupt is triggered when the pressure is lower than the programmed low-pressure threshold (PRESS_LO). 0b0: interrupt is disabled

8.2.10 PRESS_LO (Address 0x0D-0x0F)

This 3-byte register sets the pressure threshold for the low-pressure interrupt.

Table 26: Register PRESS_LO

PRESS_LO					
Address	Bits	Field Name	Default	Access	Field Description
0x0D	7:0	PRESS_LO_XL	0x00	RW	Least significant byte (LSB) of PRESS_LO
0x0E	15:8	PRESS_LO_L	0x00	RW	Middle Byte of PRESS_LO
0x0F	23:16	PRESS_LO_H	0x00	RW	Most significant byte (MSB) of PRESS_LO

8.2.11 PRESS_HI (Address 0x10-0x12)

This 3-byte register sets the pressure threshold for the high-pressure interrupt.

Table 27: Register PRESS_HI

PRESS_HI					
Address	Bits	Field Name	Default	Access	Field Description
0x10	7:0	PRESS_HI_XL	0x00	RW	Least significant byte (LSB) of PRESS_HI
0x11	15:8	PRESS_HI_L	0x00	RW	Middle Byte of PRESS_HI
0x12	23:16	PRESS_HI_H	0x00	RW	Most significant byte (MSB) of PRESS_HI

8.2.12 FIFO_CFG (Address 0x13)

FIFO configuration register

Table 28: Register FIFO_CFG

Address 0x13		FIFO_CFG		Default: 0x00
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:6	X	0b00	RW	Reserved, must be set to default value
5	FP_CLEAR	0x00	RW	FIFO clear. 0b1: the content of the FIFO is cleared; FP_CLEAR is automatically cleared after the operation. 0b0: no operation
4:0	FP_FILL_TH	0x00	RW	FIFO level threshold. The value is used as target for interrupt and status generation.

8.2.13 DATA_STAT (Address 0x14)

Data status register

Table 29: Register DATA_STAT

Address 0x14		DATA_STAT		Default: 0x00
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:4	X	0x0	Rr	Reserved
3	PO	0b0	Rr	Pressure overwrite With FIFO enabled (MODE_CFG.FIFO_MODE=1), this bit is set whenever a new pressure measurement is produced by the measurement engine while the pressure FIFO was already full. PO is cleared after reading PRESS_OUT_H or PRESS_OUT_F_H. With FIFO disabled, PO is undefined.
2	TO	0b0	Rr	Temperature overwrite This bit is set when a new temperature measurement is produced by the measurement engine and the previous data was not read. TO is cleared after reading TEMP_OUT_H.
1	PR	0b0	Rr	Pressure ready. This bit is set when new pressure data becomes available. PR is cleared after reading PRESS_OUT_H.
0	TR	0b0	Rr	Temperature ready This bit is set when new temperature data becomes available. TR is cleared after reading TEMP_OUT_H.

8.2.14 FIFO_STAT (Address 0x15)

FIFO status register

Table 30: Register FIFO_STAT

Address 0x15		FIFO_STAT			Default: 0x02
Bits	**Field Name**	**Default**	**Access**	**Field Description**	
7:3	FP_FILL	0x0	R	FIFO level Fill level of the pressure FIFO. FP_FILL is equal to 0x1F when FIFO has 31 but also 32 elements.	
2	FF	0b0	R	FF is set when the FIFO is enabled and is full (32 elements).	
1	FE	0b1	R	FE is set when the FIFO is enabled and is empty (0 elements).	
0	FH	0b0	R	FH is set when is enabled and the number of elements is greater than FP_FILL_TH.	

8.2.15 INT_STAT (Address 0x16)

Interrupt status register. Each bit except IA is set if the corresponding interrupt source has generated an event, even if the hardware interrupt is disabled by the INT_CFG register. Reading this register will clear all flags.

Table 31: Register FIFO_STAT

Address 0x16		FIFO_STAT			Default: 0x00
Bits	**Field Name**	**Default**	**Access**	**Field Description**	
7	IA	0b0	Rr	General interrupt flag IA is set if any of the interrupt flags in bits 6:0 is set and if the interrupt source is enabled in INT_CFG; see also INTF_CFG.INT_EN.	
6	TR	0b0	Rr	TR is set when a new temperature measurement is produced by the measurement engine	
5	FH	0b0	Rr	FH is set when FP_FILL > FP_FILL_TH.	
4	FF	0b0	Rr	FF is set when the FIFO becomes full.	
3	FE	0b0	Rr	FE is set when FIFO becomes empty as effect of a data read.	
2	PR	0b0	Rr	PR is set when new pressure measurement is produced (i.e. without considering oversampling; refer to Figure 22).	
1	PH	0b0	Rr	PH is set when the measurement engine produces a pressure measurement that is greater than PRESS_HI.	
0	PL	0b0	Rr	PL is set when the measurement engine produces a pressure measurement that is lower than PRESS _LO.	

8.2.16 PRESS_OUT (Address 0x17-0x19)

This 3-byte register contains a 24-bit unsigned integer representing the pressure in 1/64 Pa.

A read on this register extracts one element from the FIFO if the FIFO is enabled by FIFO_MODE = 1. When the FIFO is empty, the read returns 0x000000. HP = 1 must be set for reading from the FIFO.

When the FIFO is not enabled (bypass or moving average), reads from this register return the latest measurement result. If readouts occur faster than measurements, values are repeated.

To ensure a consistent value during readout, PRESS_OUT registers are double buffered. When PRESS_OUT_XL is read, the device copies all bytes from the internal measurement registers to the I²C registers, then reads are always directly from the I²C/SPI registers. The double buffering is thus only available if all PRESS_OUT registers are read within the same I²C or SPI transaction (see sections

7.2.2, 7.3.2). If the application does not support reading multiple bytes at once, then the user must ensure that the PRESS_OUT register is not updated during the reading. This can be achieved by reading quickly after an interrupt occurred, or by using the single-shot mode (STBY_CFG = 1).

Please note that the hardware implementation of this double buffering does not guarantee the alignment between data ready flags and data if they are accessed in the same I^2C/SPI transaction. It is advised to access the flags in a separate transaction.

Table 32: Register PRESS_OUT

PRESS_OUT					
Address	**Bits**	**Field Name**	**Default**	**Access**	**Field Description**
0x17	7:0	PRESS_OUT_XL	0x00	Rry	Least significant byte (LSB) of PRESS_OUT
0x18	15:8	PRESS_OUT_L	0x00	Rry	Middle byte of PRESS_OUT
0x19	23:16	PRESS_OUT_H	0x00	Rry	Most significant byte (MSB) of PRESS_OUT

Example: For a stored PRESS_OUT value of 0x62F340 the absolute pressure in Pa is calculated as follows:

$$P = \frac{0x062F340}{64}\ \text{Pa} = 101325\ \text{Pa}$$

See section 6.3.1 for further information.

8.2.17 TEMP_OUT (Address 0x1A-0x1B)

This 2-byte register contains a 16-bit unsigned integer representing the temperature in 1/128 K.

Reads from this register return the latest measurement result. If reads occur faster than measurements, values are repeated.

To ensure a consistent value during readout, TEMP_OUT registers are double buffered. When TEMP_OUT_L is read, the device copies all bytes from the internal measurement registers to the I^2C/SPI registers, then readouts are always directly from the I^2C/SPI registers. The hardware implementation does to guarantee the alignment between data ready flags and data if they are accessed in the same I^2C or SPI transition; it is advised to access the flags separately from the data. But please note that double buffering only works for TEMP_OUT if TEMP_OUT_L and TEMP_OUT_H are read in the same I^2C or SPI transaction with auto-increment of the register address (see 7.2.2).

Table 33: Register TEMP_OUT

TEMP_OUT					
Address	Bits	Field Name	Default	Access	Field Description
0x1A	7:0	TEMP_OUT_L	0x00	Rr	Least significant byte (LSB) of TEMP_OUT
0x1B	15:8	TEMP_OUT_H	0x00	Rr	Most significant byte (MSB) of TEMP_OUT

Example: For a stored TEMP_OUT value of 0x9513, the temperature in degrees Celsius is calculated as follows:

$$T_C = \frac{0x9513}{128}\,\mathrm{K} - 273.15\,\mathrm{K} \approx 25\,°\mathrm{C}$$

See section 6.3.2 for further information.

8.2.18 PRESS_OUT_F (Address 0x27-0x29)

This register is the same as PRESS_OUT, except that reading in a single I^2C/SPI transaction wraps around from address 0x29 to 0x27. A single I^2C/SPI transaction can thus read multiple P measurements from the FIFO.

The HP bit of the MODE_CFG register must be 1 when reading this register to access the FIFO.

Table 34: Register PRESS_OUT_F

PRESS_OUT_F					
Address	Bits	Field Name	Default	Access	Field Description
0x27	7:0	PRESS_OUT_F_XL	0x00	RFry	Least significant byte (LSB) of PRESS_OUT_F
0x28	15:8	PRESS_OUT_F_L	0x00	RFry	Middle byte of PRESS_OUT_F
0x29	23:16	PRESS_OUT_F_H	0x00	RFry	Most significant byte (MSB) of PRESS_OUT_F

9 Application information and use case examples

9.1 I²C operation circuitry (1.8 V)

The recommended application circuit for the ENS220 I²C interface operation is shown in Figure 27.

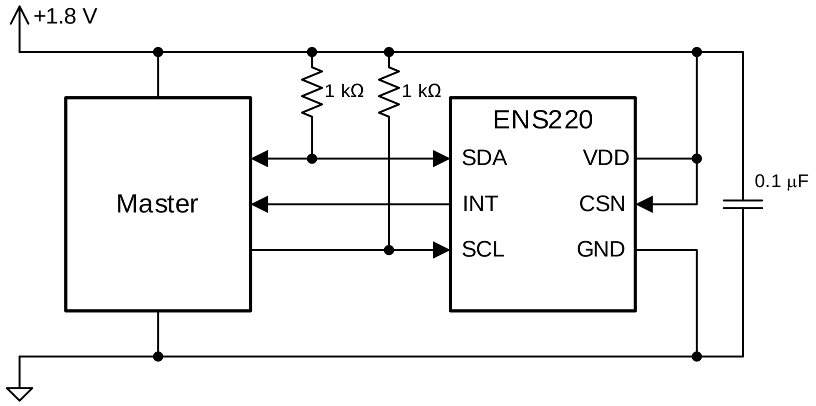


Figure 27: Recommended application circuit for I²C operation

Note(s):

1. CSN must always be high to ensure the I²C mode. CSN can be connected directly to VDD.
2. Pull-up resistors.
   The above recommendation for pull-up resistance values applies to I²C standard mode only. Pull-up resistors for SCL and SDA are assumed to be part of the host system and should be selected dependent on the intended I²C data rate and individual bus architecture.
3. A decoupling capacitor must be placed close to the VDD supply pin (pin 10) of the ENS220.
4. The INT pin is an output. If not used, it can be left unconnected.

9.2 I²C operation circuitry (3.3 V)

The I²C communication bus is 3.3 V tolerant. The power supply of the ENS220 requires a voltage regulator to 1.8 V.

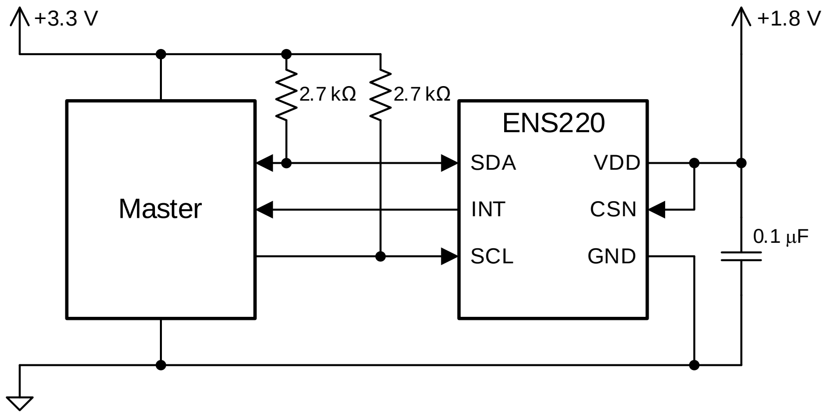


Figure 28: Recommended application circuit for I²C operation with 3.3 V

Note(s):

1. If V_{DD} is switched off, SCL and SDA need to be set to GND (0 V)
2. CSN must always be high to ensure the I^2C mode. CSN can be connected directly to VDD.
3. Pull-up resistors.
   The above recommendation for pull-up resistance values applies to I^2C standard mode only. Pull-up resistors for SCL and SDA are assumed to be part of the host system and should be selected dependent on the intended I^2C data rate and individual bus architecture.
4. A decoupling capacitor must be placed close to the VDD supply pin (pin 10) of the ENS220.
5. The INT pin is an 1.8 V output. If not used, it can be left unconnected.

9.3 SPI operation circuitry

9.3.1 Three wire SPI

The recommended application circuit for the ENS220 for three wire SPI interface is shown in Figure 29. The INT pin is an output for the interrupts. It can be left unconnected if it is not used.

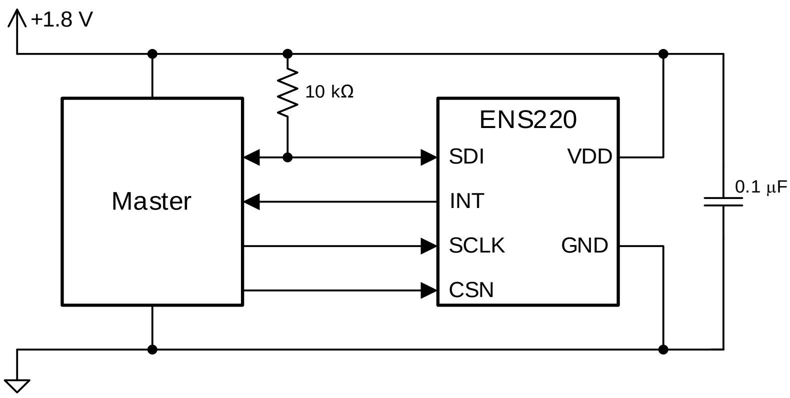


Figure 29: Recommended application circuit (three wire SPI operation)

9.3.2 Four wire SPI

The recommended application circuit for the ENS220 for four wire SPI interface is shown in Figure 30. The INT pin becomes the data output pin SDO.

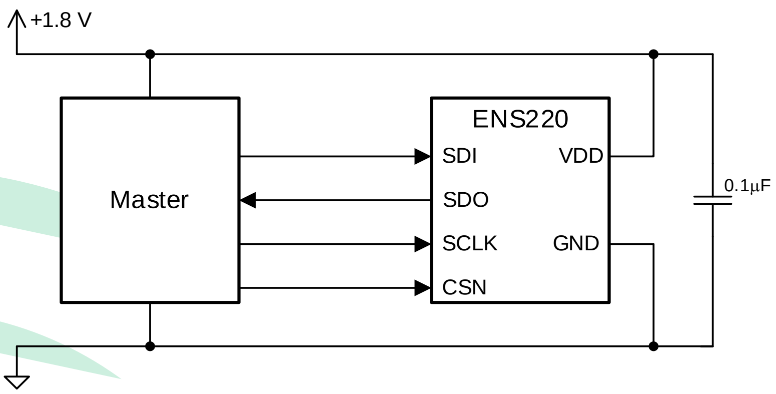


Figure 30: Recommended application circuit (four wire SPI operation)

9.4 Recommended settings for different use cases

Table 35 lists relevant use cases with the associated configuration parameters and the expected performance with these settings.

NOTE: current consumption due to the communication interface used (I^2C or SPI) is not included.

Table 35: Relevant use-cases with the associated configuration parameters and typical performance

Application examples	HP	P_CONV	PT_RATE	STBY_T	OVSP	OVST	Performance
Highest data rate - door/window opening detection	1	1 ms	128:1	0 ms (continuous)	1x	1x	1.8 Pa noise 1000 Hz 75 µA
Height tracking (drones, gaming, AR/VR)	1	4 ms	16:1	0 ms (continuous)	32x	8x	0.2 Pa noise 10 Hz 75 µA
Sports, activity tracking	1	4 ms	1:1	50 ms (pulsed)	4x	4x	0.5 Pa noise 10 Hz 30 µA
Home- and Building Automation, HVAC (filter clogging)	1	2 ms	1:1	1 s (pulsed)	4x	4x	0.7 Pa noise 1 Hz 1.7 µA
Lowest power - barometer, weather station	0	1 ms	1:1	60 s (pulsed)	1x	1x	1.5 Pa noise 1/60 Hz 0.8 µA
Lowest noise	1	4 ms	1:1	0 ms (continuous)	128x	128x	0.1 Pa noise 2 Hz

9.5 Sample communication sequence for common tasks

The following code is a functional example and does not include error checking, so it is not suited for production.

The code uses the following instructions:

- Write (reg, value): write one byte "value" via the I2C or SPI bus to the device at register address "reg". The I2C device address is 0x20 which is different from the register address "reg" that addresses a register within the device. See section 7 for details on how to address the device and registers within the device.

- Read(reg): read one byte from the device from register “reg”.
- Read N (reg, count): read “count” bytes from the device, starting at register “reg”. The register address is auto-incremented by the device.
- Wait(duration): Wait the specified amount of time (“duration”)

9.5.1 Basic example using the single shot measurement mode

```
Write(0x06, 0x08)       // soft reset, equivalent to power-up
Wait(1 ms)
// *** configuration: shortest measurement time and one-shot ***
// (default is 2 ms + continuous measurements)
Write(0x06, 0x80)       // enable all registers with bit HP=1
Wait(1 ms)
Write(0x07, 0x00)       // 1 ms conversion time
Write(0x08, 0x01)       // one-shot operation
Write(0x06, 0x00)       // switch back to ultra-low power mode HP=0
                        // (this saves power if the measurement is triggered later)
Wait(1 ms)        // wait a user-defined time until a measurement starts
// *** perform a measurement in ultra-low power mode ***
Write(0x06, 0x13)       // start T+P measurement, ultra-low power, no FIFO
Wait(10 ms)             // wait T+P measurement time (8 ms + margin)
// *** read the data ***
S = Read(0x14)          // check PR & TR in DATA_STAT to confirm that data is ready
If ((S & 3) != 3) {
      Print(“Warning: measurement was not complete” ) }
buffer = ReadN(0x17, 5) // read all five data bytes (register 0x17 to 0x1B)
// *** convert, and print the data ***
T = buffer[3] | (buffer[4] << 8)    // = buffer[0] + 256*buffer[1]
P = buffer[0] | (buffer[1] << 8) | (buffer[2] << 16)
Print(P/64.0, “Pa”)
Print(T/128.0 - 273.15, “C”)
```

9.5.2 Example using continuous mode with FIFO

```
Write(0x06, 0x08)       // soft reset, equivalent to power-up
Wait(1 ms)
Write(0x06, 0x80)       // enable all registers with HP bit
Wait(1 ms)
Write(0x07, 0x04)       // 1ms measurement time, PT-rate 32
Write(0x09, 0x00)       // OVSP 1X, OVST 1X
Write(0x0B, 0x04)       // interrupt enabled
Write(0x0C, 0x50)       // interrupt on TR and FIFO full
Write(0x06, 0xB3)       // start measurement, P+T, FIFO enabled
// *** Repeat as long as measurements are required ***
      Wait(INT==1)      // Wait until INT hardware pin is high
      S=Read(0x16)      // INT_STAT
      If (S & 0x40) {   // TR flag: temperature ready
```

```
            buffer = ReadN(0x1A, 2)                // read TEMP_OUT bytes
            T = buffer[0] | (buffer[1] << 8)       // = buffer[0] + 256*buffer[1]
            Print(T/128.0 - 273.15, “C”)
      }
      If (S & 0x10) {                        // FF flag: FIFO is full
            buffer = ReadN(0x27, 32*3)       // read all data in FIFO
            for(i=0; i<32*3; i=i+3) {
               P = buffer[i] | (buffer[i+1] << 8) | (buffer[i+2] << 16)
            Print(P/64.0, “Pa”)
            }
      }
// *** Turn off when measurements are not needed ***
Write(0x06, 0)          //HP=0: put device in ultra-low power mode
```

10 Soldering information

The ENS220 uses an open LGA package. This package can be soldered using a standard reflow process in accordance with IPC/JEDEC J-STD-020E (Figure 31). Devices have been verified against MSL1 storage conditions. Deviation from the recommended solder profile may result in degraded performance of the sensor. It is recommended to use a no-clean solder paste. There should not be any board washing processes, to prevent cleaning agents or other liquid materials contacting the sensor area.

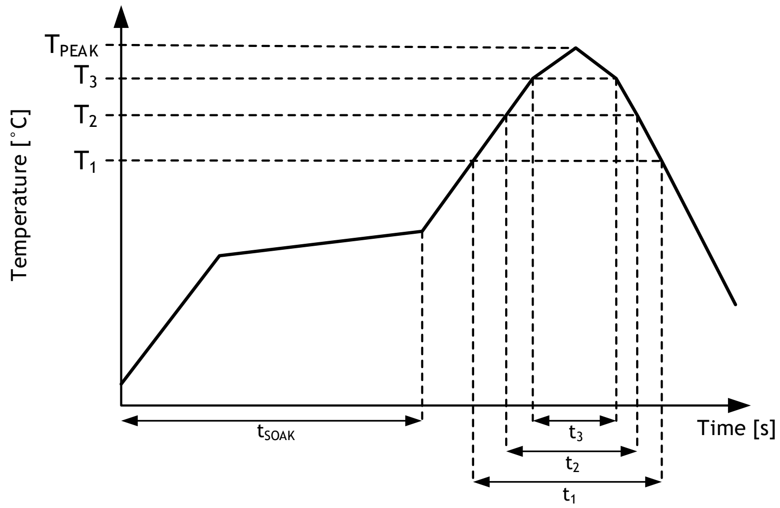


Figure 31: Graph of the solder reflow profile

The detailed settings for the reflow profile are shown in Table 36.

Table 36: Parameters of the solder reflow profile

Parameter	Reference	Value
Average temperature gradient in preheating		2.5 K/s
Soak time	t_{SOAK}	2..3 min
Soak temp range	Ts max	200 °C
	Ts min	150 °C
Time above 217 °C (T1)	t_1	Max. 60 s
Time above 230 °C (T2)	t_2	Max. 50 s
Time above TPEAK -10 °C (T3)	t_3	Max. 10 s
Peak temperature in reflow	T_{PEAK}	260 °C
Maximum ramp-down rate for cooling to T1		Max. 5 K/s

11 Package drawings & markings

11.1 Package outline dimensions

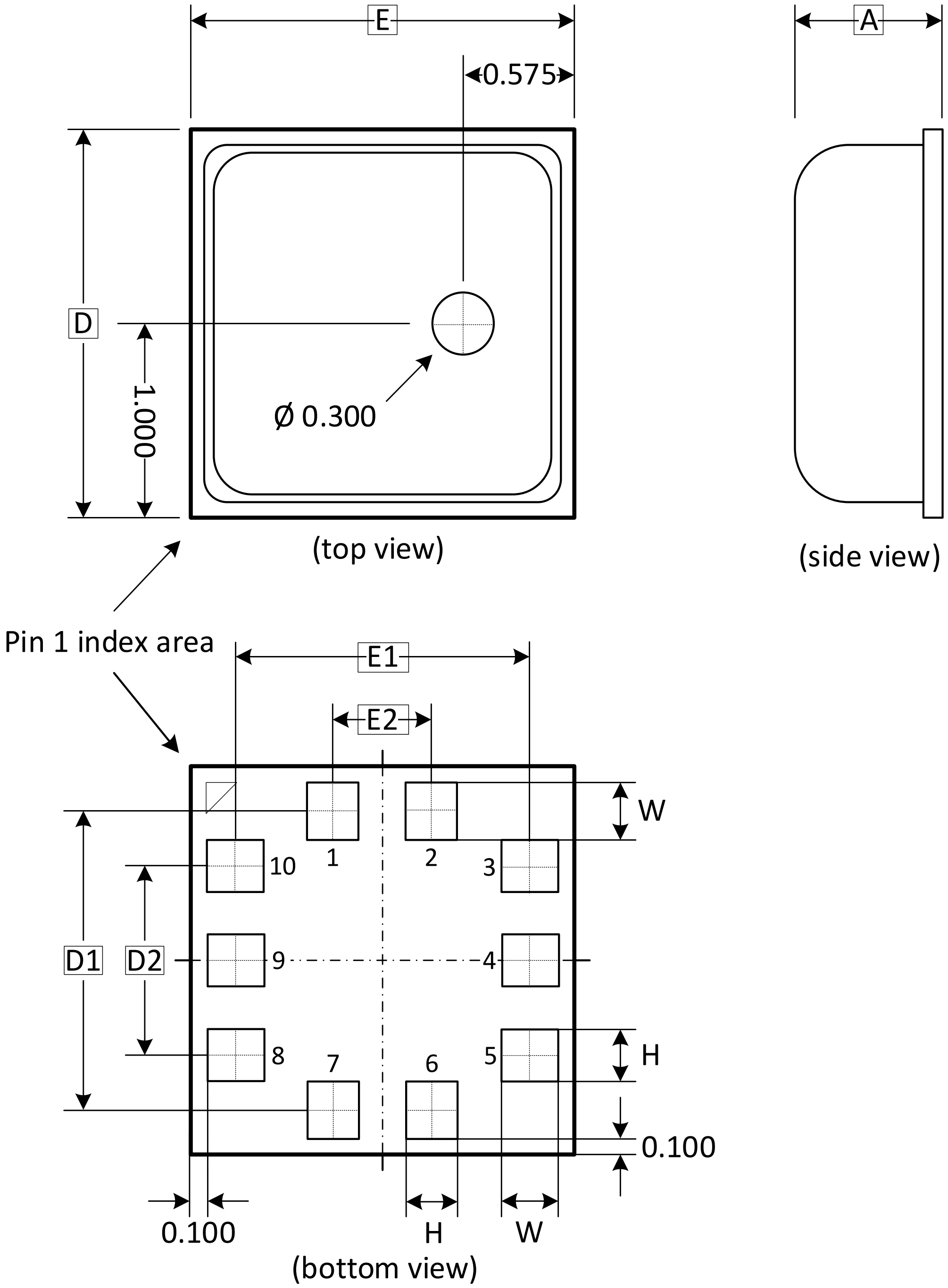


Figure 32: LGA package drawing

Table 37: LGA package dimensions in mm

Parameter	Symbol	Dimensions		
		Min	Nominal	Max
Total thickness	A	0.70	0.75	0.80
Body Size	D	1.95	2.00	2.05
	E	1.95	2.00	2.05
Lead Width	W		0.275	
Lead Length	H		0.250	
Lead Count	n		10	
Lead Centre to Centre	D1		1.525	
	D2		1.00	
	E1		1.525	
	E2		0.5	

11.2 Landing pattern

ScioSense suggests a footprint (land pattern) on the PCB as depicted in Figure 33 (which is identical to the pad pattern in the outline drawing Figure 32). Please find the dimensions in Table 37.

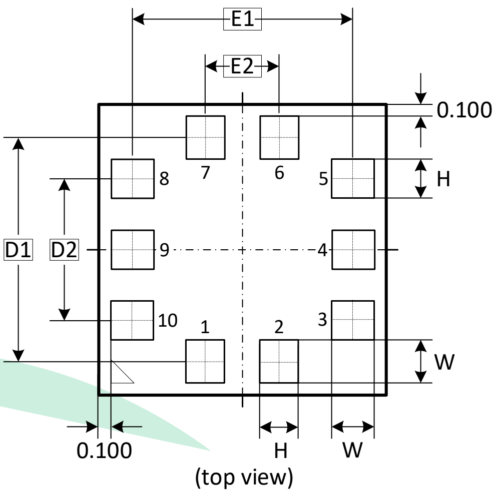


Figure 33: LGA land pattern (top view)

11.3 Device marking

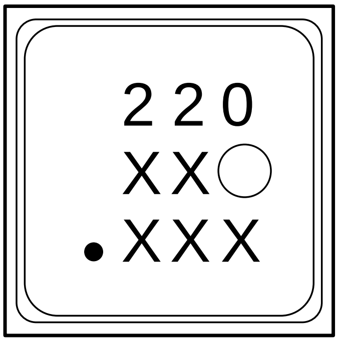


Figure 34: LGA package marking

12 Ordering information

Table 38: Ordering information

Ordering Code	Material ID	Package	Delivery Form	Delivery Quantity
ENS220S-BLGT	503700104	10-pin LGA	7" Tape & reel	3'500 pcs
ENS220-LG_EK_ST V1	503700107	PCB	Box	1 pc
ENS220-LG_DK_ST V1	503700109	Rigid-flex PCB	Box	1 pc

13 RoHS Compliance & ScioSense Green Statement

RoHS: The term RoHS compliant means that Sciosense B.V. products fully comply with current RoHS directives. Our semiconductor products do not contain any chemicals for all 6 substance categories, including the requirement that lead does not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, RoHS compliant products are suitable for use in specified lead-free processes.

ScioSense Green (RoHS compliant and no Sb/Br): ScioSense Green defines that in addition to RoHS compliance, our products are free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information: The information provided in this statement represents Sciosense B.V. knowledge and belief as of the date that it is provided. Sciosense B.V. bases its knowledge and belief on information provided by third parties and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. Sciosense B.V. has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. Sciosense B.V. and Sciosense B.V. suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

14 Copyrights & Disclaimer

15 Document status

Table 39: Document status

Document Status	Product Status	Definition
Product Preview	Pre-Development	Information in this datasheet is based on product ideas in the planning phase of development. All specifications are design goals without any warranty and are subject to change without notice.
Preliminary Datasheet	Pre-Production	Information in this datasheet is based on products in the design, validation, or qualification phase of development. The performance and parameters shown in this document are preliminary without any warranty and are subject to change without notice.
Datasheet	Production	Information in this datasheet is based on products in ramp-up to full production or full production which conform to specifications in accordance with the terms of Sciosense B.V. standard warranty as given in the General Terms of Trade.
Datasheet (Discontinued)	Discontinued	Information in this datasheet is based on products which conform to specifications in accordance with the terms of Sciosense B.V. standard warranty as given in the General Terms of Trade, but these products have been superseded and should not be used for new designs.

16 Revision information

Table 40: Revision history

Revision	Date	Comment	Pages
1.0	2023-09-19	Official release	all

Note(s) and/or Footnote(s):

1. Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
2. Correction of typographical errors is not explicitly mentioned.

ScioSense is a Joint Venture of ams AG

Address: Sciosense B.V.
High Tech Campus 10
5656 AE Eindhoven
The Netherlands

Contact: www.sciosense.com
info@sciosense.com