ScioSense

ENS212A

High Performance Digital Humidity and Temperature Sensor for Premium Consumer, Industrial and Automotive Applications

ENS212A datasheet

Version: 1.0

Release Date: 2025-10-10

Document Status: Production

The ENS212A is a premium automotive grade, high-performance temperature and relative humidity sensor with enhanced accuracies tailored to the needs of applications with increased dependence on reliable data over a wide range of operative conditions, such as building automation, smart home applications, HVAC systems and automotive applications such as windshield de-fogging, HVAC energy balancing or dewpoint detection.

Encapsulated in a tiny QFN4 package, the devices feature an I²C interface to communicate with an external host processor. Moreover, the ENS212A provides digital, pre-calibrated outputs (Kelvin and % relative humidity) and works well with ScioSense's gas sensor portfolio.

Key Features & Benefits

- **Ultra-accurate**
  - Temperature: ±0.15°C, typ.
  - Relative humidity: ±1.5%RH, typ.
- **Wide sensing range**
  - Temperature: -40 to 125 C
  - Relative humidity: 0 to 100%RH
- **Low power**
  - Standby current: 40 nA
  - Automatic low-power standby when not measuring
- **AEC Q100 grade 1** compliant device

Applications

- **Automotive**
  - Automotive HVAC management & general energy balancing
  - Windshield de-fogging
  - Dewpoint detection
  - Cabin temperature and humidity monitoring
- **Building automation, smart home, HVAC**
  - Indoor air quality detection
  - Demand-controlled ventilation
  - Smart thermostats
- **Home appliances**
  - Air cleaners / purifiers
  - Refrigerators, washing machines, dishwashers, dryers
- **Mobiles / Wearables**
- **IoT devices, weather stations**
- **Instrumentation**

Properties

- Small 2 x 2 x 0.75mm³ QFN4 package
- Operating voltage range: 1.71 to 3.60V
- Standard and fast I²C interface
- T&R packaged, reflow solderable

Table of Contents

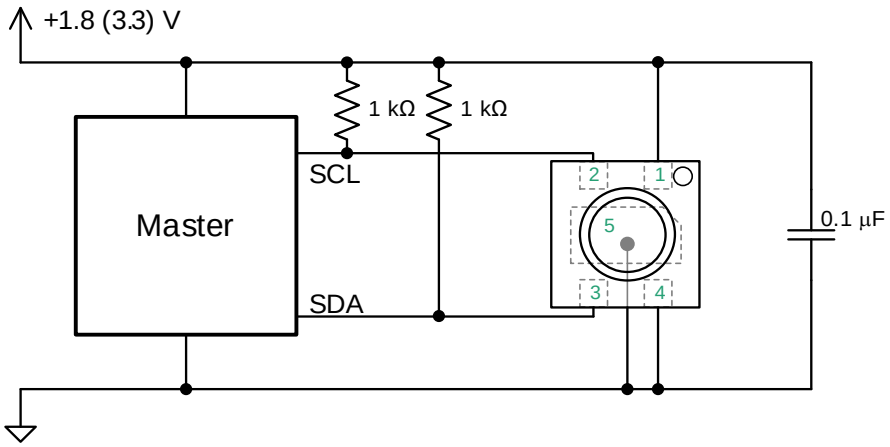
Key Features & Benefits **2**
Applications **2**
Properties **2**
Table of Contents **3**
1 Quick start **4**
2 Block diagram **5**
3 Pin assignment **5**
4 Absolute maximum ratings **6**
5 Electrical characteristics **7**
6 I²C timing characteristics **8**
7 Temperature sensor characteristics **8**
8 Relative humidity sensor characteristics **9**
9 System timing characteristics **9**
10 Functional description **10**
11 Temperature sensor **10**
12 Relative humidity sensor **12**
13 RH accuracy at various temperatures **13**
14 The I²C interface **14**
15 I²C operations on registers **14**
16 The I²C slave address **15**
17 Sensor control **16**
18 Sensor timing **17**
19 Sensor readout registers **18**
20 Computing CRC-7 ..... **19**
21 Processing T_VAL and H_VAL **20**
22 Reading PART_ID, DIE_REV and UID **22**
23 Register description **22**
23.1 Register overview 22
23.2 Detailed register description 23
23.2.1 Register PART_ID 23
23.2.2 Register DIE_REV 23
23.2.3 Register UID 23
23.2.4 Register SYS_CTRL 24
23.2.5 Register SYS_STAT 24
23.2.6 Register SENS_RUN 25
23.2.7 Register SENS_START 25
23.2.8 Register SENS_STOP 26
23.2.9 Register SENS_STAT 26
23.2.10 Register T_VAL 27
23.2.11 Register H_VAL 27
24 Application information **28**
24.1 Typical application 28
25 Recommended operating conditions **28**
26 Soldering & storage information **29**
26.1 Soldering 29
26.2 Storage and handling 30
26.3 Reconditioning 31
26.3.1 After soldering 31
26.3.2 After extreme conditions 31
27 Package drawings **32**
28 Marking information **33**
29 Ordering information **33**
30 RoHS Compliance & ScioSense Green Statement **34**
31 Copyrights & Disclaimer **34**
32 Document status **35**
33 Revision information **36**

1 Quick start

A typical application circuit for ENS212A is shown on the top side of Figure 1. The quickest way to measure humidity and temperature is pseudo-coded at the bottom of the same figure.

More code examples with more detailed explanations can be found in chapter 21.

Basic application circuit



Pseudo code

```
Write(0x22,0x03)          // Start temperature and humidity measurement
Sleep(130)                // Wait for measurement to complete
Data = Read(0x30, 6)      // Read 6 bytes from register 0x30

// Extract Temperature and Humidity (little-endian, assumes 32-bit integers)
T = Data[1] << 8 | Data[0]
H = Data[4] << 8 | Data[3]

// Print temperature and humidity
Print(T * (1.0 / 64.0) - 273.15, "°C")
Print(H * (1.0 / 512.0), "%RH")
```

Figure 1: Typical application circuit (top) and pseudo code (bottom) for easy starting. For details see section 21.

Find code resources and drivers on:
- Arduino: https://github.com/sciosense/ens21x-arduino
- Embedded: https://github.com/sciosense/ens21x-embedded

Find symbols, footprints & CAD models on: www.snapeda.com

2 Block diagram

The internal block diagram of ENS212A is shown in Figure 2. The I²C (communication) interface is connected to a controller which acts as the command interpreter and as bus master of the internal Advanced Peripheral Bus (APB). The memory and sensors are slaves of the APB. The MTP memory is used to store the sensor's calibration parameters and unique ID.

To reduce power consumption the controller only powers the measurement engine when needed.

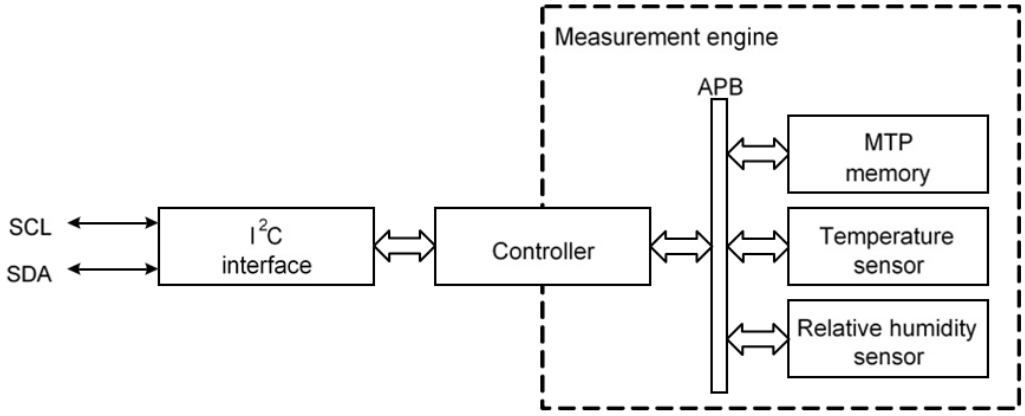


Figure 2: Functional blocks

3 Pin assignment

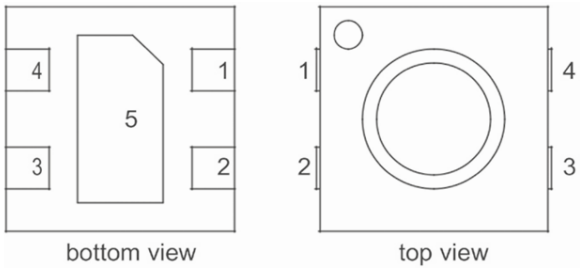


Figure 3: Pin diagram

Table 1: Pin description

Pins	Pin Name	Description
1	VDD	Supply voltage
2	SCL	I²C bus serial clock input (SCL)
3	SDA	I²C bus serial bidirectional data line (SDA)
4	VSS	Ground supply voltage; must be connected
5	VSS	Ground supply voltage; must be connected

4 Absolute maximum ratings

Stresses beyond those listed in this section may cause permanent damage to the device. These are - each at a time - stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated under Electrical characteristics is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 2: ENS212A absolute maximum ratings

Symbol	Parameter	Min	Max	Units	Comments
Electrical Parameters					
V_{DD}	Supply voltage	-0.30	4.60	V	
I_{lu}	Latch-up current		100	mA	I/O; $-0.5V_{DD} < V_I < 1.5V_{DD}$; $T_j < 125°C$
Electrostatic Discharge					
ESD_{HBM}	Human body model; all pins	±2000		V	JEDEC JS-001-2014
ESD_{CDM}	Charged model device; all pins	±750		V	JEDEC JS-002-2014
Operating and Storage Conditions					
MSL	Moisture Sensitivity Level	1			Maximum floor lifetime is unlimited
T_{STRG}	Storage temperature[1]	-40	125	°C	
T_A	Operating ambient temperature[2]	-40	125	°C	
H_A	Operating ambient relative humidity	0	100	%RH	

[1] Refer to section 25 Soldering & storage information for recommended storage conditions
[2] Refer to section 24 Recommended operating conditions for recommended operating conditions

5 Electrical characteristics

All limits are guaranteed. The parameters with min and max values are guaranteed with production tests or SQC (Statistical Quality Control) methods.

All values refer to T_A = 25 °C and at 1.80 V supply voltage, unless otherwise specified.

Table 3: Electrical characteristics

Symbol	Parameter	Conditions	Min	Typical[3]	Max	Unit
V_{DD}	Supply voltage	Max ripple 100mV_{PP} between 0-1MHz	1.71	1.80 (3.30)	3.60	V
I_{DD}	Supply current	Standby state		0.04 (0.35)		µA
		Continuous run mode		61 (62)		µA
		T and RH measurement at 1Hz		1.7 (2.1)		µA
V_{IH}	High-level input voltage		0.7×V_{DD}		V_{DD}+0.5	V
V_{IL}	Low-level input voltage		–0.5		0.3×V_{DD}	V
I_{OL}	Low-level output current	V_{OL} = 0.4V	3			mA
		V_{OL} = 0.6V	6			mA

[3] Values in parenthesis are for V_{DD}=3.30 V.

6 I²C timing characteristics

ENS212A is compliant to the I²C standard; it supports standard and fast mode as per I²C-bus specifications [UM10204, I²C-bus specification and user manual, Rev. 6, 4 April 2014].

7 Temperature sensor characteristics

Table 4: Temperature sensor characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{range}	Temperature range		-40		125	°C
T_{acc}	Temperature accuracy[4]	**T_A = -20°C to 70°C**		**±0.15**	**±0.2**	**°C**
		T_A = -40°C to 100°C		±0.4	±0.53	°C
		T_A=100ºC to 125ºC		±0.6	±0.8	°C
T_{res}	Temperature resolution			±0.016		°C
t_{resp}	Response time[5]	T step of 10°C by submersion (in 0°C to 70°C range); τ63% [6]		1		s
T_{rep}	Temperature repeatability	3σ of consecutive measurement values at constant conditions			±0.14	°C
ΔT	Temperature long term drift			±0.005		°C/year

[4] Typical and maximum accuracy specification refers to, respectively, 2 and 3 standard deviations, assuming normal distribution of accuracy errors. After industrial calibration, on each production lot through statistical analysis, a population of sensors is tested on typical room conditions (e.g. 25˚C, 45%RH) and only lots passing the verification qualify for customer deliveries.
[5] In an application the temperature response time depends on heat conductivity of the sensor PCB.
[6] 63% indicates that if a T step of 10˚C, e.g. from 20˚C to 30˚C is made, it will take t_{resp} seconds to reach 63% of that step.

8 Relative humidity sensor characteristics

Table 5: Relative humidity sensor characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
H_{range}	Relative humidity range		0		100	%RH
H_{acc}	Relative humidity accuracy	**T_A=25°C; RH=10% to 90%**		**±1.5**	**±2.25**	**%RH**
		T_A=25°C; RH=0% to 100%		±2	±3	%RH
H_{res}	Relative humidity resolution			0.03		%RH
t_{resp}	Response time[7]	RH step of 20% RH (in 40% to 80% range); τ_{63}; 1m/s flow; T_A = 25°C			3	s
H_{hys}	Relative humidity hysteresis	T_A=25°C; RH= 20%...90%; 2 hours exposure time		±0.55		%RH
H_{rep}	Relative humidity repeatability	3σ of consecutive measurement values at T_A=25°C and RH=40%		±0.1		%RH
ΔH	Relative humidity long term drift[8]	T_A=25°C		±0.25		%RH/year

9 System timing characteristics

Table 6: System timing characteristics

Symbol	Parameter	Conditions	Min	Typ.	Max	Unit
$t_{booting}$	Booting time[9]			1	1.2	ms
t_{conv}	Conversion time	T and RH, single shot (includes $t_{booting}$)		30	32	ms
		T and RH, continuous		55	59	ms

[7] Performance at device level only. The application environment may influence final performance.
[8] Values are linearized averages over the lifetime of the product. Due to non-linear behavior a larger drift is expected in the first years.
[9] Time in transient state booting (see Figure 4).

10 Functional description

The ENS212A integrates two sensor blocks: temperature and relative humidity.

The device is normally in the *standby* state (Figure 4): the measurement engine (see Figure 2) is unpowered, but the I²C interface is operational and register write/read operation can be performed. When a measurement command is given, the device is first *booting* to *active* then it starts a measurement. When the measurement is completed, the device returns to the *standby* state. Since the I²C interface is operational in *standby*, the measurement result can be read out.

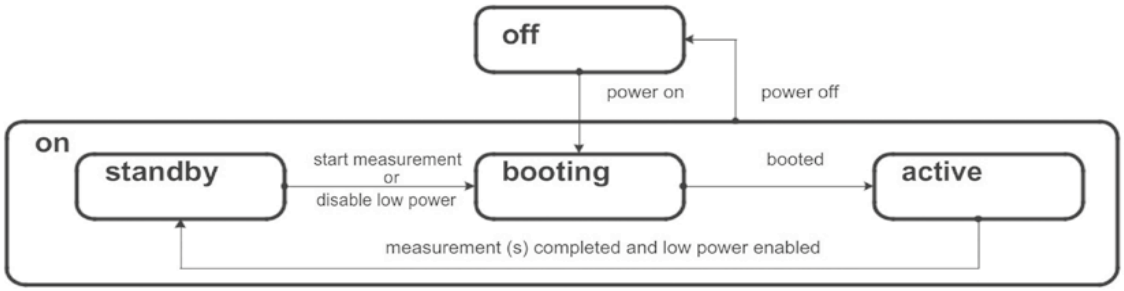


Figure 4: ENS212A power states

In continuous run mode (see Register SENS_RUN) or when low power is disabled (see Register SYS_CTRL), the device remains in *active* state.

The system power status is observable (see Register SYS_STAT).

When powering up from *off* state, the device goes first to *booting* status, to *active*, and then falls immediately back to *standby* (since no measurement is pending, and by default low power is enabled). Note that the *booting* state is a transient state (the system automatically transitions to state *active*); the booting time is given in Table 6.

11 Temperature sensor

The temperature sensor block (Figure 5) determines the ambient temperature, and outputs a calibrated value in Kelvin.

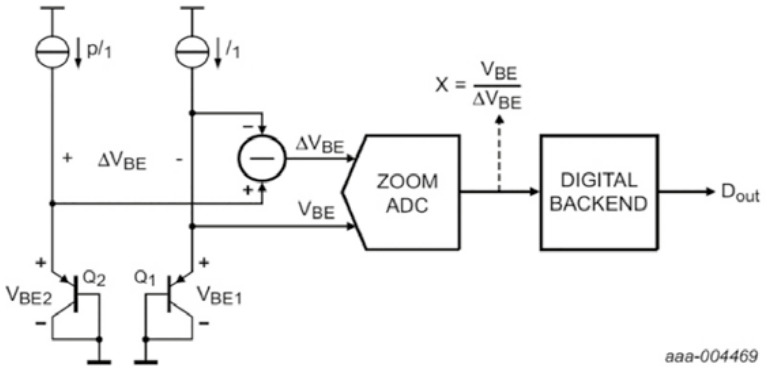


Figure 5: Bandgap for temperature measurement

The temperature is measured using a high-precision (12 bits) zoom-ADC. The analog part is able to measure a strongly temperature dependent X = VBE/ΔVBE. The X is found by first applying a

coarse search (successive approximation), and then a sigma-delta in a limited range. The accuracy of the sensor is shown in Figure 6. The conversion time is shown in Table 6.

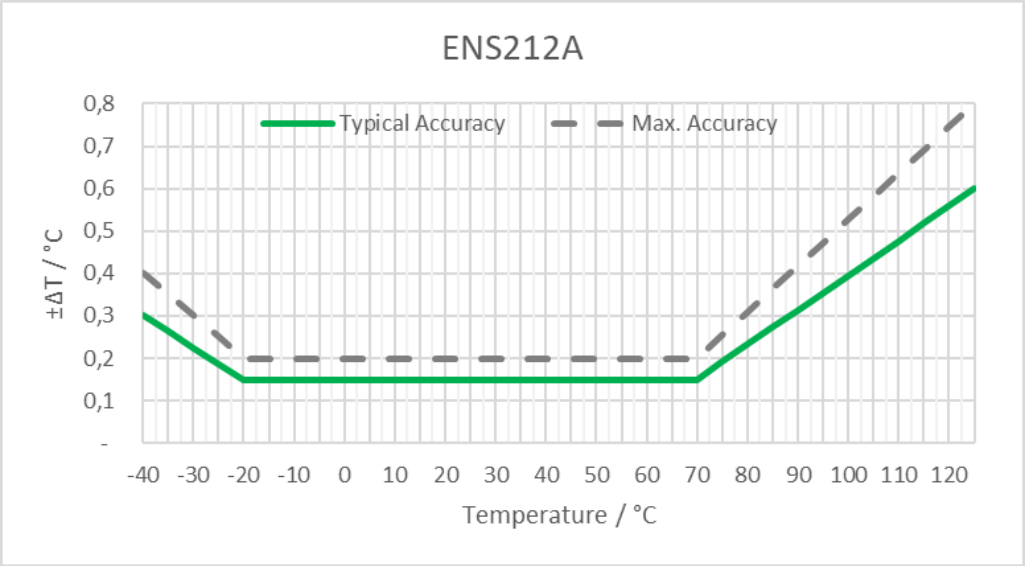


Figure 6: Temperature sensor accuracies

12 Relative humidity sensor

The relative humidity sensor as shown in Figure 7 determines the ambient relative humidity and outputs a calibrated value in %RH. The transducer (C_X on the top left) consists of a large-area capacitor covered with humidity-sensitive material. The capacitance change is proportional to the change in relative humidity and has a linear dependence on temperature. The capacitance is measured by a high-precision 2nd order sigma-delta converter.

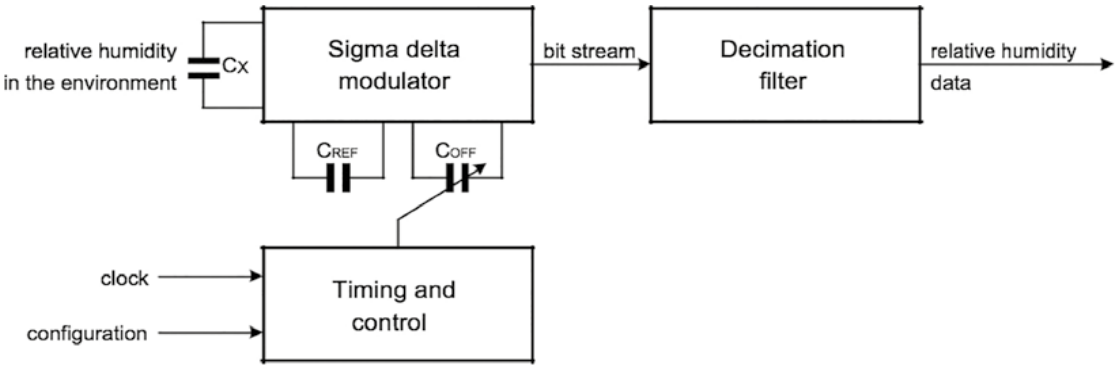


Figure 7: Relative humidity sensor

Reading the relative humidity sensor will output a temperature compensated value. The accuracy of the sensor is shown in Figure 8. The conversion time is shown in Table 6.

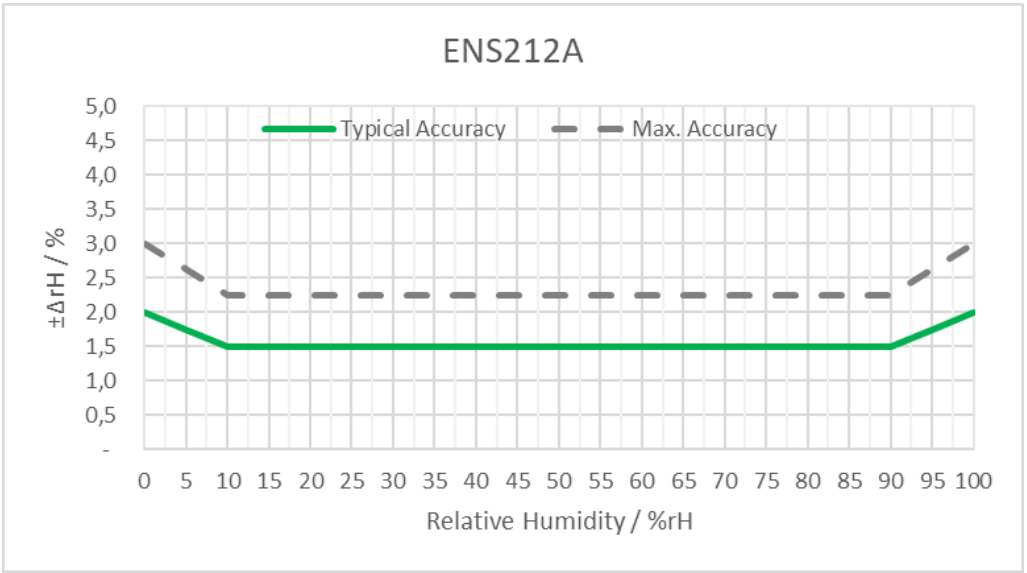


Figure 8: Relative humidity sensor accuracies[10]

[10] At 25°C.

13 RH accuracy at various temperatures

Typical RH accuracy at 25°C is defined in Table 5 and Figure 8. The relative humidity accuracy has also been evaluated at temperatures other than 25°C.

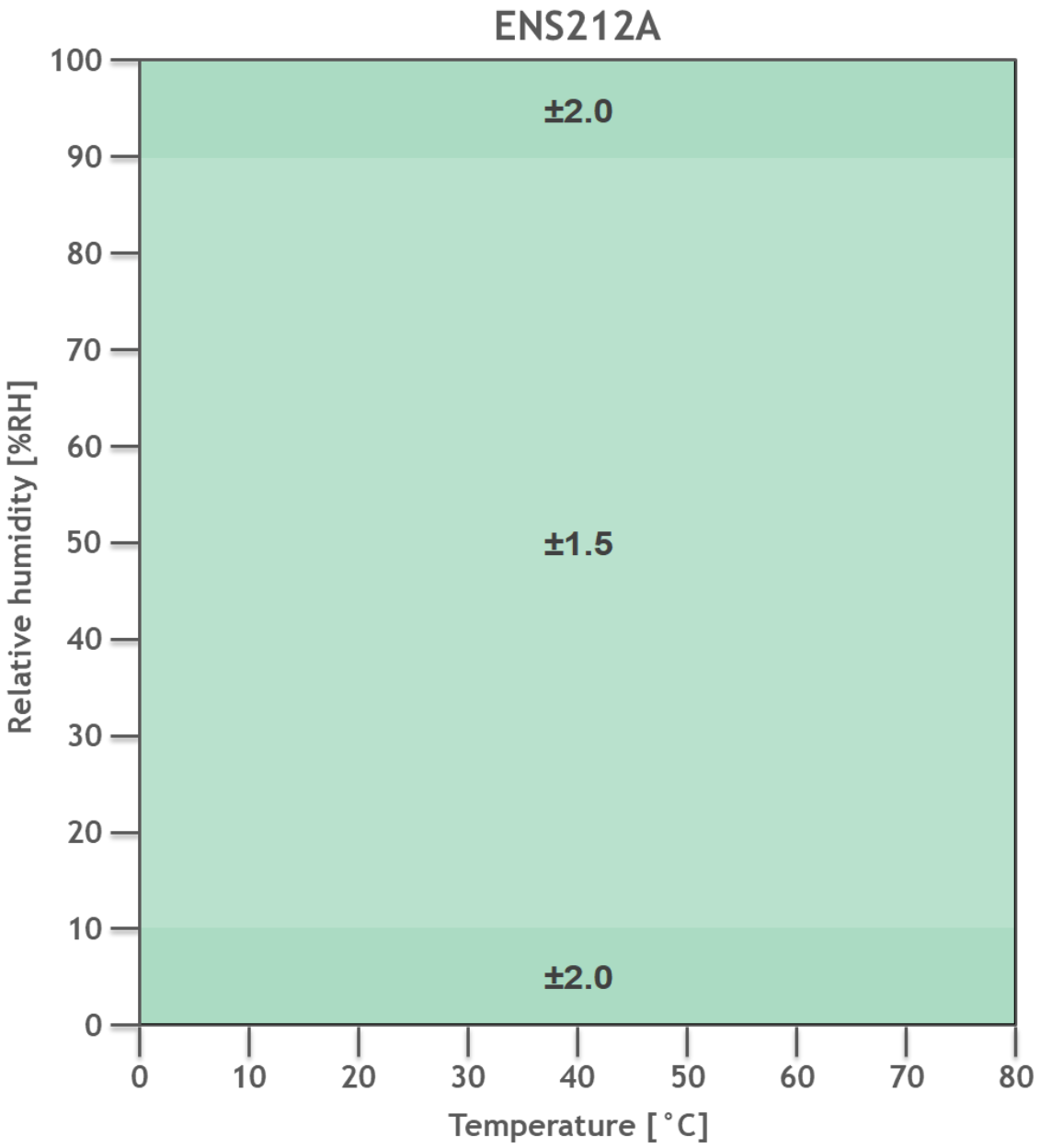


Figure 9: Accuracy of relative humidity measurements (%RH) as function of temperature and relative humidity[11]

[11] Typical values.

14 The I²C interface

The ENS212A is an I²C slave device. The I²C interface supports standard (100kbit/s) and fast (400kbit/s) mode.

Details on the I²C protocol are according to I²C-bus specifications *[UM10204, I²C-bus specification and user manual, Rev. 6, 4 April 2014]*.

The devices apply all mandatory I²C protocol features for slaves: START, STOP, Acknowledge, 7-bit slave address. ENS212A does not use clock stretching.

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).

15 I²C operations on registers

The ENS212A uses a register model to interact with it. This means that a I²C master can write a value to one of the registers of a slave, or that it can read from one of the registers of the slave. In the ENS212A, registers are addressed using 1 byte. The values stored in a register are also 1 byte. However, the ENS212A implements “auto increment” which means that it is possible to read, for example, two bytes by supplying the address of the first byte and then reading two bytes.

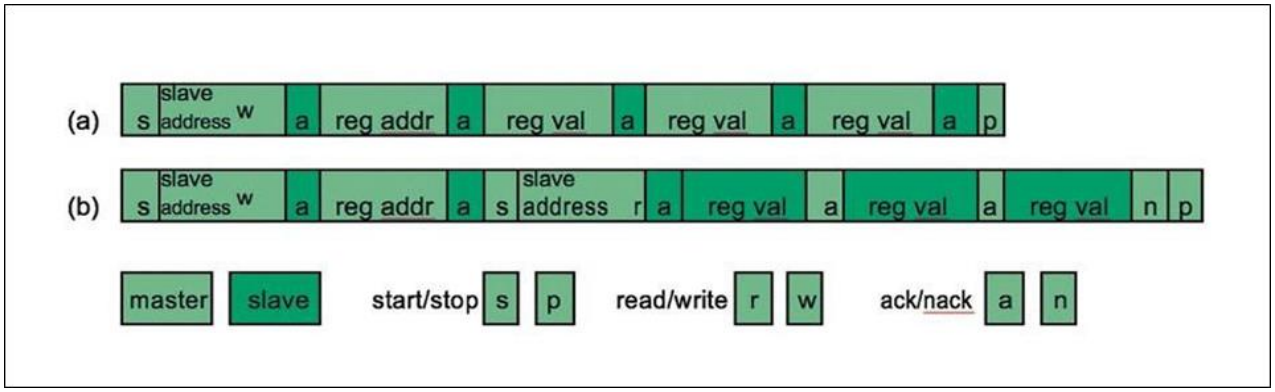


Figure 10: I²C transaction formats

A typical *write* transaction (see Figure 10) therefore has the following format. The master initiates a transaction with a so-called start condition “s”. This blocks the bus. Next, the master sends the 7-bit ENS212A slave address followed by a 1-bit direction (a 0 indicating write “w”).

This byte is acknowledged with “a” by the slave. The master continues by sending the 8-bit *register address*, which is acknowledged by the slave.

This register address is stored in an internal CRA register (“Current Register Address”). Finally, the master sends the 8-bit *register value*, which is acknowledged by the slave (or nack’ed when the address is not writeable). This value is written to the register pointed to by the CRA, and the CRA is incremented by 1.

Optionally, the master sends more 8-bit values, for the next registers (auto incrementing CRA), each of which is (n)ack'ed by the slave. Finally, the master generates a stop condition "p", unblocking the bus for other transactions.

A *read* transaction (see Figure 10) starts with a write (of the register address), followed by a read. Consequently, it has the following format. The master initiates the transaction with a start condition. Next, the master sends the 7-bit ENS212A *slave address* followed by a 1-bit direction (a 0 indicating write). This byte is acknowledged by the slave. The master continues by sending the 8-bit *register address*, which is acknowledged by the slave and stored in the CRA register.

Then the master sends another start condition (a so-called repeated start condition, keeping the bus blocked) followed by the 7 bits ENS212A slave address followed by a 1-bit direction (a 1 indicating read "r"), which is acknowledged by the slave. Next, the slave sends an 8-bit *register value* from the register pointed to by the CRA register, and the CRA is incremented by 1. This byte is acknowledged by the master. The master may read another 8 bits (auto increment feature) from the slave and acknowledge that, until the master sends a nack "n" followed by a stop to unblock the bus.

The ENS212A has an 8-bit address space, potentially addressing 256 registers. In reality, only few addresses are actually backed by a register (see Register overview). All other addresses are reserved. A write transaction to a reserved (or read-only) register causes a not-acknowledge. A read transaction for a reserved register will return a 0.

16 The I²C slave address

The ENS212A is an I²C slave device with fixed slave addresses (Table 7). This means that the first byte after a start condition is 1000 101x, where x indicates the data direction, so 0x8A (1000 1010) for write and 0x8B (1000 1011) for read.

Table 7: ENS212A slave addresses

Product Variant	Slave Address	Byte after Start Condition (read)	Byte after Start Condition (write)
ENS212A	0x45	0x8B	0x8A

17 Sensor control

The ENS212A embeds a temperature and a relative humidity sensor. Both sensors have two run modes: single shot run mode and continuous run mode (enabled via SENS_RUN), see Figure 11.

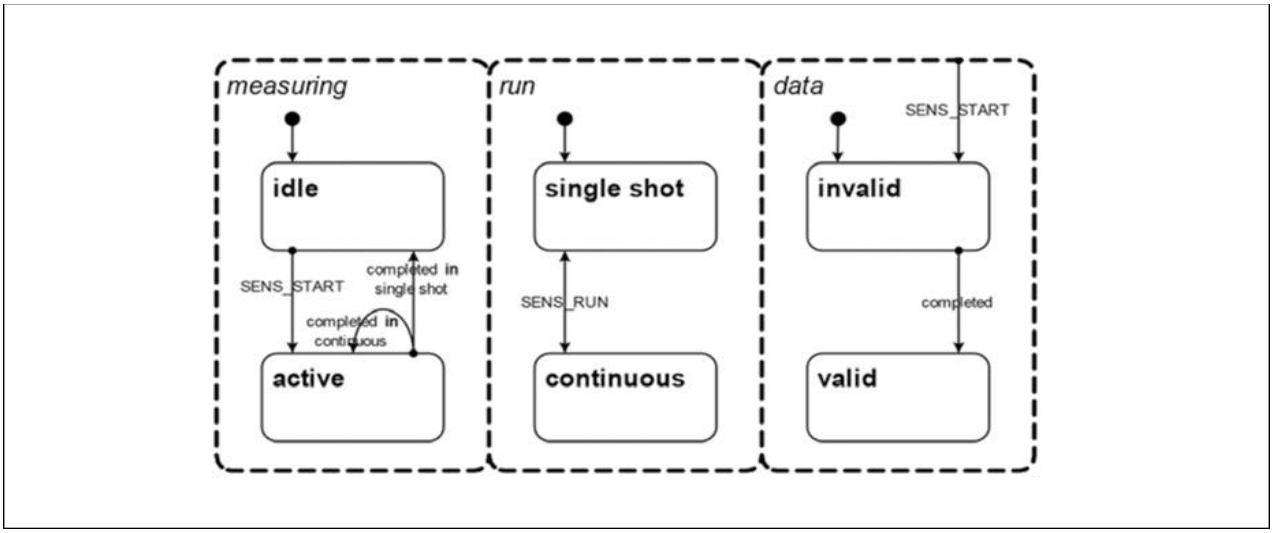


Figure 11: The ENS212A sensor modes

When in the *single shot* run mode, starting a measurement is under control of the master. By default a sensor is idle; it can be started by writing a 1 to the corresponding bit in SENS_START. After a start, the sensor stops when the measurement is completed. Whether a sensor is idle or active measuring can be detected by reading SENS_STAT. The measured values can be obtained via their respective readout registers (T_VAL and H_ VAL). Writing to SENS_STOP in single shot has no effect.

When in the *continuous* run mode, the sensor performs measurement after measurement after a 1 is written to the corresponding bit in SENS_START. The result of each measurement is stored in the aforementioned readout registers. Writing 1 to the corresponding bit in SENS_STOP stops the repeat cycle after the ongoing measurement is completed.

The device operates in a stepwise way. In each step, either one or both sensors are active. The step ends when the measurement(s) are completed. For the next step, the device inspects its register settings, and either one or both sensors are activated again, or there is no measurement request, and the device goes into standby (unless low power is disabled by SYS_ CTRL).

This means that multiple writes to START during a step have no effect; the measurement is started once, and only a write to START after the measurement has completed starts the measurement again. Similarly, multiple writes to STOP have no effect; when the measurement completes (in continuous mode) the stop request is effectuated once. When START and STOP are both requested, the measurement is started, and when completed, stopped.

18 Sensor timing

There are differences between single shot measurements and continuous measurements. Figure 12 shows the timing of a single shot T measurement.

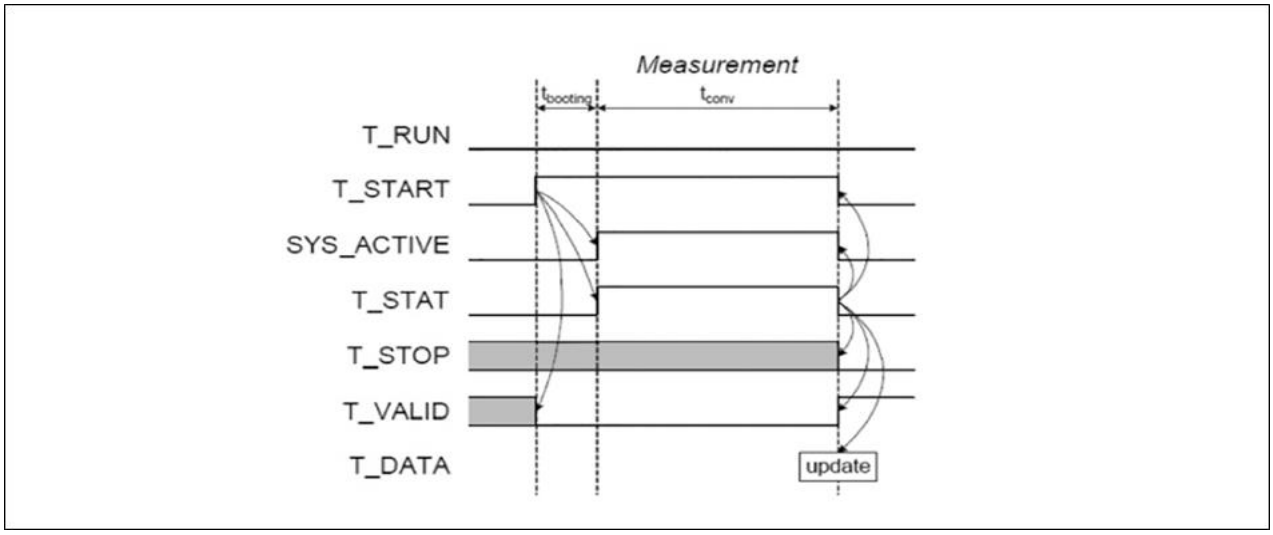


Figure 12: Single shot temperature measurement

Signal T_RUN is written low to select a single shot measurement. Note that T_STOP is typically low (cleared by a previous measurement), but its state is ignored in a single shot measurement. T_START is written high to start measuring: T_VALID in T_VAL is cleared and the device starts booting to active. Once active SYS_ACTIVE goes high, and measurement starts (T_STAT goes high).

When the measurement is completed (T_STAT goes low) the data register (T_DATA) becomes valid (T_VALID goes high) and the device goes back to standby (SYS_ACTIVE goes low). The T_START and T_STOP are cleared. Figure 13 shows the timing of a continuous T measurement.

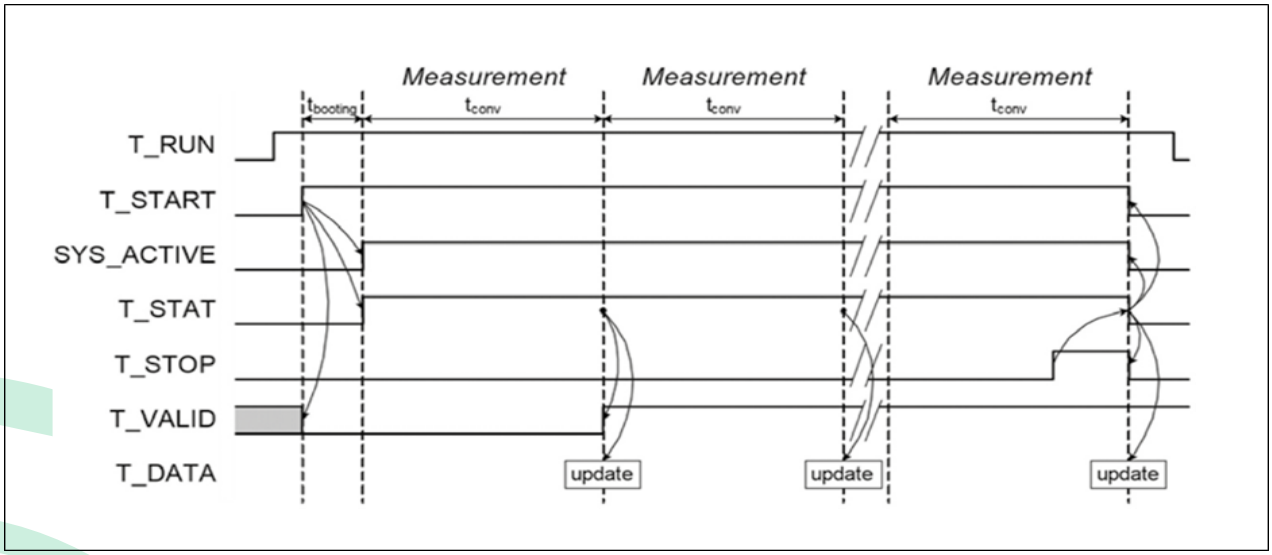


Figure 13: Continuous temperature measurement

Signal T_RUN is written high to select a continuous measurement. Note that T_STOP is typically low (cleared by a previous measurement), and it should stay low otherwise continuous mode will

stop after one measurement. T_START is written high to start measuring: T_VALID in T_VAL is cleared and the device starts *booting* to *active*. Once *active* SYS_ACTIVE goes high, and measurement starts (T_STAT goes high).

When the first measurement is completed the data register (T_DATA) becomes valid (T_VALID goes high), and the device starts a new measurement. When the next measurement is completed the data register (T_DATA) is updated; T_VALID stays high. The device starts a new measurement.

At some point in time, a stop command is given (T_STOP is written high). As soon as the current measurement is completed, the data register (T_DATA) is once more updated and the device goes back to *standby* (SYS_ACTIVE goes low). The T_START and T_STOP are cleared.

Note that writes to the SENS_XXX registers only take effect when no measurement is ongoing. In other words, measurements are always sequential (so we can have three types: T only, RH only or T and RH and changes occur when the measurements are finished.

19 Sensor readout registers

The sensor readout registers (T_VAL and H_VAL) consist of three parts: the actual measured data, a valid flag and a checksum (see Figure 14). It is not mandatory to read the valid flag or the checksum when reading the data.

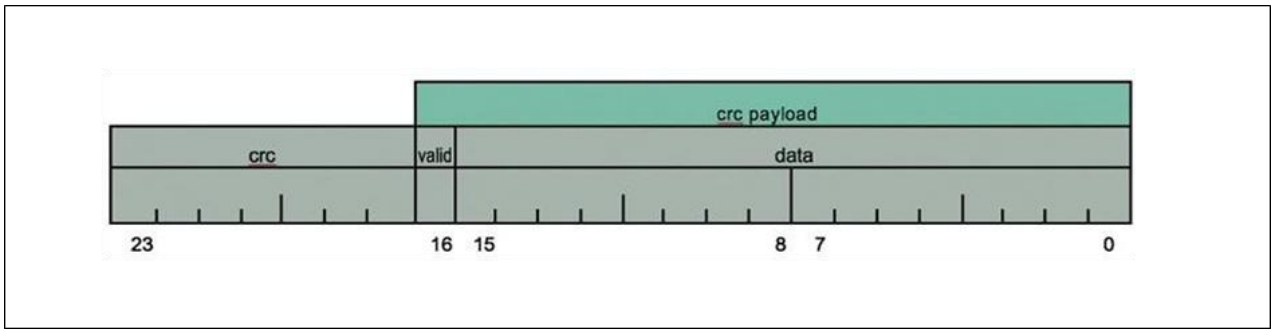


Figure 14: Layout of the sensor readout registers

The *checksum* is a cyclic redundancy check over the data and the valid flag; the stored checksum is the result of CRC-7[12] with 0x7F as initial vector (i.e. with all bits flipped), see Computing CRC-7 for sample C code.

The valid flag is cleared when a measurement is started (irrespective of the run mode). Once the measurement is completed the valid flag is set. In continuous mode, a new measurement is then started without clearing the valid flag; so data is always valid after the first measurement (but it might be several milliseconds old).

The data field is a 16 bits fixed point number, whose format and unit depends on the sensor (see Register T_VAL and Register H_VAL).

To ensure consistent view, these multi-byte readout registers are double buffered. When the first byte (i.e. the byte with the lowest register address) is read, the device copies all bytes from

[12] Polynomial x7+x3+1, see https://en.wikipedia.org/wiki/Cyclic_redundancy_check

the measurement registers to the I²C registers, and then the value from the first I²C register is returned. Reads to the other bytes of the multi-byte register (i.e. with higher register addresses) are always directly from the I²C registers.

20 Computing CRC-7

CRC algorithm uses a 7-bit polynomial (see lines 4, 5, and 6), and a 17-bit payload. The `crc7()` function below uses the following constants defining the CRC width, (the coefficients of the) polynomial and the initial vector (start value of the CRC), and some constants describing the payload data size.

```
//                          7654   3211
// Polynomial       0b     1000   1001 ~ x^7+x^3+x^0
//                  0x     8      9
#define CRC7WIDTH   7      // 7 bits CRC has polynomial of 7th order (has 8 terms)
#define CRC7POLY    0x89   // The 8 coefficients of the polynomial
#define CRC7IVEC    0x7F   // Initial vector has all 7 bits high
// Payload data
#define DATA7WIDTH  17
#define DATA7MASK   ((1UL<<DATA7WIDTH)-1) // 0b 1 1111 1111 1111 1111
#define DATA7MSB    (1UL<<(DATA7WIDTH-1)) // 0b 1 0000 0000 0000 0000
```

The `crc7(val)` function returns the CRC-7 of a 17-bit value `val`.

```
// Compute the CRC-7 of 'val' (should only have 17 bits)
uint32_t crc7( uint32_t val ) {
       // Setup polynomial
       uint32_t pol= CRC7POLY;
       // Align polynomial with data
       pol = pol << (DATA7WIDTH-CRC7WIDTH-1);
       // Loop variable (indicates which bit to test, start with highest)
       uint32_t bit = DATA7MSB;
       // Make room for CRC value
       val = val << CRC7WIDTH;
       bit = bit << CRC7WIDTH;
       pol = pol << CRC7WIDTH;
       // Insert initial vector
       val |= CRC7IVEC;
       // Apply division until all bits done
       while( bit & (DATA7MASK<<CRC7WIDTH) ) {
              if( bit & val ) val ^= pol;
              bit >>= 1;
              pol >>= 1;
              }
       return val;
}
```

Suppose that T_VAL (address 30, 31 and 32) reads FD 49 0B, corresponding (little endian) with the number 0B49FD, see Figure 15. This leads to a CRC of 05 over a payload of 149FD. See the next paragraph for details on processing this data.

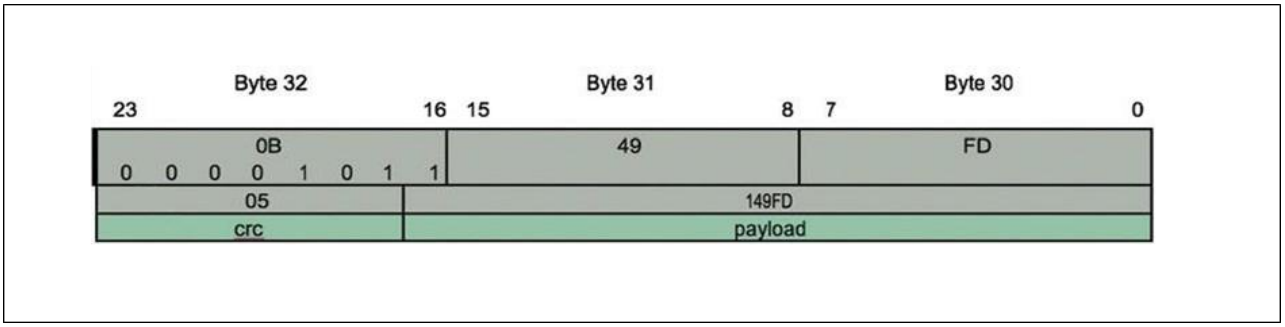


Figure 15: T_VAL readout

21 Processing T_VAL and H_VAL

This paragraph shows a possible implementation of reading T and RH.

The following fragment starts a combined single shot measurement on ENS212A, waits and reads the measurement results. It assumes the availability of `i2c_reg_write` and `i2c_reg_read` primitives as well as a sleep routine (`rtk_tsk_sleep`). The format specifiers in the printf's are kept simple (`%d` instead of `%ld` or even `PRId32` from `inttypes.h`); they need adaptation on e.g. 16-bit platforms.

```
// Record I²C transaction status
bool i2c_ok= true;
// Start T and H (write 03 to register 22 in device 8A)
uint8_t wbuf[]= { 0x03 };
i2c_ok &= i2c_reg_write(0x8A, 0x22, wbuf, sizeof wbuf );
// Wait for measurements to complete
#define CONVERSION_TIME_T_H_MS   130
rtk_tsk_sleep(CONVERSION_TIME_T_H_MS);
// Read T and H (read 6 bytes starting from 0x30 in device 8B)
uint8_t rbuf[6];
i2c_ok &= i2c_reg_read(0x8A, 0x30, rbuf, sizeof rbuf );
// Extract T_VAL and H_VAL (little endian), assumes 32 bits word size
uint32_t  t_val=  (rbuf[2]<<16)  +  (rbuf[1]<<8)  +  (rbuf[0]<<0);  uint32_t  h_val=
(rbuf[5]<<16) + (rbuf[4]<<8) + (rbuf[3]<<0);
```

The following fragment processes the T measurement as available in `t_val`. It relies on the `crc7()` function as shown previously.

```
// Extract (and print) the fields
uint32_t t_data = (t_val>>0) & 0xffff;
uint32_t t_valid= (t_val>>16) & 0x1;
uint32_t t_crc = (t_val>>17) & 0x7f;
printf("ENS212A: T: %06x %02x %01x %04x\n", t_val, t_crc, t_valid, t_data); // Check the CRC
uint32_t t_payl = (t_val>>0 ) & 0x1ffff; bool t_crc_ok= crc7(t_payl)==t_crc;
// Convert to float (and print)
float TinK = (float)t_data / 64; // Temperature in Kelvin
float TinC = TinK - 273.15; // Temperature in Celsius
float TinF = TinC * 1.8 + 32.0; // Temperature in Fahrenheit
printf("ENS212A: T: (i2c=%d crc=%d valid=%d) %5.1fK %4.1fC %4.1fF\n", i2c_ok, t_crc_ok, t_valid, TinK, TinC, TinF );
```

The following fragment processes the RH measurement as available in `h_val`. It is similar to the `t_val` processing.

```
// Extract (and print) the fields
uint32_t h_data = (h_val>>0 ) & 0xffff;
uint32_t h_valid = (h_val>>16) & 0x1;
uint32_t h_crc = (h_val>>17) & 0x7f;
printf("ENS212A: H: %06x %02x %01x %04x\n", h_val, h_crc, h_valid, h_data ); // Check the CRC
uint32_t h_payl = (h_val>>0 ) & 0x1ffff;
bool h_crc_ok= crc7(h_payl)==h_crc;
// Convert to float (and print)
float H = (float)h_data/512;     // relative humidity (in %)
printf("ENS212A: H: (i2c=%d crc=%d valid=%d) %2.0f%%\n", i2c_ok, h_crc_ok, h_valid, H);
```

If registers 30 to 35 contained *fd 49 0b 6c 2e f5* (i.e. T_VAL in blue and H_VAL in green), the code would print

ENS212A: T: *0b49fd* 05 1 49fd

ENS212A: T: (i2c=1 crc=1 valid=1) 296.0K 22.8C 73.0F

ENS212A: H: *f52e6c* 7a 1 2e6c

ENS212A: H: (i2c=1 crc=1 valid=1) 23%

22 Reading PART_ID, DIE_REV and UID

The first 3 registers (PART_ID, DIE_REV and UID) are only available in active state. There are two ways to read them:

- Dedicated read action
  - Disable low power (set LOW_POWER to 0)
  - Wait for $t_{booting}$ to get into active state (check SYS_ACTIVE to be 1)
  - Read the ID register(s)
  - Re-enable low power (set LOW_POWER to 1)

- Piggybacking on a measurement
  - Start a measurement (write 0b01, 0b10, or 011 to SENS_START)
  - Wait for $t_{booting}$ to get into active state (check SYS_ACTIVE to be 1)
  - Read the ID register(s)
  - Ensure the device is still in active state (check SYS_ACTIVE to be 1)

23 Register description

This section describes the I²C registers of the ENS212A.

23.1 Register overview

Note that some registers are actually spread over multiple addresses. For example, T_VAL at address 30 is spread over 3 addresses (its "Size" is 3). This could be rephrased as follows: there are three registers T_VAL0, T_VAL1, and T_VAL2 at addresses 30, 31, and 32 respectively.

Table 8: Register overview

Address	Name	Size	Access	Description
0x00	PART_ID	2	Read (active only)	Identifies the part as ENS212A
0x02	DIE_REV	2	Read (active only)	Identifies the die revision version
0x04	UID	8	Read (active only)	Unique identifier
0x0C	<reserved>	4		
0x10	SYS_CTRL	1	Read/Write	System configuration
0x11	SYS_STAT	1	Read	System status
0x12	<reserved>	14		
0x21	SENS_RUN	1	Read/Write	The run mode (single shot or continuous)
0x22	SENS_START	1	Write	Start measurement

Address	Name	Size	Access	Description
0x23	SENS_STOP	1	Write	Stop continuous measurement
0x24	SENS_STAT	1	Read	Sensor status (idle or measuring)
0x25	<reserved>	11		
0x30	T_VAL	3	Read	Temperature readout
0x33	H_VAL	3	Read	Relative humidity readout
0x36	<reserved>	202		

23.2 Detailed register description

23.2.1 Register PART_ID

This 2-byte register identifies the part number in little endian (ENS212A). This register is only available in *active* state; see Reading PART_ID, DIE_REV and UID for instructions of reading it.

Table 9: Register PART_ID (address 0x00)

Address 0x00		PART_ID		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
15:0	PART_ID	0x0212	Read	Identifies this device as an ENS212(A)

23.2.2 Register DIE_REV

This 2-byte register identifies the die revision number of the ENS212A. This register is only available in active state; see Reading PART_ID, DIE_REV and UID for instructions of reading it.

Table 10: Register DIE_REV (address 0x02)

Address 0x02		DIE_REV		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
15:0	DIE_REV	0x0001	Read	Identifies the die revision version number

23.2.3 Register UID

This 8-byte register uniquely identifies a single device among all ENS21x devices. This register is only available in active state; see Reading PART_ID, DIE_REV and UID for instructions of reading it.

Table 11: Register UID (address 0x04)

Address 0x04		UID		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
63:0	UID	N/A	Read	Unique device ID

23.2.4 Register SYS_CTRL

This 1-byte register controls the system.

Table 12: Register SYS_CTRL (address 0x10)

Address 0x10		SYS_CTRL		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7	RESET	0	Write	Write 1 to reset the device
6:1	<reserved>	0b000000	Read/Write	Keep to 0's
0	LOW_ POWER	0b1	Read/Write	Controls the automatic low power. 0: Disabled (device stays in active) 1: Enabled (device goes to standby when measurement complete)

23.2.5 Register SYS_STAT

This 1-byte register indicates the system status.

Table 13: Register SYS_STAT (address 0x11)

Address 0x11		SYS_STAT		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:1	<reserved>	0b0000000	Read	Reads 0's
0	SYS_ACTIVE	0b1	Read	The system power state 0: System is in *standby* or *booting* state 1: System is in *active* state

23.2.6 Register SENS_RUN

This 1-byte register configures the run modes (*single shot* or *continuous*) of the sensors.

Table 14: Register SENS_RUN (address 0x21)

Address 0x21		SENS_RUN		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:2	<reserved>	0b000000	Read/Write	Keep to 0's
1	H_RUN	0b0	Read/Write	The run mode of the relative humidity sensor 0: Relative humidity sensor operates in single shot mode 1: Relative humidity sensor operates in continuous mode
0	T_RUN	0b0	Read/Write	The run mode of the temperature sensor 0: Temperature sensor operates in single shot mode 1: Temperature sensor operates in continuous mode

23.2.7 Register SENS_START

This 1-byte register starts a measurement for the sensors.

Table 15: Register SENS_START (address 0x22)

Address 0x22		SENS_ START		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:2	<reserved>	0b000000	Read/Write	Keep to 0's
1	H_START	0b0	Read/Write	Write a 1 to start a relative humidity sensor measurement Writing 0 has no effect (helps in multiple access)
0	T_START	0b0	Read/Write	Write a 1 to start a temperature sensor measurement Writing 0 has no effect (helps in multiple access)

23.2.8 Register SENS_STOP

This 1-byte register stops a continuous measurement for the sensors.

Table 16: Register SENS_ STOP (address 0x23)

Address 0x23		SENS_ STOP		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:2	<reserved>	0b000000	Write	Write 0's
1	H_STOP	0b0	Write	Write a 1 to stop a continuous relative humidity sensor measurement Writing 0 has no effect (helps in multiple access)
0	T_STOP	0b0	Write	Write a 1 to stop a continuous temperature sensor measurement Writing 0 has no effect (helps in multiple access)

23.2.9 Register SENS_STAT

This 1-byte register indicates the measuring status (idle or active) of the sensors.

Table 17: Register SENS_STAT (address 0x24)

Address 0x24		SENS_ STAT		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
7:2	<reserved>	0b000000	Read	Write 0's
1	H_STAT	0b0	Read	Indicates the measuring status of the relative humidity sensor 0: Relative humidity sensor is idle (not measuring) 1: Relative humidity sensor is active measuring
0	T_STAT	0b0	Read	Indicates the measuring status of the temperature sensor 0: Temperature sensor is idle (not measuring) 1: Temperature sensor is active measuring

23.2.10 Register T_VAL

This 3-byte register contains the last measured temperature data. Furthermore, it has a data valid flag and a CRC over the former two. Note that these bytes are double buffered; they are latched in by accessing the first byte, see Sensor readout registers for details.

See Section Processing T_VAL and H_VAL for example code of processing this register.

Table 18: Register T_VAL (address 0x30)

Address 0x30		T_VAL		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
23:17	T_CRC	-	Read	CRC over T_DATA and T_VALID
16	T_VALID	-	Read	Data valid indication (1 means T_DATA is valid)
15:0	T_DATA	-	Read	Last measured temperature, stored as a little endian 16 bits unsigned value in 1/64 Kelvin

23.2.11 Register H_VAL

This 3-byte register contains the last measured relative humidity data. Furthermore, it has a data valid flag and a CRC over the former two. Note that these bytes are double buffered; they are latched in by accessing the first byte, see Sensor readout registers for details.

See Processing T_VAL and H_VAL for example code of processing this register.

Table 19: Register H_VAL (address 0x33)

Address 0x33		H_VAL		
Bits	**Field Name**	**Default**	**Access**	**Field Description**
23:17	H_CRC	-	Read	CRC over H_DATA and H_VALID
16	H_VALID	-	Read	Data valid indication (1 means H_DATA is valid)
15:0	H_DATA	-	Read	Last measured relative humidity, stored as a little endian 16 bits unsigned value in 1/512% RH

24 Application information

24.1 Typical application

Figure 16 shows a typical application.

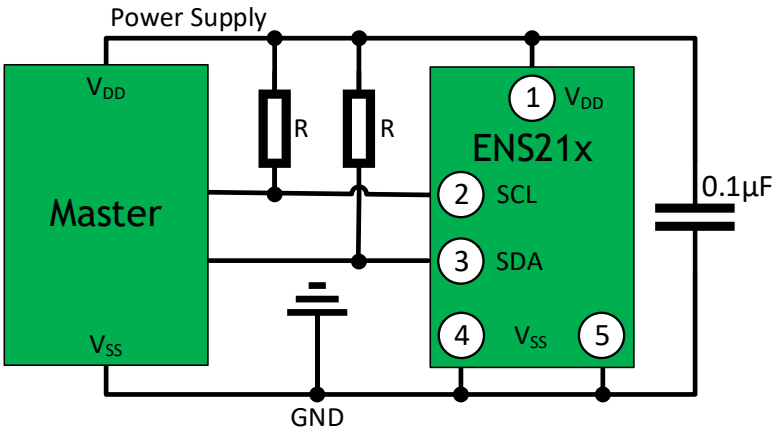


Figure 16: ENS212A typical application

25 Recommended operating conditions

The recommended temperature and relative humidity operating range for the ENS212A is 5 °C to 60 °C and 20 %RH to 80 %RH, see Figure 17.

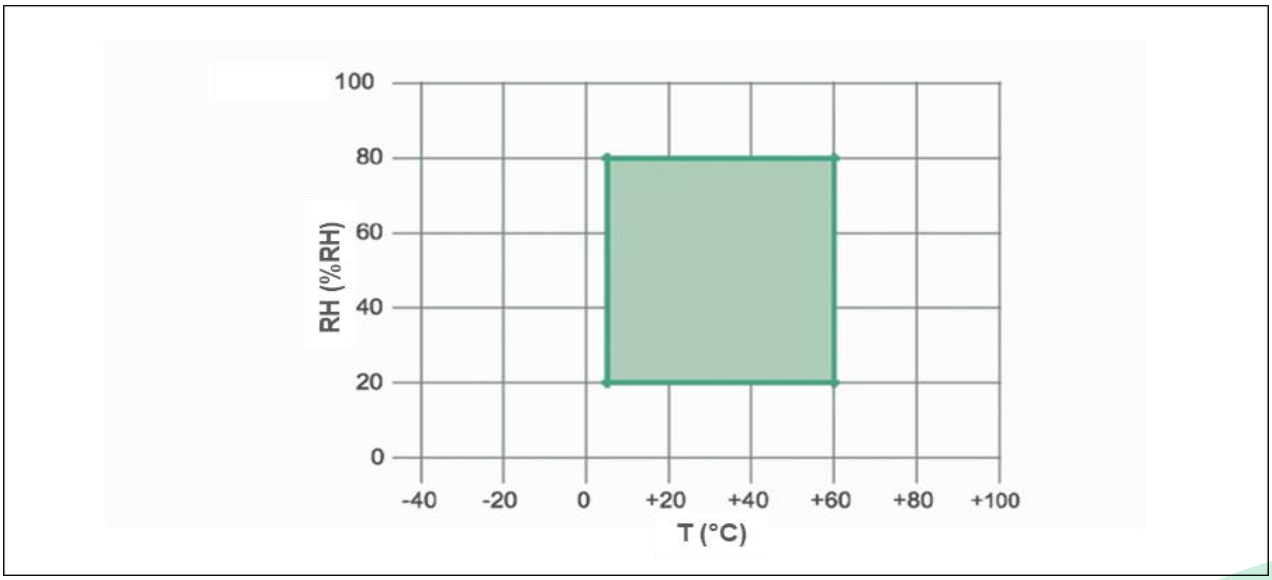


Figure 17: Recommended operating conditions

Long term exposure outside these recommended operating conditions may temporarily offset the relative humidity readout. After such exposure, the device will slowly return to its accuracy limits at 25 °C (can be a matter of hours or weeks, depending on stress conditions). Re-conditioning (bake + hydration, see section 25.3) will accelerate kinetics of returning to its accuracy limits at 25 °C.

26 Soldering & storage information

26.1 Soldering

The ENS212A uses a cavity package. This package can be soldered using a standard reflow process in accordance with IPC/JEDEC J-STD-020D. See picture below.

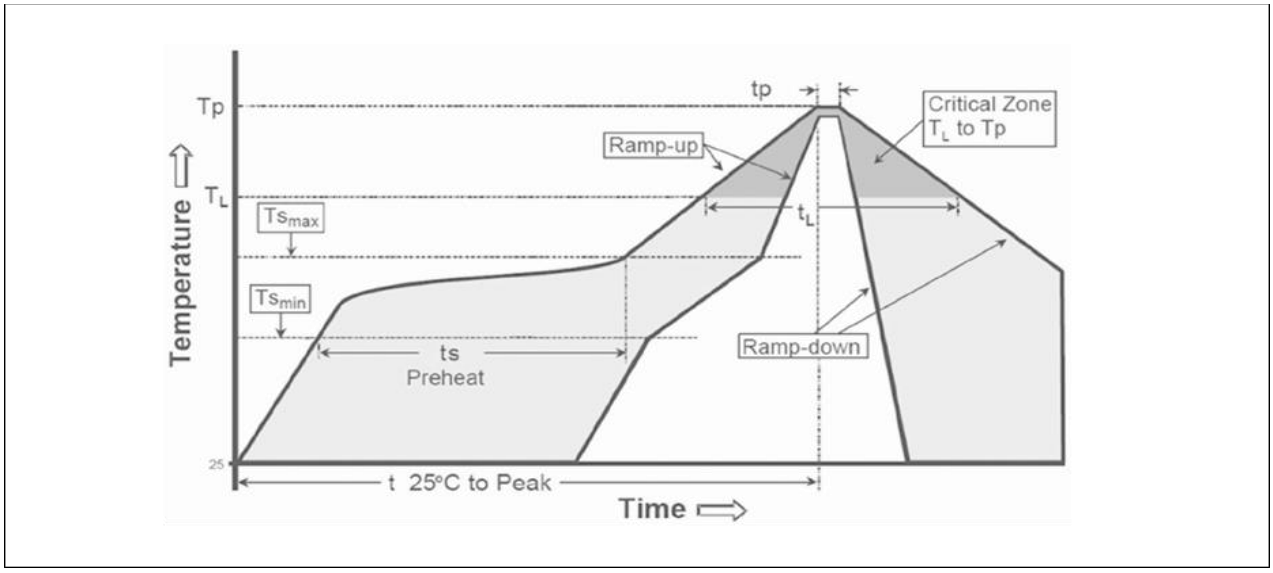


Figure 18: Soldering recommendations

The detailed settings for the reflow profile are shown in Table 20.

Table 20: Soldering recommendations table

Reference Parameter	Rate / Unit
T_P	260 °C
t_P (time within 5°C of T_P)	20-40 seconds
T_L	217 °C
t_L	60-150 seconds
Ts_{max}	200 °C
Ts_{min}	150 °C
t_S (preheat)	60 to 180 seconds
t 25°C to Peak	8 minutes max.
ramp up	3 °C/s max.
ramp down	6 °C/s max.

It is recommended to use a no-clean solder paste for soldering the sensor component on a PCB. There should not be any board wash process, to prevent the sensor area to get in contact with cleaning agents or other liquid materials.

The recommended ENS212A landing pattern can be found in the drawing below in blue. A 100 µm thick stencil can be used and the stencil apertures are indicated in violet.

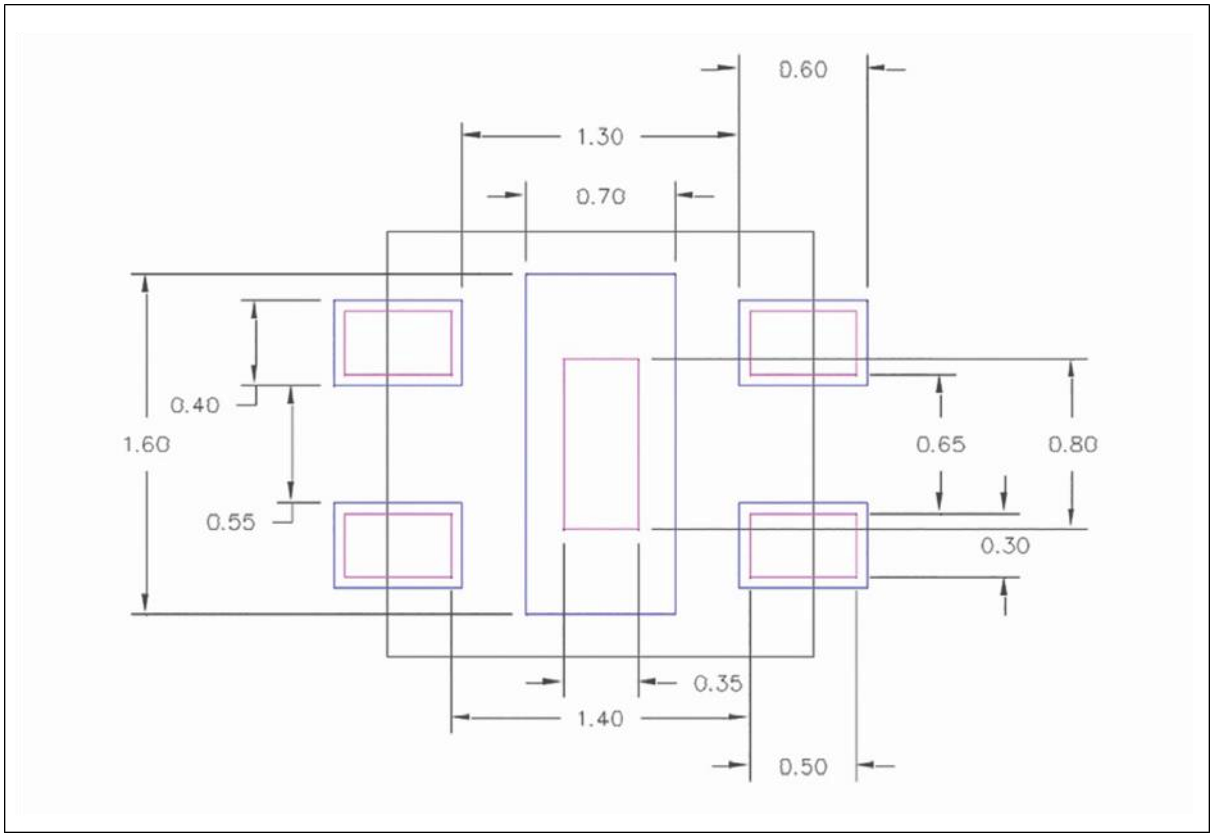


Figure 19: Footprint design (all dimensions are in mm)

26.2 Storage and handling[13]

The ENS212A moisture sensitivity level is 1 (MSL1), which corresponds to an unlimited out-of-bag lifetime at T = 30 °C; RH = 85 %RH maximum. The recommended storage conditions are 10 – 50 °C and 20 – 60 % relative humidity, preferably in the original sealed ESD bag.

Precautions should be taken to prevent electrostatic discharge (ESD) from damaging the sensor product.

All input and output pins are protected against electrostatic discharge (ESD) under normal handling. When handling ensure that the appropriate precautions are taken as described in JESD625-A or equivalent standards.

The pick-up nozzle of the pick and place machine must be positioned in such a way on the component that it fully covers the cavity of the package, to avoid the leakage of air. Because

[13] Also refer to Application Note ENS21x(A) Design Guidelines.

the ENS212A use a cavity package, where the sensor is in direct contact with the environment, physical contact with sensor should be prevented at all times. If required, dust particles can be removed by gently blowing air inside the cavity of the package. Do not brush or wipe.

For proper operation of the product, exposure to Volatile Organic Compounds (VOCs) should be avoided. During manufacturing, transport and storage, VOCs may originate from out-gassing of glues, adhesive tapes and packaging materials such as bags and (pink) foams.

In operation, VOCs might naturally be present in the environment as vapors of, for example, ethanol, acetone and isopropyl alcohol. It is important to realize that some of these contaminants can cause offsets in the sensor reading that may not recover naturally. The same holds for atmospheric pollutants such as ammonia, nitric oxide and chlorine.

Offsets in the sensor reading due to exposure to contaminants may be reversed in certain cases by applying the recommended Reconditioning Procedure (see below Reconditioning). Direct contact with liquid cleaning agents, or rubbing the surface with brushes or cotton-tip sticks, should be avoided at all times.

The sensor is not damaged by water immersion or condensation. The sensor will recover completely when the water evaporates.

Sensor readings are not impacted by high intensity light. This was validated by exposing the sensor to the complete spectrum of a Xe-arc lamp at light levels of 400W/m^2 in the range of 290-800 nm, in accordance with ASTM4459.

26.3 Reconditioning

The procedures indicated below accelerate the reconditioning of the sensor back to its calibrated state.

26.3.1 After soldering

After soldering according to Figure 18, RH reading may show an offset of -2%RH compared to its calibrated value. This offset will slowly disappear if the device is exposed to normal ambient conditions (e.g. T = 25 ºC, RH = 45 %RH, for a week). To accelerate return to its initial calibrated state, we recommend exposing devices to 55 ºC and 50 %RH for 24 hours. This will reduce the time it has to recover at normal ambient conditions before usage.

26.3.2 After extreme conditions

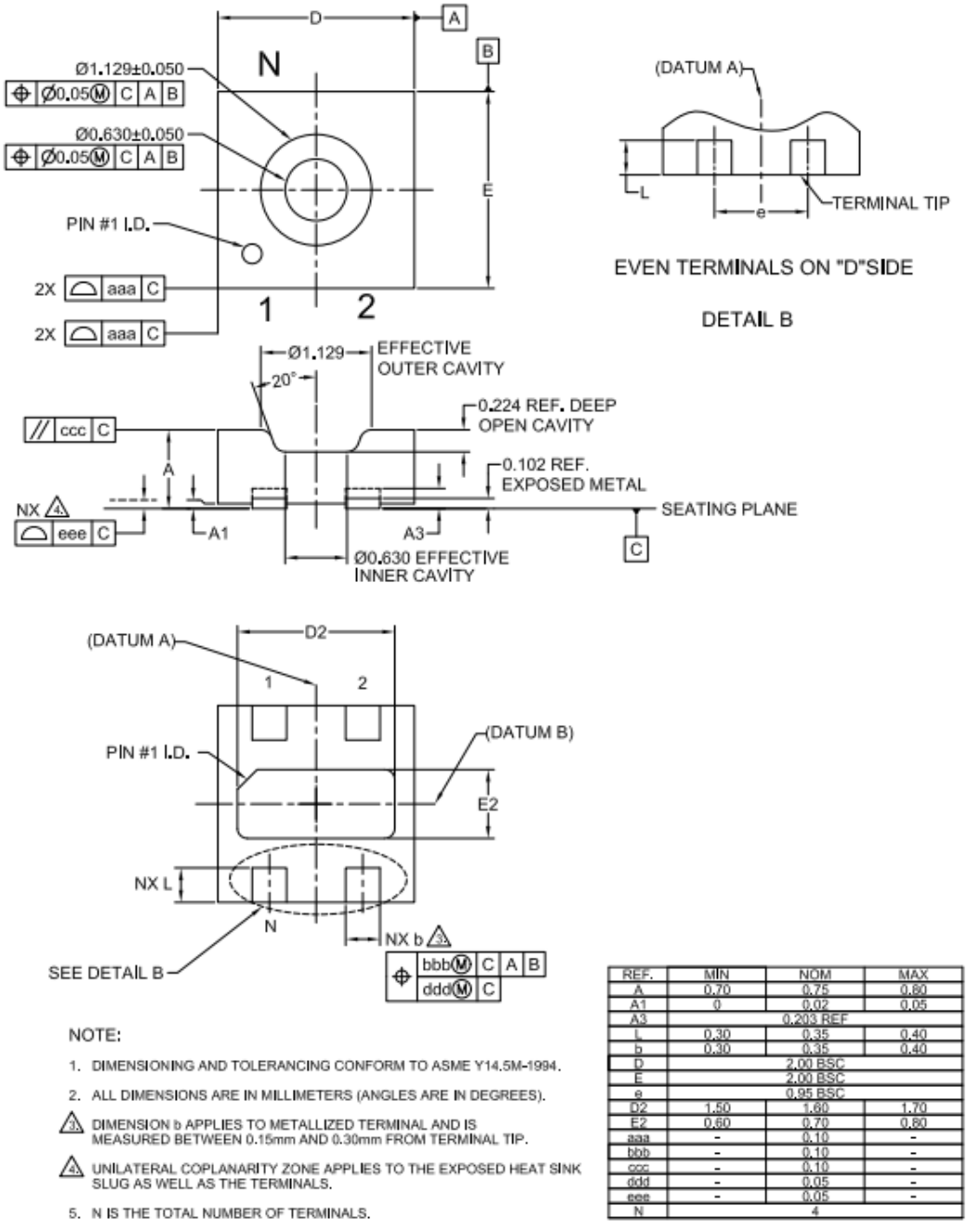
If the device is exposed to conditions outside the ‘specified safe operating range’ for a long time (e.g. 96 h at 85 °C/85 %RH), RH reading will show a temporary offset compared to its calibrated value.

The following procedure accelerates the reconditioning of the sensor back to its calibrated state:

- A mild baking step at 105 °C for 12 hours, to evaporate the contaminant.
- A hydration step at 55 °C and 50 %RH for 24 hours, to rehydrate the sensor material.
- A soak step at normal ambient conditions (e.g. 23 °C ±3 °C, 35-55 %RH) for 24 hours.

27 Package drawings

The ENS212A has a QFN4 package: plastic thermal enhanced very thin small outline package; no leads; 4 terminals; body 2.0 x 2.0 x 0.75 mm, see Figure 20.



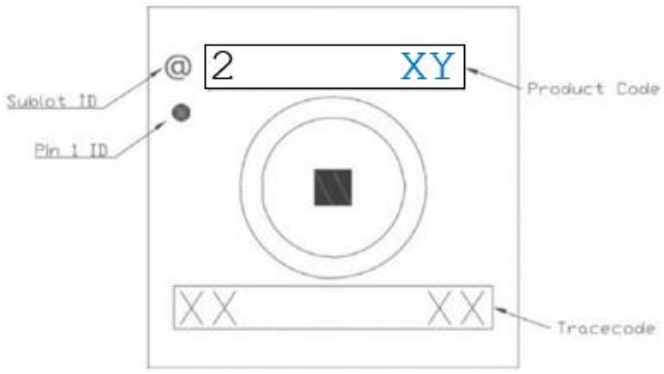
NOTE:

1. DIMENSIONING AND TOLERANCING CONFORM TO ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS (ANGLES ARE IN DEGREES).
3. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM TERMINAL TIP.
4. UNILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.
5. N IS THE TOTAL NUMBER OF TERMINALS.

REF.	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
A3		0.203 REF	
L	0.30	0.35	0.40
b	0.30	0.35	0.40
D		2.00 BSC	
E		2.00 BSC	
e		0.95 BSC	
D2	1.50	1.60	1.70
E2	0.60	0.70	0.80
aaa	-	0.10	-
bbb	-	0.10	-
ccc	-	0.10	-
ddd	-	0.05	-
eee	-	0.05	-
N		4	

Figure 20: Package outline

28 Marking information



Product Variant	Top Marking – 1st line
ENS212(A)	@2 1B **or** @2 12

Figure 21: ENS212A marking

29 Ordering information

Table 21: Ordering information

Device	Ordering Code	Material ID	Package	Description	Delivery Quantity
ENS212A	ENS212A-LQFM	503700500	QFN4	7” Tape & Reel in dry pack	3,500 pcs
Evaluation Kit	ENS21X_EK V1 USB	503700031	Box	ENS21x Evaluation Kit, containing set of current ENS21x flavors	1pc

30 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.

31 Copyrights & Disclaimer

Copyright **Sciosense B.V High Tech Campus 10, 5656 AE Eindhoven, The Netherlands**. Trademarks Registered. All rights reserved. The material herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner.

Devices sold by Sciosense B.V. are covered by the warranty and patent indemnification provisions appearing in its General Terms of Trade. Sciosense B.V. makes no warranty, express, statutory, implied, or by description regarding the information set forth herein. Sciosense B.V. reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with Sciosense B.V. for current information. This product is intended for use in commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by Sciosense B.V. for each application. This product is provided by Sciosense B.V. “AS IS” and any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

This product is not intended for use as safety device and/or in applications where failure of the product could result in personal injury. Sciosense B.V. shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of Sciosense B.V. rendering of technical or other services.

32 Document status

Table 22: 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.

33 Revision information

Table 23: Revision history

Revision	Date	Comment	Page
1.0	2025-10-10	Production Version	All

Notes:

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.

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

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