SCD4x

Breaking the size barrier in CO_2 sensing

Features

- Photoacoustic NDIR sensor technology PASens®
- Smallest form factor: 10.1 x 10.1 x 6.5 mm^3
- Reflow solderable for cost-effective assembly
- Digital I^2C interface
- Integrated temperature and humidity sensor

Product Variants

- SCD40: Base accuracy, specified measurement range 400 –2'000 ppm
- SCD41: High accuracy, specified measurement range 400 – 5'000 ppm, compatible with relevant IAQ standards, several power modes

Product Summary

The SCD4x is Sensirion's next generation miniature CO_2 sensor. This sensor builds on the photoacoustic NDIR sensing principle and Sensirion's patented PASens® and CMOSens® technology to offer high accuracy at an unmatched price and smallest form factor. SMD assembly allows cost- and space-effective integration of the sensor combined with maximal freedom of design. On-chip signal compensation is realized with the built-in SHT4x humidity and temperature sensor.

CO_2 is a key indicator for indoor air quality (IAQ) as high levels compromise humans' cognitive performance and well-being. The SCD4x enables smart ventilation systems to regulate ventilation in the most energy-efficient and human-friendly way. Moreover, indoor air quality monitors and other connected devices based on the SCD4x can help maintain low CO_2 concentration for a healthy, productive environment.

Product Overview

Products	Details
SCD40-D-R2	Base accuracy, specified range 400 – 2'000 ppm
SCD41-D-R2	High accuracy, specified range 400 – 5'000 ppm, low power modes supported

Functional Block Diagram

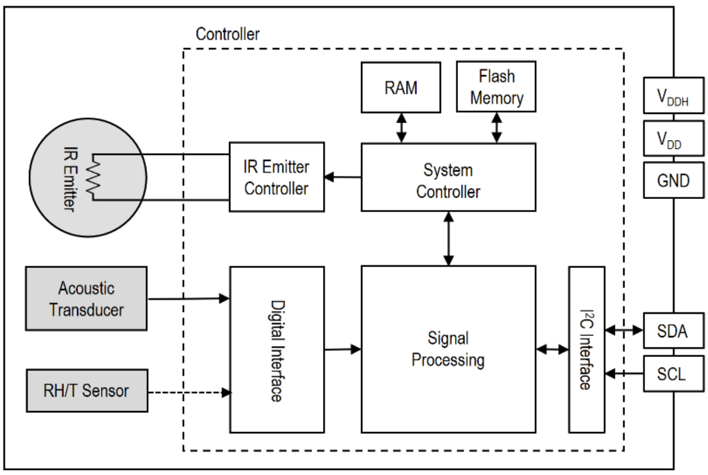

Table of Contents

1 Sensor Performance

1.1 CO_2 Sensing Performance

Default conditions of 25 °C, 50 %RH, ambient pressure 1013 mbar, periodic measurement and 3.3 V supply voltage apply to values in the table below, unless otherwise stated.

Parameter	Conditions	Value
CO_2 output range[1]	-	0 – 40’000 ppm
SCD40 CO_2 measurement accuracy[2]	400 ppm – 2’000 ppm	±(50 ppm + 5% of reading)
SCD41 CO_2 measurement accuracy[2]	400 ppm – 1’000 ppm	±(50 ppm + 2.5% of reading)
	1’001 ppm – 2’000 ppm	±(50 ppm + 3% of reading)
	2’001 ppm – 5’000 ppm	±(40 ppm + 5% of reading)
Repeatability	Typical	±10 ppm
Response time[3]	$\tau_{63\%}$, typical	60 s
Additional accuracy drift after five years with automatic self-calibration (ASC) algorithm enabled[4]	Typical, 400 – 2000 ppm	±(5 ppm + 0.5 % of reading)

Table 1: SCD40 and SCD41 CO_2 sensor specifications

1.2 Humidity Sensing Performance

Parameter	Conditions	Value
Humidity measurement range	-	0 %RH – 100 %RH
Accuracy (typ.)	15 °C – 35 °C, 20 %RH – 65 %RH	±6 %RH
	-10 °C – 60 °C, 0 %RH – 100 %RH	±9 %RH
Repeatability	Typical	±0.4 %RH
Response time[3]	$\tau_{63\%}$, typical	90 s
Accuracy drift	-	<0.25 %RH / year

Table 2: SCD4x humidity sensor specifications[5]

1.3 Temperature Sensing Performance

Parameter	Conditions	Value
Temperature measurement range	-	- 10 °C – 60 °C
Accuracy (typ.)	15 °C – 35 °C	± 0.8 °C
	-10 °C – 60 °C	± 1.5 °C
Repeatability	-	± 0.1 °C
Response time[3]	$\tau_{63\%}$, typical	120 s
Accuracy drift	-	< 0.03 °C / year

Table 3: SCD4x temperature sensor specifications[5]

[1] Exposure to CO_2 concentrations smaller than 400 ppm can affect the accuracy of the sensor if the ASC is on.
[2] Deviation from a high-precision reference with gas mixtures having a ±2% tolerance. Rough handling, shipping and sensor assembly can temporarily impact the accuracy. Accuracy can be fully restored through the forced recalibration (FRC) or ASC algorithms at least 5 days after sensor assembly (See Section 3.7)
[3] Time for achieving 63% of a respective step function when operating the SCD41 Evaluation Kit in periodic measurement mode. Response time depends on design-in, signal update rate and environment of the sensor in the final application.
[4] For proper function of the ASC algorithm, the SCD4x must be exposed to air with CO_2 concentrations of 400 ppm on a weekly basis. Maximum accuracy drift after five years estimated from stress tests is ±(5 ppm + 2% of reading). Higher drift values may occur if the sensor is not handled according to its handling instructions.
[5] Design-in of the SCD4x in final application, self-heating of the sensor and the environment around the sensor impacts the accuracy of the RH/T sensor. To realize indicated specifications, the temperature-offset of the SCD4x inside the customer device must be set correctly (see Section 3.6).

2 Specifications

2.1 Electrical Specifications

Parameter	Symbol	Conditions	Min.	Typical	Max.	Units
Supply voltage DC[6]	V_{DD}		2.4	3.3 or 5.0	5.5	V
Unloaded supply voltage ripple peak to peak[7]	V_{RPP}				30	mV
Peak supply current[8]	I_{peak}	V_{DD} = 3.3 V		175	205	mA
		V_{DD} = 5 V		115	137	mA
Average supply current for periodic measurement mode, 1 measurement every 5 seconds	I_{DD}	V_{DD} = 3.3 V		15	18	mA
		V_{DD} = 5 V		11	13	mA
Average supply current for low power periodic measurement mode, 1 measurement every 30 seconds	I_{DD}	V_{DD} = 3.3 V		3.2	3.5	mA
		V_{DD} = 5 V		2.8	3	mA
Average supply current for single shot mode, 1 measurement every 5 minutes (SCD41 only)[9]	I_{DD}	V_{DD} = 3.3 V		0.45	0.5	mA
		V_{DD} = 5 V		0.36	0.4	mA
Input high level voltage	V_{IH}		0.65 x V_{DD}		1 x V_{DD}	-
Input low level voltage	V_{IL}				0.3 x V_{DD}	-
Output low level voltage	V_{OL}	3 mA sink current			0.66	V

Table 4: SCD4x electrical specifications

2.2 Absolute Maximum Ratings

Stress levels beyond those listed in **Table 5** may cause permanent damage to the device. Exposure to minimum/maximum rating conditions for extended periods may affect sensor performance and device reliability.

Parameter	Conditions	Value
Temperature operating conditions		-10 – 60 °C
Humidity operating conditions[10]	Non-condensing	0 – 95 %RH
MSL Level		3
DC supply voltage		-0.3 V – 6.0 V
Max voltage on pins SDA, SCL, GND		-0.3 V – V_{DD} + 0.3 V
Input current on pins SDA, SCL, GND		-280 mA – 100 mA
Short term storage temperature[11]		-40 °C – 70 °C
Recommended storage temperature		10 °C – 50 °C
ESD HBM (pads and metal cap)		2 kV
ESD CDM		500 V
Maintenance Interval	Maintenance free when the ASC algorithm[12] is used.	None
Sensor lifetime[13]	Typical operating conditions	>10 years

Table 5: SCD4x operation conditions, lifetime and maximum ratings

[6] Supply voltage must be kept constant for stable sensor operation.
[7] Valid only for the supply voltage without the load of the sensor.
[8] Refers to sustained current.
[9] On-demand measurement with freely adjustable interval. See Section 3.10.
[10] Accuracy can be reduced at relative humidity levels lower than 10%.
[11] Short term storage refers to temporary conditions during e.g., transport.
[12] For proper function of the ASC field-calibration algorithm, the SCD4x must be exposed to air with CO_2 concentrations of 400 ppm on a weekly basis.
[13] Sensor tested over simulated lifetime of >10 years for indoor environment mission profile.

2.3 Interface Specifications

The SCD4x comes in an LGA package (**Table 6**). The package outline is schematically displayed in Section 4.1. The landing pattern of the SCD4x can be found in Section 4.2.

Name	Comments
VDD	Supply voltage
VDDH	Supply voltage IR source, must be connected to VDD on customer PCB
GND	Ground contact
SDA	I^{2}C Serial data, bidirectional
SCL	I^{2}C Serial clock
DNC	Do not connect, pads must be soldered to a floating pad on the customer PCB

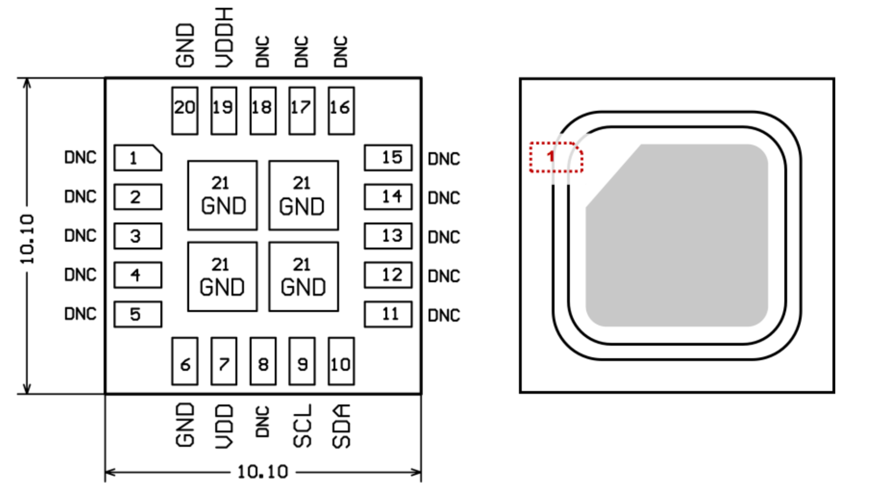

Table 6: Pin assignment (top view). The notched corner of the protection membrane serves as a polarity mark to indicate pin 1 location.

VDD and VDDH are used to supply the sensor and must always be kept at the same voltage, i.e. both should be connected to the same power supply. The combined maximum current drawn on VDD and VDDH is indicated in **Table 4**. VDD and VDDH must be connected to each other close to the sensor on the customer PCB.

For the sensor operation, a low noise power supply, such as a low-dropout regulator (LDO), should be chosen which can handle the peak supply current and voltage ripple peak to peak as specified in **Table 4**. Due to the sensor's internal regulation, higher transient currents (on the order of microseconds) may be observed. These transient currents can be neglected in typical design-in scenarios due to the parasitic R/L/C of the leads as well as the load regulation characteristics of the supply. Additionally, to avoid interference with the sensor regulation, the non-loaded supply voltage must not vary by more than 30 mV (e.g. ripples or drops caused by other loads). Operating the sensor with a separate LDO is recommended.

SCL is used to synchronize the I^{2}C communication between the master (microcontroller) and the slave (sensor). The SDA pin is used to transfer data to and from the sensor. For safe communication, the timing specifications defined in the I^{2}C manual[14] must be met. Both SCL and SDA lines should be connected to external pull-up resistors (e.g. $R_p = 10$ kΩ, see **Figure 1**). To avoid signal contention, the microcontroller must only drive SDA and SCL low. For dimensioning resistor sizes please take bus capacity and communication frequency into account (see example in Section 7.1 of NXPs I^{2}C Manual for more details[14]). It should be noted that pull-up resistors may be included in the I/O circuits of microcontrollers.

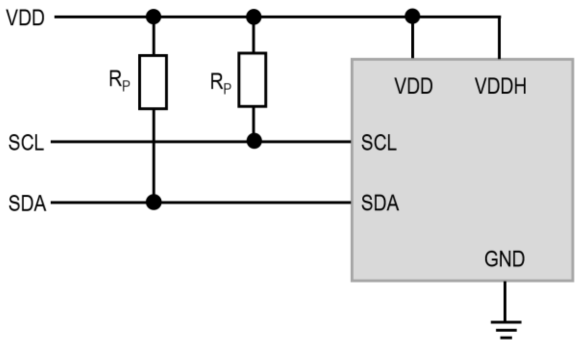

Figure 1: Typical application circuit (representative and not to scale).

[14] NXP I2C-bus specification and user manual UM10204, Rev.6, 4 April 2014

2.4 Timing Specifications

Table 7 lists the timings of the ASIC[15].

Parameter	Condition	Min.	Max.	Unit
Power-up time	After hard reset, $V_{DD} \geq 2.25$ V	-	30	ms
Soft reset time	After re-initialization (i.e. reinit)	-	30	ms
SCL clock frequency	-	0	400	kHz

Table 7: System timing specifications.

2.5 Material Contents

The device is fully REACH and RoHS compliant.

15 Timing specifications based on the NXP I2C-bus specification and user manual UM10204, Rev.6, 4 April 2014

3 Digital Interface Description

3.1 Power-Up and Communication Start

The sensor starts powering-up after reaching the power-up threshold voltage $V_{DD,min}$ and will take up to the maximum of the power-up time to enter the idle state. Once the idle state has been reached, it is ready to receive commands from the master. Each transmission sequence begins with a START condition (S) and ends with a STOP condition (P) as described in the I²C-bus specification.

3.2 Data Type & Length

Data sent to and received from the sensor consists of a sequence of 16-bit commands and/or 16-bit words (each to be interpreted as unsigned integer with the most significant byte transmitted first). Each data word is immediately succeeded by an 8-bit CRC. In write direction it is mandatory to transmit the checksum. In read direction it is up to the master to decide if it wants to process the checksum (see Section 3.11).

3.3 Command Sequence Types

All SCD4x commands and data are mapped to a 16-bit address space.

SCD4x	Hex. Code
I²C address	0x62

Table 8: I²C device address

The SCD4x features four different I²C command sequence types: "*read I²C sequences*", "*write I²C sequences*", "*send I²C command"* and *"send command and fetch result"* sequences. **Figure 2** illustrates how the I²C communication for the different sequence types is built-up.

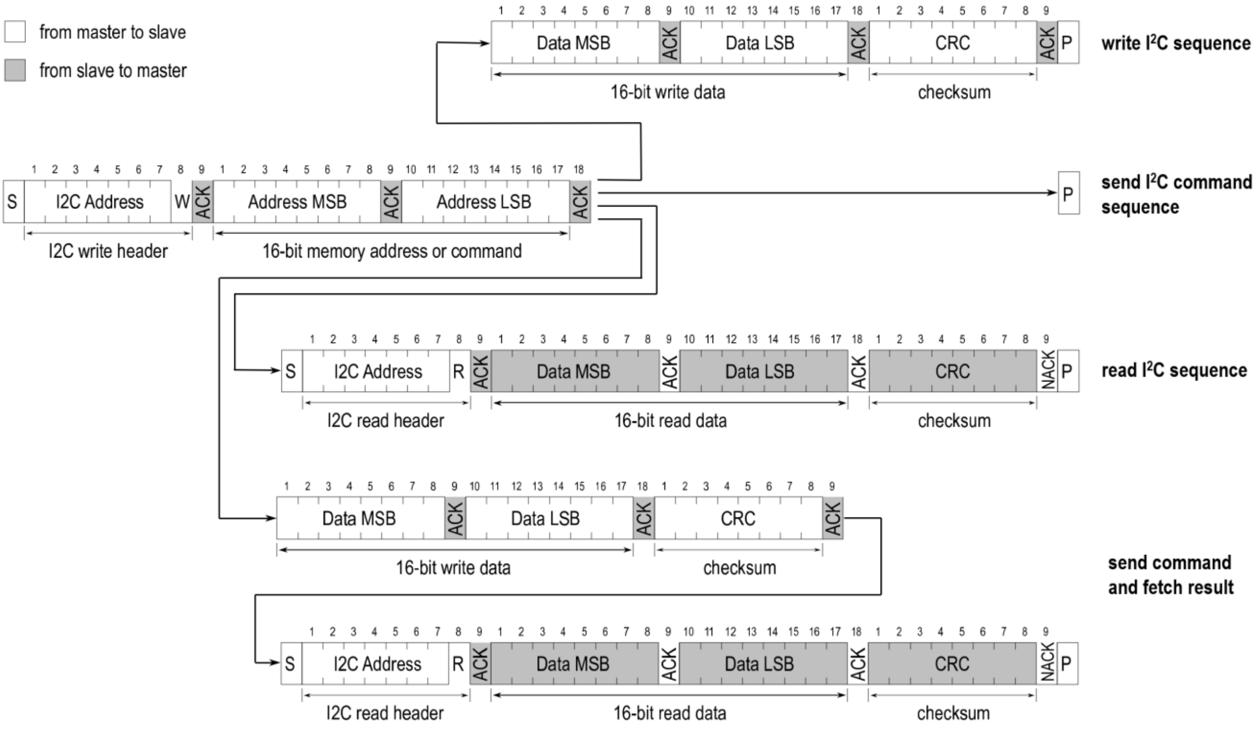

Figure 2: Command sequence types: *"write"*, *"send* command", *"read"*, and *"send command and fetch result"*

For the *"read"*" or *"send command and fetch results"* sequences, after writing the address and/or data to the sensor and sending the ACK bit, the sensor needs the *execution time* (see **Table 9**) to respond to the I²C read header with an ACK bit. Hence, it is required to wait the command *execution time* before issuing the read header. Commands must not be sent while a preceding command is being processed.

3.4 SCD4x Command Overview

An overview of the available SCD4x commands can be found in **Table 9.** A detailed description for each command can be found in the following sections.

Domain	Command	Hex. Code	I²C sequence type (see Section 3.3)	Execution time [ms]	Execution During meas.
Basic Commands Section 3.5	start_periodic_measurement	0x21b1	send command	-	no
	read_measurement	0xec05	read	1	yes
	stop_periodic_measurement	0x3f86	send command	500	yes
On-chip output signal compensation Section 3.6	set_temperature_offset	0x241d	write	1	no
	get_temperature_offset	0x2318	read	1	no
	set_sensor_altitude	0x2427	write	1	no
	get_sensor_altitude	0x2322	read	1	no
	set_ambient_pressure	0xe000	write	1	yes
	get_ambient_pressure	0xe000	read	1	yes
Field calibration Section 3.7	perform_forced_recalibration	0x362f	send command and fetch result	400	no
	set_automatic_self_calibration_enabled	0x2416	write	1	no
	get_automatic_self_calibration_enabled	0x2313	read	1	no
Low power periodic measurement mode Section 3.8	start_low_power_periodic_measurement	0x21ac	send command	-	no
	get_data_ready_status	0xe4b8	read	1	yes
Advanced features Section 3.9	persist_settings	0x3615	send command	800	no
	get_serial_number	0x3682	read	1	no
	perform_self_test	0x3639	read	10000	no
	perform_factory_reset	0x3632	send command	1200	no
	reinit	0x3646	send command	30	no
Single shot measurement mode (SCD41 only) Section 3.10	measure_single_shot	0x219d	send command	5000	no
	measure_single_shot_rht_only	0x2196	send command	50	no
	power_down	0x36e0	send command	1	no
	wake_up	0x36f6	send command	30	no
	set_automatic_self_calibration_initial_period	0x2445	write	1	no
	get_automatic_self_calibration_initial_period	0x2340	read	1	no
	set_automatic_self_calibration_standard_period	0x244e	write	1	no
	get_automatic_self_calibration_standard_period	0x234b	read	1	no

Table 9: List of SCD4x sensor commands. The final column ('During meas.') indicates whether the command can be executed while a periodic measurement is running.

3.5 Basic Commands

This section lists the basic SCD4x commands that are necessary to start a periodic measurement and subsequently read out the sensor outputs.

The typical communication sequence between the I²C master (e.g., a microcontroller) and the SCD4x sensor is as follows:

1. The sensor is powered up
2. The I²C master sends a *start_periodic_measurement* command. The signal update interval is 5 seconds.
3. The I²C master periodically reads out data with the *read_measurement* command.
4. To put the sensor back to idle mode, the I²C master sends a *stop_periodic_measurement* command.

While a periodic measurement is running, no other commands must be issued with the exception of *read_measurement*, *get_data_ready_status, stop_periodic_measurement, set_ambient_pressure* and *get_ambient_pressure.*

3.5.1 start_periodic_measurement

Description: start periodic measurement mode. The signal update interval is 5 seconds.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x21b1	-	-	-	-	not applicable
Example: start periodic measurement					
Write *(hexadecimal)*	0x21b1 *Command*				

Table 10: start_periodic_measurement I2C sequence description

3.5.2 read_measurement

Description: read sensor output. The measurement data can only be read out once per signal update interval as the buffer is emptied upon read-out. If no data is available in the buffer, the sensor returns a NACK. To avoid a NACK response, the *get_data_ready_status* can be issued to check data status (see Section 3.8.2 for further details). The I²C master can abort the read transfer with a NACK followed by a STOP condition after any data byte if the user is not interested in subsequent data.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** CO_2, Temperature, Relative Humidity		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0xec05	-	-	3	CO_2 [ppm] = word[0]	1
			3	$T = -45 + 175 * \frac{word[1]}{2^{16} - 1}$	
			3	$RH = 100 * \frac{word[2]}{2^{16} - 1}$	

Example: read sensor output (500 ppm, 25 °C, 37 %RH)

Write *(hexadecimal)*	0xec05 *Command*					
Wait	1 ms	*command execution time*				
Response *(hexadecimal)*	0x01f4 *CO_2 = 500 ppm*	0x7b *CRC of 0x01f4*	0x6667 *Temp. = 25 °C*	0xa2 *CRC of 0x6667*	0x5eb9 *RH = 37%*	0x3c *CRC of 0x5eb9*

Table 11: read_measurement I²C sequence description

3.5.3 stop_periodic_measurement

Description: stop periodic measurement mode to change the sensor configuration or to save power. Note that the sensor will only respond to other commands 500 ms after the *stop_periodic_measurement* command has been issued.

Write (hexadecimal)	**Input parameter: -**		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x3f86	-	-	-	-	500
Example: stop periodic measurement					
Write (hexadecimal)	0x3f86 Command				

Table 12: stop_periodic_measurement I²C sequence description¨

3.6 On-Chip Output Signal Compensation

The SCD4x features on-chip signal compensation to counteract pressure and temperature effects. Feeding the SCD4x with the pressure or altitude enables highest accuracy of the CO_2 output signal across a large pressure range. Setting the temperature offset improves the accuracy of the relative humidity and temperature output signal. Note that the temperature offset does not impact the accuracy of the CO_2 output.

To change or read sensor settings, the SCD4x must be in idle mode. A typical sequence between the I²C master and the SCD4x is described as follows:

1. If the sensor is operated in a periodic measurement mode, the I²C master sends a *stop_periodic_measurement* command.
2. The I²C master sends one or several commands to get or set the sensor settings.
3. If settings need to be preserved after power-cycle events, the *persist_settings* command must be sent (see Section 3.9.1)
4. The I²C master sends a *start_periodic_measurement* command to set the sensor in the operating mode again.

3.6.1 set_temperature_offset

Description: Setting the temperature offset of the SCD4x inside the customer device allows the user to optimize the RH and T output signal. Note that the temperature offset can depend on several factors such as the SCD4x measurement mode, self-heating of close components, the ambient temperature and air flow. Thus, the SCD4x temperature offset should be determined inside the customer device under its typical operation conditions (including the operation mode to be used in the application) and in thermal equilibrium. By default, the temperature offset is set to 4 °C. To save the setting to the EEPROM, the *persist_settings* (see Section 3.9.1) command must be issued. Equation (1) shows how the characteristic temperature offset can be obtained. Recommended temperature offset values are between 0 °C and 20 °C.

$$T_{offset\_actual} = T_{SCD4x} - T_{Reference} + T_{offset\_previous} \tag{1}$$

Write (hexadecimal)	**Input parameter:** Offset temperature		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x241d	3	$word[0] = T_{offset}[°C] * \frac{2^{16}-1}{175}$	-	-	1
Example: set temperature offset to 5.4 °C					
Write (hexadecimal)	0x241d Command	0x07e6 T_{offset} = 5.4 °C	0x48 CRC of 0x7e6		

Table 13: set_temperature_offset I²C sequence description

3.6.2 get_temperature_offset

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** Offset temperature		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2318	-	-	3	$T_{offset}[°C] = word[0] * \frac{175}{2^{16}-1}$	1
Example: temperature offset is 6.2 °C					
Write *(hexadecimal)*	0x2318 *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x0912 *T_{offset} = 6.2 °C*	0x63 *CRC of 0x0912*			

Table 14: get_temperature_offset I²C sequence description

3.6.3 set_sensor_altitude

Description: Reading and writing the sensor altitude must be done while the SCD4x is in idle mode. Typically, the sensor altitude is set once after device installation. To save the setting to the EEPROM, the *persist_settings* (see Section 3.9.1) command must be issued. The default sensor altitude value is set to 0 meters above sea level. Valid input values are between 0 – 3000 m.

Write (hexadecimal)	**Input parameter:** Sensor altitude		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2427	3	word[0] = Sensor altitude [m]	-	-	1
Example: set sensor altitude to 1'950 m					
Write *(hexadecimal)*	0x2427 *Command*	0x079e *Sensor altitude = 1'950 m*	0x09 *CRC of 0x079e*		

Table 15: set_sensor_altitude I²C sequence description

3.6.4 get_sensor_altitude

Description: The *get_sensor_altitude* command can be sent while the SCD4x is in idle mode to read out the previously saved sensor altitude value set by the *set_sensor_altitude* command.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** Sensor altitude		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2322	-	-	3	Sensor altitude [m] = word[0]	1
Example: sensor altitude is 1'100 m					
Write *(hexadecimal)*	0x2322 *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x044c *Sensor altitude = 1'100 m*	0x42 *CRC of 0x044c*			

Table 16: get_sensor_altitude I²C sequence description

3.6.5 set_ambient_pressure

Description: The *set_ambient_pressure* command can be sent during periodic measurements to enable continuous pressure compensation. Note that setting an ambient pressure overrides any pressure compensation based on a previously set sensor altitude. Use of this command is highly recommended for applications experiencing significant ambient pressure changes to ensure sensor accuracy. Valid input values are between 70'000 – 120'000 Pa. The default value is 101'300 Pa.

Write (hexadecimal)	**Input parameter:** Ambient pressure		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0xe000	3	word[0] = ambient P [Pa] / 100	-	-	1
Example: set ambient pressure to 98'700 Pa					
Write *(hexadecimal)*	0xe000 *Command*	0x03db *Ambient P = 98'700 Pa*	0x42 *CRC of 0x03db*		

Table 17: set_ambient_pressure I²C sequence description

3.6.6 get_ambient_pressure

Description: The *get_ambient_pressure* command can be sent during periodic measurements to read out the previously saved ambient pressure value set by the *set_ambient_pressure* command.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** Ambient pressure		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0xe000	-	-	3	ambient P [Pa] = word[0] * 100	1
Example: ambient pressure is 98'700 Pa					
Write *(hexadecimal)*	0xe000 *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x03db *Ambient P = 98'700 Pa*	0xb6 *CRC of 0x03db*			

Table 18: get_ambient_pressure I²C sequence description

3.7 Field Calibration

To realize high initial and long-term accuracy, the SCD4x includes two field calibration features. Forced recalibration (FRC) immediately restores high accuracy with the assistance of an external CO_2 reference value. Typically, an FRC is applied to compensate for drifts (e.g. the sensor assembly process). Automatic self-calibration (ASC) ensures the highest long-term stability of the SCD4x without the need of manual action steps from the user. The ASC algorithm assumes that the sensor is exposed to air with CO_2 concentrations of 400 ppm at least once per week.

3.7.1 perform_forced_recalibration

Description: To successfully conduct an accurate FRC, the following steps need to be carried out:

1. Operate the SCD4x in the operation mode later used in normal sensor operation (e.g. periodic measurement) for at least 3 minutes in an environment with a homogenous and constant CO_2 concentration. The sensor must be operated at the voltage desired for the application when performing the FRC sequence.
2. Issue the *stop_periodic_measurement* command. Wait 500 ms for the command to complete.
3. Issue the *perform_forced_recalibration* command and optionally read out the FRC correction (i.e. the magnitude of the correction) after waiting for 400 ms for the command to complete. A return value of 0xffff indicates that the FRC has failed.

Note that the sensor will fail to perform a FRC if it was not operated before sending the command.

Write (hexadecimal)	**Input parameter:** Target CO_2 concentration		**Response parameter:** FRC-correction		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x362f	3	word[0] = Target concentration [ppm CO_2]	3	FRC correction [ppm CO_2] = word[0] – 0x8000 word[0] = 0xffff in case of failed FRC	400
Example: perform forced recalibration, reference CO_2 concentration is 480 ppm					
Write *(hexadecimal)*	0x362f *Command*	0x01e0 *Input: 480 ppm*	0xb4 *CRC of 0x01e0*		
Wait	400 ms	*command execution time*			
Response *(hexadecimal)*	0x7fce *Response: - 50 ppm*	0x7b *CRC of 0x7fce*			

Table 19: perform_forced_recalibration I²C sequence description

3.7.2 set_automatic_self_calibration_enabled

Description: Set the current state (enabled / disabled) of the ASC. By default, ASC is enabled. To save the setting to the EEPROM, the *persist_settings* (see Section 3.9.1) command must be issued.

Write (hexadecimal)	**Input parameter:** ASC enabled		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2416	3	word[0] = 1 → ASC enabled word[0] = 0 → ASC disabled	-	-	1
Example: set ASC status: enabled					
Write *(hexadecimal)*	0x2416 *Command*	0x0001 *ASC enabled*	0xb0 *CRC of 0x0001*		

Table 20: set_automatic_self_calibration_enabled I²C sequence description.

3.7.3 get_automatic_self_calibration_enabled

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** ASC enabled		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2313	-	-	3	word[0] = 1 → ASC enabled word[0] = 0 → ASC disabled	1
Example: read ASC status: disabled					
Write *(hexadecimal)*	0x2313 *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x0000 *ASC disabled*	0x81 *CRC of 0x0000*			

Table 21: get_automatic_self_calibration_enabled I²C sequence description

3.8 Low Power Periodic Measurement Mode

To enable use-cases with a constrained power-budget, the SCD4x features a low power periodic measurement mode with a signal update interval of approximately 30 seconds.
The low power periodic measurement mode is initiated and read-out in a similar manner as the periodic measurement mode. Please consult Section 3.5.2 for further instructions. To avoid receiving a NACK in case the result of a subsequent measurement is not ready yet, use the *get_data_ready_status* command to check whether new measurement data is available for read-out.

3.8.1 start_low_power_periodic_measurement

Description: Start the low power periodic measurement mode. The signal update interval is approximately 30 seconds.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x21ac	-	-	-	-	not applicable
Example: start low power periodic measurement					
Write *(hexadecimal)*	0x21ac *Command*				

Table 22: start_low_power_periodic_measurement I²C sequence description

3.8.2 get_data_ready_status

Description: Poll the sensor for whether data from a periodic or single shot measurement is ready to be read out.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** data ready status		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0xe4b8	-	-	3	If the least significant 11 bits of word[0] are: 0 → data not ready else → data ready for read-out	1
Example: read data ready status: data not ready					
Write *(hexadecimal)*	0xe4b8 *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x8000 *Least significant 11 bits are 0 → data not ready*		0xa2 *CRC of 0x8000*		

Table 23: get_data_ready_status I²C sequence description

3.9 Advanced Features

3.9.1 persist_settings

Description: Configuration settings such as the temperature offset, sensor altitude and the ASC enabled/disabled parameters are by default stored in the volatile memory (RAM) only and will be lost after a power-cycle. The *persist_settings* command stores the current configuration in the EEPROM of the SCD4x, ensuring the settings persist across power-cycling. To avoid unnecessary wear of the EEPROM, the *persist_settings* command should only be sent when persistence is required and if actual changes to the configuration have been made. The EEPROM is guaranteed to withstand at least 2000 write cycles. Note that field calibration history (i.e. FRC and ASC, see Section 3.7) is automatically stored in a separate EEPROM dimensioned for the specified sensor lifetime.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x3615	-	-	-	-	800
Example: persist settings					
Write *(hexadecimal)*	0x3615 *Command*				

Table 24: persist_settings I²C sequence description

3.9.2 get_serial_number

Description: Reading out the serial number can be used to identify the chip and to verify the presence of the sensor. The *get_serial_number* command returns 3 words, and every word is followed by an 8-bit CRC checksum. Together, the 3 words constitute a unique serial number with a length of 48 bits (big endian format).

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** serial number		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x3682	-	-	9	Serial number = word[0] << 32 | word[1] << 16 | word[2]	1

Example: serial number is 273'325'796'834'238						
Write *(hexadecimal)*	0x3682 *Command*					
Wait	1 ms	*command execution time*				
Response *(hexadecimal)*	0xf896 *word[0]*	0x31 *CRC of 0xf896*	0x9f07 *word[1]*	0xc2 *CRC of 0x9f07*	0x3bbe *word[2]*	0x89 *CRC of 0x3bbe*

Table 25: get_serial_number I²C sequence description

3.9.3 perform_self_test

Description: The *perform_self_test* command can be used as an end-of-line test to check the sensor functionality.

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** sensor status		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x3639	-	-	3	word[0] = 0 → no malfunction detected word[0] ≠ 0 → malfunction detected	10000

Example: perform self-test, no malfunction detected		
Write *(hexadecimal)*	0x3639 *Command*	
Wait	10000 ms	*command execution time*
Response *(hexadecimal)*	0x0000 *No malfunction detected*	0x81 *CRC of 0x0000*

Table 26: perform_self_test I²C sequence description

3.9.4 perfom_factory_reset

Description: The *perform_factory_reset* command resets all configuration settings stored in the EEPROM and erases the FRC and ASC algorithm history.

Write (hexadecimal)	**Input parameter: -**		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x3632	-	-	-	-	1200
Example: perform factory reset					
Write *(hexadecimal)*	0x3632 *Command*				

Table 27: perform_factory_reset I²C sequence description

3.9.5 reinit

Description: The *reinit* command reinitializes the sensor by reloading user settings from EEPROM. Before sending the *reinit* command, the *stop_periodic_measurement* command must be issued. If the *reinit* command does not trigger the desired re-initialization, a power-cycle should be applied to the SCD4x.

Write (hexadecimal)	**Input parameter: -**		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x3646	-	-	-	-	30
Example: reinit					
Write *(hexadecimal)*	0x3646 *Command re-initialization*				

Table 28: reinit I²C sequence description

3.10 Single Shot Measurement Mode (SCD41 Only)

The SCD41 additionally features a single shot measurement mode for on-demand measurements.
The typical communication sequence is as follows:

1. The sensor is powered up with the *wake_up* command.
2. The I²C master sends a *measure_single_shot* command and waits for the indicated *max. command duration* time.
3. The I²C master reads out data with the *read_measurement* command (3.5.2) within the specified *max. command duration* time.
4. Repeat steps 2–3 as required by the application.
5. If desired, power down the sensor with with the *power_down* command.

To reduce noise levels, the I²C master can perform several single shot measurements in a row and average the CO_2 output values. After a power cycle, the initial single shot reading should be discarded to maximize accuracy.
The ASC is enabled per default in single shot operation and optimized for single shot measurements performed every 5 minutes. Longer or shorter measurement intervals will result in less or more frequent ASC corrections. To optimize the ASC for measurement intervals other than 5 minutes, the ASC initial and standard intervals can be reconfigured (see relevant commands in following subsections and relevant supporting documentation[16]).

For extreme low-power applications, the sensor may be power cycled between measurements either by cutting/re-applying the supply voltage or by using the *power_down/wake_up* commands. Note that for power-cycled single shot operation, ASC is not available in either case.

[16] More information on ASC settings and SCD4x low power modes can be found in the application note on "Low Power Operation SCD4x"

3.10.1 measure_single_shot

Description: On-demand measurement of CO_2 concentration, relative humidity and temperature. The sensor output is read out by using the *read_measurement* command (Section 3.5.2).

Write (hexadecimal)	**Input parameter: -**		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x219d	-	-	-	-	5000
Example: measure single shot					
Write *(hexadecimal)*	0x219d *Command*				

Table 29: measure_single_shot I²C sequence description

3.10.2 measure_single_shot_rht_only

Description: On-demand measurement of relative humidity and temperature only. The sensor output is read out by using the *read_measurement* command (Section 3.5.2). CO_2 output is returned as 0 ppm.

Write (hexadecimal)	**Input parameter: -**		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2196	-	-	-	-	50
Example: measure single shot, RH and T output only					
Write *(hexadecimal)*	0x2196 *Command*				

Table 30: measure_single_shot_rht_only I²C sequence description

3.10.3 power_down

Description: Put the sensor from idle to sleep to reduce current consumption. Can be used to power down when operating the sensor in power-cycled single shot mode.

Write (hexadecimal)	**Input parameter: -**		**Response parameter: -**		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x36e0	-	-	-	-	1
Example: power down the sensor					
Write *(hexadecimal)*	0x36e0 *Command*				

Table 31: power_down I²C sequence description

3.10.4 wake_up

Description: Wake up the sensor from sleep mode into idle mode. Note that the SCD4x does not acknowledge the *wake_up* command. The sensor idle state after wake up can be verified by reading out the serial number (Section 3.9.2).

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x36f6	-	-	-	-	30
Example: wake up the sensor					
Write *(hexadecimal)*	0x36f6 *Command*				

Table 32: wake_up I²C sequence description

3.10.5 set_automatic_self_calibration_initial_period

Description: Set the initial period for ASC correction (in hours). By default, the initial period for ASC correction is 44 hours. Allowed values are integer multiples of 4 hours. Note: Assumes an average measurement interval of 5 minutes in single shot operation. For different average single shot measurement intervals, the parameter value should be scaled inversely proportional to this (e.g. by factor 0.5 for 10 minutes average single shot interval). Note: a value of 0 results in an immediate correction. To save the setting to the EEPROM, the *persist_settings* (see Section 3.9.1) command must be issued.

Write (hexadecimal)	**Input parameter:** ASC initial period		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2445	3	word[0] = ASC initial period [hours]	-	-	1
Example: write ASC initial period of 76 hours					
Write *(hexadecimal)*	0x2445 *Command*	0x004c *Initial period 76 hours*	0xc1 *CRC of 0x004c*		

Table 33: set_automatic_self_calibration_initial_period I²C sequence description

3.10.6 get_automatic_self_calibration_initial_period

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** ASC initial period		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x2340	-	-	3	ASC initial period [hrs] = word[0]	1
Example: read ASC initial period of 76 hours					
Write *(hexadecimal)*	0x2340 *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x004c *76 hours*	0xc1 *CRC of 0x004c*			

Table 34: get_automatic_self_calibration_initial_period I²C sequence description

3.10.7 set_automatic_self_calibration_standard_period

Description: Set the standard period for ASC correction (in hours). By default, the standard period for ASC correction is 156 hours. Allowed values are integer multiples of 4 hours. Note: Assumes an average measurement interval of 5 minutes in single shot operation. For different average single shot measurement intervals, the parameter value should be scaled inversely proportional to this (e.g. by factor 0.5 for a 10 minutes average single shot interval). To save the setting to the EEPROM, the *persist_settings* (see Section 3.9.1) command must be issued.

Write (hexadecimal)	**Input parameter:** ASC standard period		**Response parameter:** -		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x244e	3	word[0] = ASC standard period [hrs]	-	-	1
Example: set automatic self-calibration standard period of 156 hours					
Write *(hexadecimal)*	0x244e *Command*	0x009c *Standard period 156 hours*	0xc5 *CRC of 0x009c*		

Table 35: set_automatic_self_calibration_standard_period I²C sequence description

3.10.8 get_automatic_self_calibration_standard_period

Write (hexadecimal)	**Input parameter:** -		**Response parameter:** ASC standard period		Max. command duration [ms]
	length [bytes]	signal conversion	length [bytes]	signal conversion	
0x234b	-	-	3	word[0] = ASC standard period [hrs]	1
Example: read ASC standard period of 156 hours					
Write *(hexadecimal)*	0x234b *Command*				
Wait	1 ms	*command execution time*			
Response *(hexadecimal)*	0x009c *Standard period 156 hours*	0xc5 *CRC of 0x009c*			

Table 36: get_automatic_self_calibration_standard_period I²C sequence description

3.11 Checksum Calculation

The 8-bit CRC checksum transmitted after each data word is generated by a CRC algorithm. Its properties are displayed in **Table 37**. The CRC covers the contents of the two previously transmitted data bytes. To calculate the checksum only these two previously transmitted data bytes are used. Note that command words are not followed by CRC.

Property	Value	Example code (C/C++)
Name	CRC-8	(see code below)
Width	8 bit	
Protected Data	read and/or write data	
Polynomial	0x31 ($x^8 + x^5 + x^4 + 1$)	
Initialization	0xff	
Reflect input	False	
Reflect output	False	
Final XOR	0x00	
Examples	CRC (0xbeef) = 0x92	

```c
#define CRC8_POLYNOMIAL 0x31
#define CRC8_INIT 0xff

uint8_t sensirion_common_generate_crc(const uint8_t* data, uint16_t count) {
    uint16_t current_byte;
    uint8_t crc = CRC8_INIT;
    uint8_t crc_bit;
    /* calculates 8-Bit checksum with given polynomial */
    for (current_byte = 0; current_byte < count; ++current_byte) {
        crc ^= (data[current_byte]);
        for (crc_bit = 8; crc_bit > 0; --crc_bit) {
            if (crc & 0x80)
                crc = (crc << 1) ^ CRC8_POLYNOMIAL;
            else
                crc = (crc << 1);
        }
    }
    return crc;
}
```

Table 37: I²C CRC properties

4 Mechanical specifications

4.1 Package Outline

Figure 3 schematically displays the package outline, as well as key nominal dimensions and their tolerances in millimeters. The notched corner of the protection membrane serves as a polarity mark to indicate the location of pin 1. Note that the white protection membrane on top of the sensor must not be removed or tampered with to ensure proper sensor operation. The weight of the sensor is approx. 0.6 g.

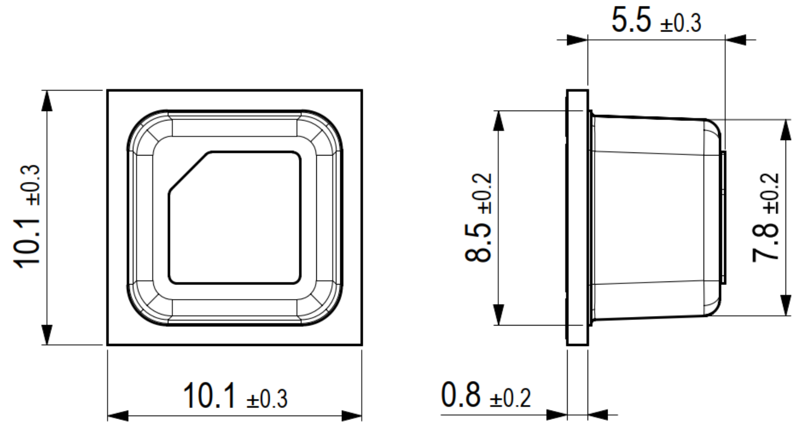

Figure 3: Packaging outline drawing of the SCD4x: (left) top view and (right) side view. Dimensions are given in millimeters.

4.2 Land Pattern

Recommended land pattern, solder paste and solder mask are shown in **Figure 4**. The exact mask geometries, distances and stencil thicknesses must be adapted to the customer soldering processes.

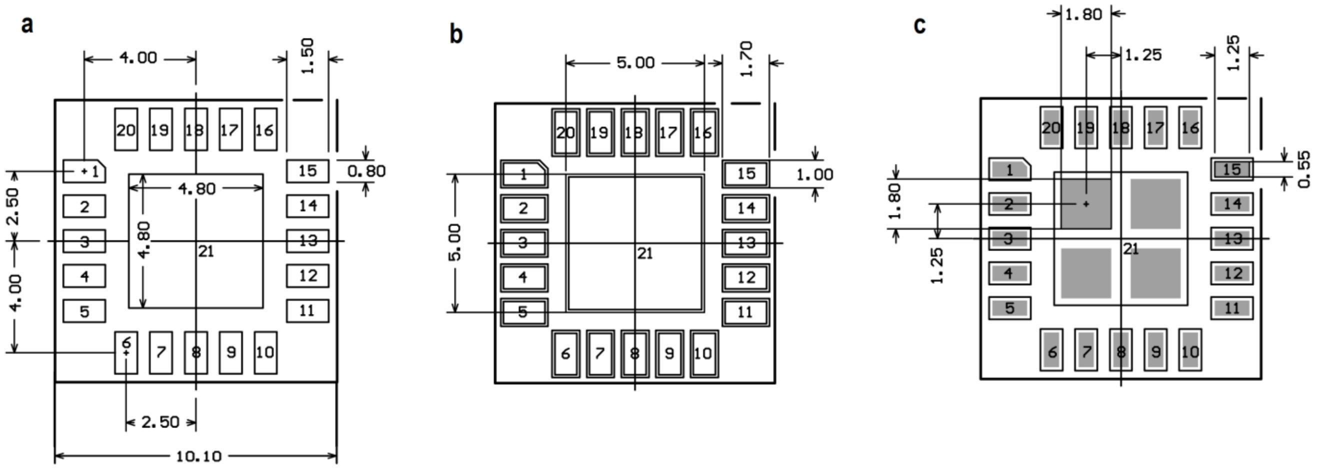

Figure 4: SCD4x footprint (top view): landing pads (**a**), solder mask (**b**) and solder paste (**c**). Dimensions given in millimeters.

4.3 Tape & Reel Package

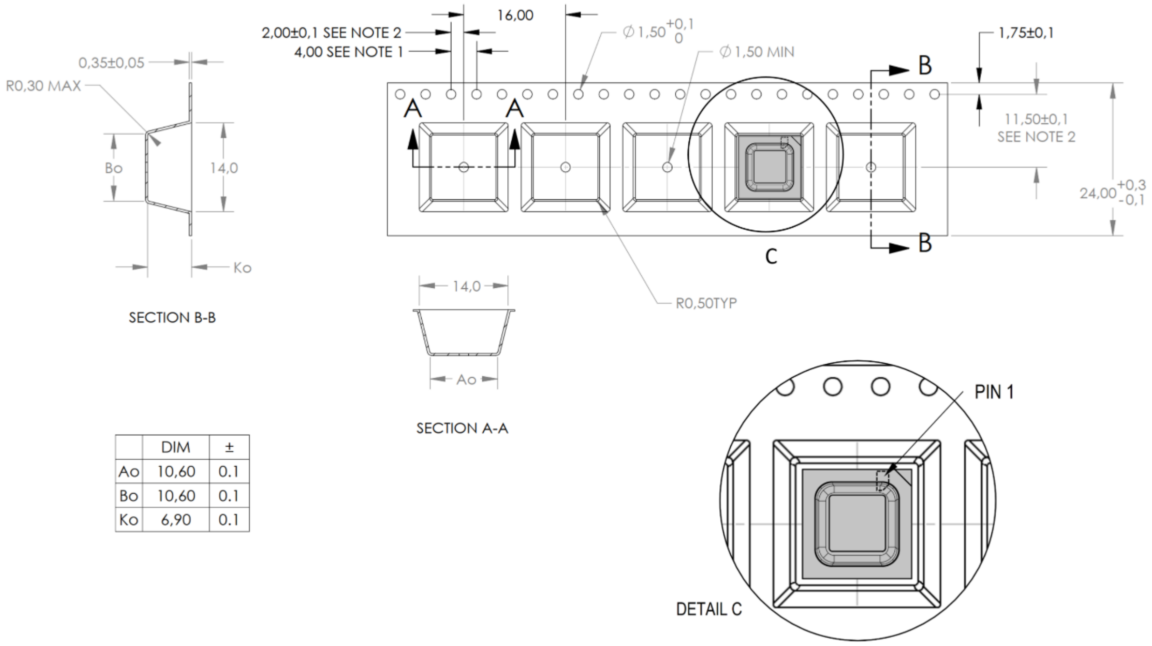

Figure 5: Technical drawing of the packaging tape with sensor orientation in tape. Header tape is to the right and trailer tape to the left on the drawing. Dimensions are given in millimeters.

4.4 Moisture Sensitivity Level

Sensirion SCD4x sensors must be treated according to Moisture Sensitivity Level 3 (MSL3) as described in IPC/JEDEC J-STD-033B1. Exposure to moisture levels or solder reflow temperatures which exceed the limits as stated in this document can result in yield loss and reliability degradation[17].
The manufacturing floor time (out of bag) at the customer's end is 168 hours at normal factory conditions (≤30 °C and 60 %RH). If sensors are not mounted within this time, or have been exposed to higher temperatures and humidity (>30 °C and >60 %RH), or there is any doubt about the airtight integrity of the dry pack, the parts should be baked (for baking parameters see **Table 38**). The maximum allowed baking temperature is 40 °C if the sensors are inside the reel.

SCD4x package type	Baking temperature	Min. baking time	Baking condition
Sensors removed from tape	90 °C	48 hours	RH < 5 %
Sensors in tape	40 °C	23 days	RH < 5 %

Table 38: Baking conditions for SCD4x if manufacturing floor time with open bag is exceeded.

[17] More information on SCD4x packing and storage can be found in the user guide "Handling Instructions SCD4x"

4.5 Soldering Instructions

The sensors are designed to withstand a soldering profile based on IPC/JEDEC J-STD-020, with a maximum peak temperature of 245°C up to 30 sec and Pb-free assembly in IR/Convection reflow ovens. See **Table 39** for more details.

Note that due to the size and shape of the SCD4x sensor, significant temperature differences across the sensor element can occur during reflow soldering. Specifically, the temperature within the sensor cap can be higher than the temperature measured at the pad using usual temperature monitoring methods. Care must be taken that a temperature of 245°C is not exceeded at any time in any part of the sensor.

The SCD4x is not compatible with vapor phase reflow soldering. The dust cover on top of the cap must not be removed or wetted with any liquid. Do not apply extra flux during the reflow soldering or reflow solder more than once. Do not apply any board wash process step subsequently to the reflow soldering[18].

Minor temporary accuracy deviations of the CO_2 reading can result from the reflow soldering of the SCD4x. Full sensor accuracy is restored after at most five days after the soldering process, independently on whether the sensor is operated or not.

Average ramp-up rate	< 3 °C / second
Liquid phase ▪ T_L ▪ t_L	> 220 °C < 60 seconds
Peak temperature ▪ T_P ▪ t_P	≤ 245°C < 30 seconds
Ramp-down rate	< 4 °C / seconds for temperature > T_L

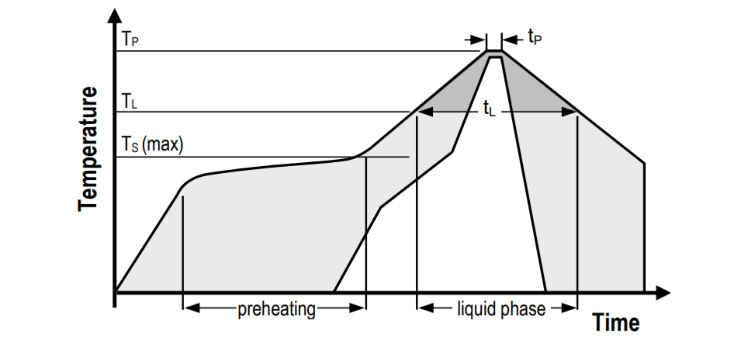

Table 39: Soldering profile parameters

4.6 Traceability and Identification

All SCD4x sensors have a distinct electronic serial number for identification and traceability (see Section 3.9.2). The serial number can be decoded by Sensirion only and allows for tracking through production, calibration, and testing.

All SCD4x sensors include a laser marking on the sidewall of the sensor cap. The laser marking contains the product variant (i.e., SCD40 or SCD41) and the product serial number within a data matrix (**Figure 6**).

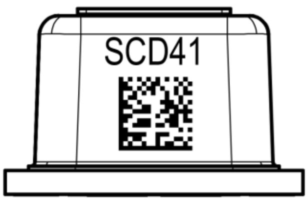

Figure 6: Technical drawing of the laser marking of product type and data matrix on the sidewall of the sensor cap.

[18] More information on SCD4x reflow soldering can be found in the user guide "Handling Instructions SCD4x"

5 Ordering Information

Use the part names and product numbers shown in **Table 40** when ordering the SCD4x CO_2 sensor. For the latest product information and local distributors, please visit the Sensirion website.

Part Name	Description	Ordering quantity (pcs)	Product Number
SCD40-D-R1	SCD40 CO2 sensor SMD component as reel, I2C	60 sensors per reel	3.000.496
SCD40-D-R2	SCD40 CO2 sensor SMD component as reel, I2C	600 sensors per reel	3.000.521
SCD41-D-R1	SCD41 CO2 sensor SMD component as reel, I2C	60 sensors per reel	3.000.960
SCD41-D-R2	SCD41 CO2 sensor SMD component as reel, I2C	600 sensors per reel	3.000.961
SEK-SCD41-Sensor	SEK-SCD41-Sensor set; SCD41 on development board with cables	1	3.000.455
SEK-SensorBridge	Sensor Bridge to connect SEK-SCD41-Sensor to computer	1	3.000.124

Table 40: Active part names and product numbers for ordering SCD4x

5.1 Historical Information

The parts / product numbers of the SCD4x product family shown in **Table 41** are obsolete.

Period Active	Product Number	Note
Before 01.08.2023	3.000.497	For applicable specifications, see Version 1.3 of the SCD4x Datasheet
Before 01.08.2023	3.000.498	For applicable specifications, see Version 1.3 of the SCD4x Datasheet

Table 41: Obsolete ordering information

6 Revision History

Date	Version	Page(s)	Changes
January 2021	1	all	Initial release
April 2021	1.1	16 - 17	Adjustment max. command time self-test (Section 3.9) and single shot (Section 3.10), minor revisions on other pages
May 2022	1.2	3 12 18 22 all	Clarification on additional sensor accuracy drift (Table 1) Clarification of set_ambient_pressure command description (Section 3.6.5) Addition of power_down and wake_up commands (Section 3.10) Addition of minor temporary accuracy deviation after reflow soldering (Section 4.5) Minor editorial revisions
September 2022	1.3	1,22 All	Correction of hyperlink Minor editorial revisions
February 2023	1.4	3 4 5 6 7 8 10 11 12 17 19 20 22 24 All	Updated SCD41 accuracy values, updated drift parameters and drift conditions (Table 1), correction of tolerance in footnote #2, clarification of footnotes #2, 4 and 5 Clarification of operation mode per average supply current (Table 4), additional information on ESD HBM (Table 5), additional footnote #8, clarification of footnotes #7 and 12 Clarification of recommendations on power supply for sensor operation (Section 2.3) Correction of power-up time and soft reset time, increase of maximum SCL clock frequency to 400 kHz (Section 2.4) Minor editorial revisions for clarification (Section 3.1) Addition of get_ambient_pressure, set_automatic_self_calibration_initial_period, get_automatic_self_calibration_initial_period, set_automatic_self_calibration_standard_period, get_automatic_self_calibration_standard_period ~~and set_automatic_self_calibration_target~~ commands (Table 9), correction of reinit and wake_up execution times (Section 3.4) Addition of recommended temperature offset range, formula correction of signal conversion (Section 3.6.1) Formula correction of signal conversion (Section 3.6.2), addition of valid sensor altitude input values (Section 3.6.3) Addition of valid ambient pressure input values to set_ambient_pressure command (Section 3.6.5) and addition of get_ambient_pressure command (Section 3.6.6) Correction of reinit max. command duration (Section 3.9.5), clarification of typical communication sequence for single shot measurement mode (Section 3.10) Correction of wake_up max. command duration (Section 3.10.4), addition of set_automatic_self_calibration_initial_period (Section 3.10.5) and get_automatic_self_calibration_initial_period commands (Section 3.10.6) Addition of set_automatic_self_calibration_standard_period command (Section 3.10.7) and get_automatic_self_calibration_standard_period command (Section 3.10.8) Additional information on white protection membrane (Section 4.1) Increase of peak reflow soldering temperature to 245°C, clarification of soldering guidance (Section 4.5), addition of information concerning product laser marking (Section 4.6) Minor editorial revisions
July 2023	1.5	4 15 19, 20 22 25 All	Clarified description of supply voltage ripple specification (Section 2, Table 4) Addition of description for get_data_ready_status command (Section 3.8.2) Clarification on ASC availability in power-cycled single shot operation (Section 3.10), Addition of clarification on ASC period parameter scaling for single shot operation (Sections 3.10.5 and 3.10.7) Moved dimensions into Figure 3, removed separate table with dimensions (Section 4.1) Updated product numbers for SCD41, addition of historical ordering information (Section 5) Minor editorial revisions

Important Notices

Warning, Personal Injury

Do not use this product as safety or emergency stop devices or in any other application where failure of the product could result in personal injury. Do not use this product for applications other than its intended and authorized use. Before installing, handling, using or servicing this product, please consult the data sheet and application notes. Failure to comply with these instructions could result in death or serious injury.

If the Buyer shall purchase or use SENSIRION products for any unintended or unauthorized application, Buyer shall defend, indemnify and hold harmless SENSIRION and its officers, employees, subsidiaries, affiliates and distributors against all claims, costs, damages and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if SENSIRION shall be allegedly negligent with respect to the design or the manufacture of the product.

ESD Precautions

The inherent design of this component causes it to be sensitive to electrostatic discharge (ESD). To prevent ESD-induced damage and/or degradation, take customary and statutory ESD precautions when handling this product.

Warranty

SENSIRION warrants solely to the original purchaser of this product for a period of 12 months (one year) from the date of delivery that this product shall be of the quality, material and workmanship defined in SENSIRION's published specifications of the product. Within such period, if proven to be defective, SENSIRION shall repair and/or replace this product, in SENSIRION's discretion, free of charge to the Buyer, provided that:

- notice in writing describing the defects shall be given to SENSIRION within fourteen (14) days after their appearance;
- such defects shall be found, to SENSIRION's reasonable satisfaction, to have arisen from SENSIRION's faulty design, material, or workmanship;
- the defective product shall be returned to SENSIRION's factory at the Buyer's expense; and
- the warranty period for any repaired or replaced product shall be limited to the unexpired portion of the original period.

This warranty does not apply to any equipment which has not been installed and used within the specifications recommended by SENSIRION for the intended and proper use of the equipment. EXCEPT FOR THE WARRANTIES EXPRESSLY SET FORTH HEREIN, SENSIRION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE PRODUCT. ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE EXPRESSLY EXCLUDED AND DECLINED.

SENSIRION is only liable for defects of this product arising under the conditions of operation provided for in the data sheet and proper use of the goods. SENSIRION explicitly disclaims all warranties, express or implied, for any period during which the goods are operated or stored not in accordance with the technical specifications.

SENSIRION does not assume any liability arising out of any application or use of any product or circuit and specifically disclaims any and all liability, including without limitation consequential or incidental damages. All operating parameters, including without limitation recommended parameters, must be validated for each customer's applications by customer's technical experts. Recommended parameters can and do vary in different applications.

SENSIRION reserves the right, without further notice, (i) to change the product specifications and/or the information in this document and (ii) to improve reliability, functions and design of this product.

Headquarters and Subsidiaries

Sensirion AG
Laubisruetistr. 50
CH-8712 Staefa ZH
Switzerland

phone: +41 44 306 40 00
fax: +41 44 306 40 30
info@sensirion.com
www.sensirion.com

Sensirion Inc., USA
phone: +1 312 690 5858
info-us@sensirion.com
www.sensirion.com

Sensirion Korea Co. Ltd.
phone: +82 31 337 7700~3
info-kr@sensirion.com
www.sensirion.com/kr

Sensirion Japan Co. Ltd.
phone: +81 3 3444 4940
info-jp@sensirion.com
www.sensirion.com/jp

Sensirion China Co. Ltd.
phone: +86 755 8252 1501
info-cn@sensirion.com
www.sensirion.com/cn

Sensirion Taiwan Co. Ltd
phone: +886 3 5506701
info@sensirion.com
www.sensirion.com

To find your local representative, please visit
www.sensirion.com/distributors