ENS21x

Using a humidity/temperature sensor on I²C cable length of 2m and above without hassle

ENS21x Application Note

Revision: 1.0
Release Date: 2024-05-13
Document Status: Production

Content Guide

1 Introduction

Many applications already offer a cost-effective I²C bus, allowing the incorporation of multiple sensors. There's no necessity for additional hardware, such as an expensive ADC feature within the microcontroller, to acquire temperature, humidity, or other readings. Only digital GPIOs are essential, and the widely integrated I²C/TWI interface can be utilized. Furthermore, by sharing the bus, several other sensor devices can be connected.

This article explores the implementation of a humidity and temperature sensor via an unconventional and long cable harness for digital I²C communication. Contrary to common assumptions, reliable communication can be easily established over distances of a few meters.

The application note details a real-life scenario and test of the ENS21x sensor under these conditions.

2 Electical

2.1 Integration

ENS21x is compliant to the I²C standard; it supports standard (100kbit/s) and fast mode (400kbit/s) as per I²C-bus specifications [1].

The integration is straightforward following I²C guidance.

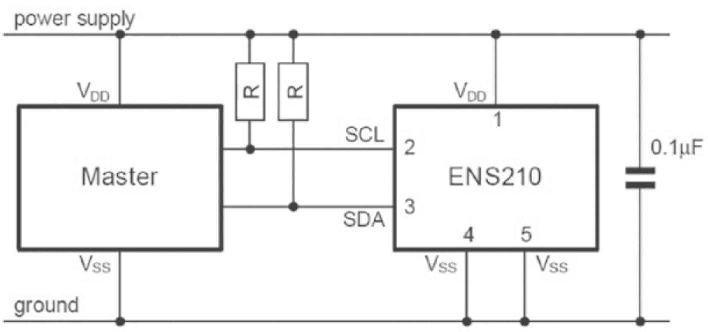
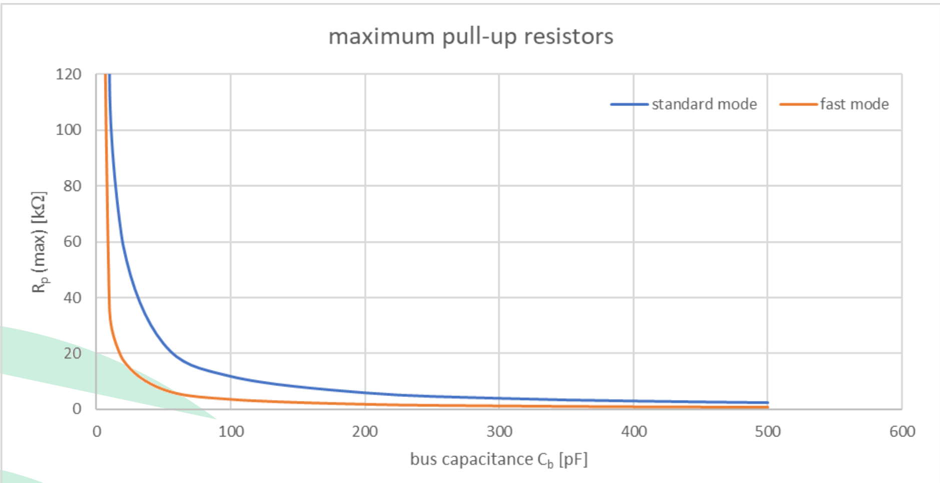


Figure 1: generic schematic

When choosing pull-up resistors, exercise caution. On one hand, they should not be too small to maintain load current control. Conversely, they should not be excessively large, as this could introduce low-pass filtering with a low cut-off frequency to your bus. Referring to Table 11 in the "UM10204 I²C-bus specification and user manual" [1] , for standard-mode communication, the rise time must not exceed 1 ms, while for fast-mode, it should not exceed 300 ns. This information leads to the following equation and the resulting diagram for determining the maximum pull-up resistor values.



$R_{p(\max)} = \frac{t_r}{0.8473 \cdot C_b}$ where

R_p = Pullup-Resistor
t_r = tolerable rise time
C_b = bus capacity

Figure 2: max pull-up resistor

As a rule of thumb Rp=2k..10k should be used for proper function.

2.2 Limitations

2.2.1 Capacitance (time constant) and bit rate

Most critical parameter is the capacitive load of the bus line, as data will be loaded once the clock pulse returns to low

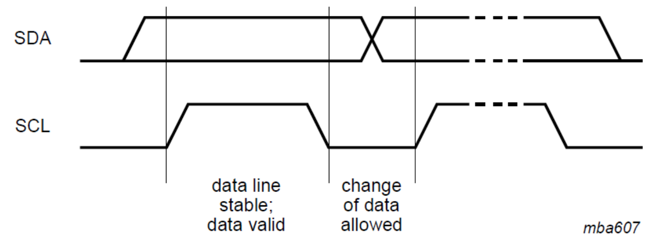


Figure 3: data is loaded after full clock pulse (source [1] fig. 4)

The basic limit for standard mode (100kHz) is 400pF. This is about 8m of 4 way ribbon cable (typically 40-50pF/m).

2.2.2 Noise and RF interference

Unshielded or unprotected cables are vulnerable to external signal interference. To enhance reliability, opt for screened and twisted cables. Specifically, avoid twisting the SDA (data) and SCL (clock) lines together. Instead, pair them with GND (ground) or create a pair where SDA is twisted to VDD (supply voltage) and SCL is connected to GND. For increased robustness, consider operating the bus at higher currents than the standard 1.5 mA by utilizing lower pull-up resistor (R_p) values.

2.2.3 Ground bounce

Since the bus operates in a single-ended configuration, any voltage drop or noise on the ground (GND) and supply voltage (VDD) wires directly impacts the signal threshold. To address this, consider the following recommendations:

Separate Power Wires:

- For extended cable runs or situations where modules consume substantial current, it’s advisable to provide dedicated power wires for the high-current circuits.
- By doing so, you can isolate the power supply for these modules from the I²C bus, minimizing voltage fluctuations and noise.

Utilize Bus Power for I²C Interface Chips:

- Leverage the bus power (VDD) to directly operate the I²C interface chips.
- This approach ensures consistent power delivery to the interface chips, reducing the impact of voltage variations and noise.

Remember that maintaining stable power and minimizing noise is crucial for reliable I²C communication, especially over extended cable distances. Following these guidelines enhances the robustness of your I²C connections.

2.2.4 Clock speed

To extend the I²C bus length, consider reducing the clock speed. Depending on the cable used and its length, it's feasible to lower the clock frequency below 1 kHz for the ENS21x sensor family. In addition to electrical optimizations, software adjustments allow easy modification of the clock speed.

Subsequent chapters will delve into testing these configurations

For further information see also [3]

2.2.5 EMC

The impact of electromagnetic compatibility (EMC) needs to be assessed on a case-by-case basis. While this test describes conditions for a specific cable, EMC testing should be conducted considering the actual wiring used and the unique environmental condition.

2.3 Signal quality

As criteria we define signal quality as:

- Strength refers to the amplitude or power of the signal. A strong signal ensures robust transmission and reception, minimizing the risk of data loss or corruption.
- Clarity pertains to the absence of noise or interference. A clear signal is free from disturbances that could distort the original information being transmitted.
- Reliable signals maintain consistent performance over time. They are resilient to external factors such as environmental conditions, interference, and equipment variations.

3 Test

3.1 Sample test code

Experiments were conducted on a controller powered by an Arduino Nano, utilizing the ATMega328P microcontroller. By adjusting the TWI prescaler and clock configurations, we achieved SCL frequencies ranging from 0.5 kHz to 400 kHz[1]

The clock frequency can be configured by setting the TWBR register and the TWPS1:0 bits within the TWSR register

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{TWPS}}$$

Clock frequency	TWBR value	TWPS value
100kHz	72 / 0x48	0
10kHz	192 / 0xC0	1
5kHz	100 / 0x64	2
1kHz	125 / 0x7D	3

Figure 4: clock frequency calculation and helpful values

To run the tests, go to https://github.com/sciosense/ens21x-arduino , take e.g. 02_Continuous_Mode/ENS212 and modify after the `Wire.begin()` instruction

```
Wire.begin();

// Apply TWI clock speed, uncomment the required clock rate
// -------------
// TWBR = 0x48;    uint8_t TWPS = 0;  //set SCL to 100 kHz
// -------------
// TWBR = 0xC0;    uint8_t TWPS = 1;  //set SCL to 10 kHz
// -------------
TWBR = 0x64;    uint8_t TWPS = 2;  //set SCL to 5 kHz
// -------------
// TWBR = 0x7D;    uint8_t TWPS = 3;  //set SCL to 1 kHz
// -------------
TWSR = (TWSR & ~0x03) | TWPS;          // set TWPS Bits
```

Figure 5: screenshot for code modification to change clock speed

This easily allows to demonstrate TWI operation at different speed.

[1] Eventually use a level shifter to 1.8-3.3V for SCL/SDA in order not to exceed max V_{DD}

3.2 Test setup

For the test the following setup has been used:

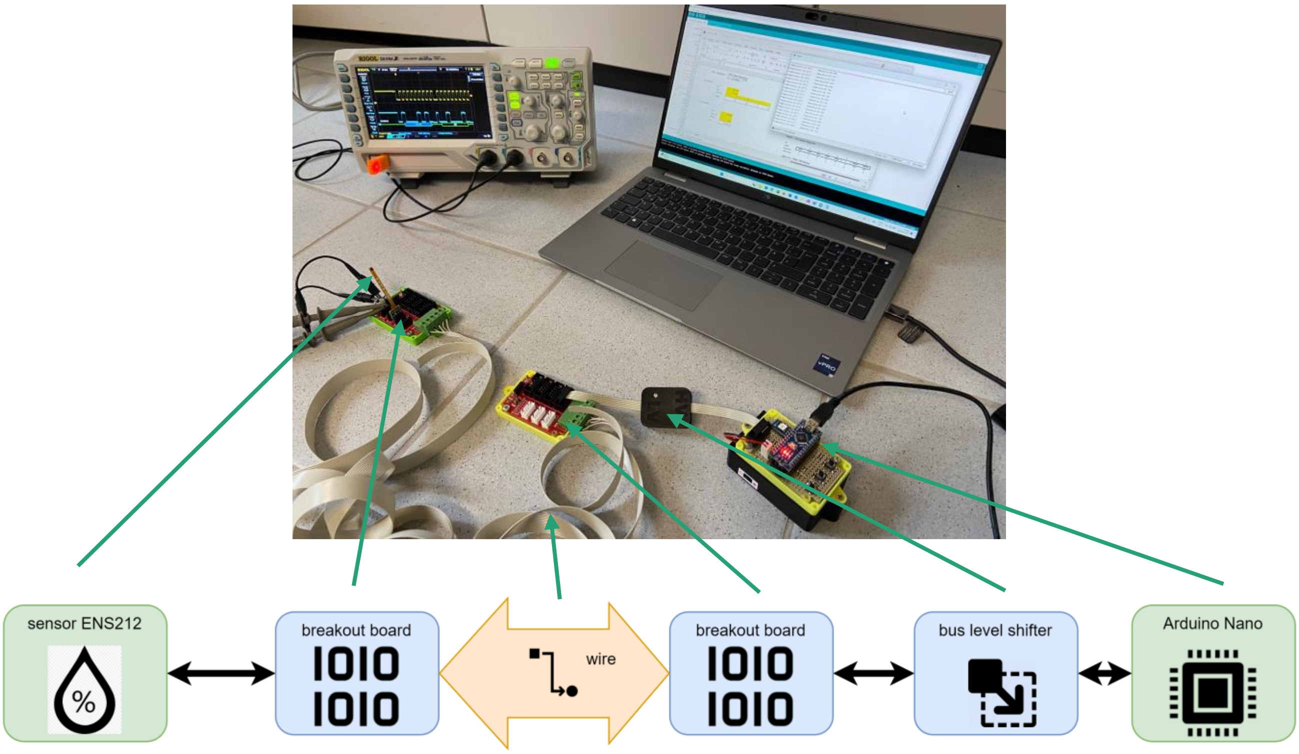


Figure 6:test setup

Used modules:

- Controller: An Arduino Nano was utilized as the controller board.
- Level Shifter: A Level Shifter was employed to adjust the I²C Bus voltage level to 3.3V.
- Breakout Boards: Two breakout boards were used, the second one hosted the sensor ENS211.

Measurement Setup:

- A standard digital oscilloscope was used.
- The oscilloscope probed the SCL (Serial Clock) and SDA (Serial Data) lines right next to the sensor.
- During the test, readings for temperature and relative humidity were observed.
- Serial data from the microcontroller was logged simultaneously.

Test Conditions:

- Standard 26AWG ribbon cable was employed.
- The cable length was approximately 3.5 meters.

Procedure:

- The measurements were taken at clock speeds of 100 kHz and 1 kHz.

4 Result

During bus communication the communication those signals have been recorded

Figure 7: signal waveform at clock speed 100kHz and zoom in

Figure 8: signal waveform at clock speed 1kHz and zoom in

Higher Frequency and Cable Capacity:

- As the frequency increases, the cable's capacity has a more pronounced impact on signal quality and rise/fall time.
- Even at a clock speed of 100 kHz, the I²C bus maintains a high-quality signal form.
- The rise time and overshoot remain within acceptable tolerance levels.
- Readings can be safely and repeatably performed.

Safety Margin and Robust Communication:

- The rise time is comfortably distant from the limit, providing an adequate safety margin.
- For more robust data communication over longer cable lengths, consider reducing the clock speed.
- By adjusting the corresponding registers, you can decrease the clock speed—for example, down to 1 kHz.

5 Conclusion

The performed test leads to the following conclusions

Repeatable Data Readings:

- The tests consistently yielded repeatable data readings.
- Reliable communication was achieved over a simple and cost-effective wire.

Application Implications:

- Developers can confidently utilize high-performing digital sensors (such as the ENS21x) in their applications.
- To ensure reliable communication, pay attention to the bus clock speed, which should correspond to the characteristics of the cable harness used.

In essence, thoughtful consideration of clock speed and cable quality enables successful integration of these sensors into various applications.

6 Ressources

- [1] UM10204 I2C-bus specification and user manual
  https://www.nxp.com/docs/en/user-guide/UM10204.pdf
- [2] NXP AN10216-01 I2C MANUAL
  https://www.nxp.com/docs/en/application-note/AN10216.pdf
- [3] AN11084 Very large I2C-bus systems and long buses
  https://www.nxp.com/docs/en/application-note/AN11084.pdf
- [4] Datasheet ScioSense ENS21x
  ENS21x family of high-performance digital temperature and humidity sensors – ScioSense

7 Copyrights & Disclaimer

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.

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.[2][2]

8 Revision information

Table 1: Revision history

Revision	Date	Comment	Page
1.0	20.06.2024	First edition	All

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

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