

I²C DOLPHIN INTERFACE

I²C DOLPHIN INTERFACE CONNECTING SENSORS THROUGH THE I²C BUS

11 October 2026



Observe precautions! Electrostatic sensitive devices!

Patent protected:

WO98/36395, DE 100 25 561, DE 101 50 128,
WO 2004/051591, DE 103 01 678 A1, DE 10309334,
WO 04/109236, WO 05/096482, WO 02/095707,
US 6,747,573, US 7,019,241

I²C DOLPHIN INTERFACE

REVISION HISTORY

The following major modifications and improvements have been made to this document:

Version	Author	Reviewer	Date	Major Changes
1.0	MH		2010.10	Original version
2.0	AZ	JB, SA	2026.10	Document maintenance

Published by EnOcean GmbH, St.-Martin-Straße 76, 81541 Munich, Germany
www.enocean.com, info@enocean.com, phone +49 (89) 6734 689-0

© EnOcean GmbH, All Rights Reserved

Important!

This information describes the type of component and shall not be considered as assured characteristics. No responsibility is assumed for possible omissions or inaccuracies. Circuitry and specifications are subject to change without notice. For the latest product specifications, refer to the EnOcean website: <http://www.enocean.com>.

As far as patents or other rights of third parties are concerned, liability is only assumed for modules, not for the described applications, processes and circuits.

EnOcean does not assume responsibility for use of modules described and limits its liability to the replacement of modules determined to be defective due to workmanship. Devices or systems containing RF components must meet the essential requirements of the local legal authorities.

The modules must not be used in any relation with equipment that supports, directly or indirectly, human health or life or with applications that can result in danger for people, animals or real value.

Components of the modules are considered and should be disposed of as hazardous waste.

Local government regulations are to be observed.

Packing: Please use the recycling operators known to you.

I²C DOLPHIN INTERFACE

TABLE OF CONTENTS

1 Introduction..... 4

1.1 References 4

1.2 Content of the SHT21 and STM 300 Package..... 5

1.3 The I²C Bus 5

2 Configuring the I²C Interface on Dolphin 5

2.1 SCL Pin 6

2.2 SDA Pin..... 6

3 Operating the I²C Interface on Dolphin..... 6

4 Application and Sample Source Code 7

4.1 Radio Settings 8

4.2 Ultra-Low Power and Line-Powered Operation 8

5 Getting Started Step by Step 9

LIST OF FIGURES

Figure 1 Overview of the application architecture 4

Figure 2 Sensor activity diagram 7

Figure 3 Measurement activity diagram..... 8

Figure 4 Linking the radio settings file in the Keil project workspace 9

LIST OF TABLES

Table 1 Pins of the GPIO_0 port 6

I²C DOLPHIN INTERFACE

1 Introduction

This application note shows how an EnOcean Dolphin module is connected to a sensor through the I²C bus. The example uses an STM 300 module [9] as the bus master and a Sensirion SHT21 temperature and humidity sensor [7] as the bus slave.

The application note is the result of a joint project between EnOcean and Sensirion. It describes how the I²C bus is implemented and configured on the Dolphin chip. The corresponding application note from Sensirion [8] describes the reference design for the connection of the SHT21 sensor to an STM 300 module. Figure 1 shows the structure of the application.

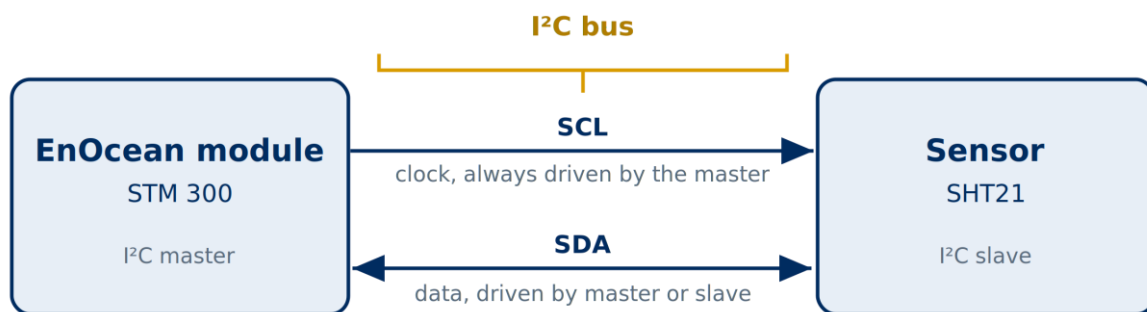


Figure 1 Overview of the application architecture

The sample source code is written for the STM 300. The routines that operate the I²C bus can be ported to any other Dolphin based product. The sample source code also contains functions that are specific to the SHT21 sensor, for example sensor configuration and value encoding. These functions are kept in separate files and can be excluded from the project. The project can therefore be used as a template for other applications that combine an EnOcean module with a sensor that has an I²C interface.

1.1 References

- [1] EnOcean: Dolphin Core Description
https://www.enocean.com/.../dolphin_core_description_v1.1.pdf
- [2] EnOcean Alliance: EnOcean Equipment Profiles (EEP), technical specifications,
www.enocean-alliance.org/specifications
- [3] EnOcean: Software, tools and kits, download area for DolphinAPI and DolphinStudio,
www.enocean.com/en/support/software-tools
- [4] EnOcean: Support, support@enocean.com
- [5] NXP Semiconductors: I²C-bus specification and user manual, UM10204,
www.nxp.com/docs/en/user-guide/UM10204.pdf
- [6] Keil: C51 development tools for the 8051 architecture,
www.keil.com/products/c51/c51.asp
- [7] Sensirion: SHT21 product page and data sheet,
sensirion.com/products/catalog/SHT21

I²C DOLPHIN INTERFACE

[8] Sensirion: Application note "SHT21 interfacing with EnOcean STM300", [sensirion.com/resource/application_note/sht/sht21_stm300](https://www.sensirion.com/resource/application_note/sht/sht21_stm300)

[9] EnOcean: STM 300 product page, www.enocean.com/en/product/stm-300

1.2 Content of the SHT21 and STM 300 Package

The package for the combination of the SHT21 sensor and the STM 300 module contains the following items:

- this application note
- the application note from Sensirion [8]
- the sample source code
- the energy budget calculation for the STM 300 with an SHT2x sensor
- the data of the extension board for the SHT21 and the STM 300: bill of materials, schematic, assembly drawing and Gerber files

1.3 The I²C Bus

The I²C bus is a simple bidirectional two-wire bus for communications between integrated circuits. It was developed by Philips Semiconductors, today NXP Semiconductors. Only two bus lines are required:

- serial data line (SDA)
- serial clock line (SCL)

The clock is generated by the master, which is the STM 300 in this example. The communication is serial, byte oriented and bidirectional. The SDA line is driven either by the master or by one of the slaves. Slaves are addressed with 7 or 10 bits. The following data rates are defined:

- up to 100 kbit/s in Standard-mode
- up to 400 kbit/s in Fast-mode
- up to 1 Mbit/s in Fast-mode Plus
- up to 3.4 Mbit/s in High-speed mode

The complete specification of the I²C bus is published by NXP [5].

On the Dolphin chip the I²C interface is emulated in software on GPIO pins [1]. Conformity of this emulation with the electrical specification of the I²C bus [5] has not been confirmed by measurement. In this example the STM 300 is the only master on the bus. Routines for a configuration with several masters are not implemented.

2 Configuring the I²C Interface on Dolphin

Two pins are required for I²C communications: one pin for the serial data line (SDA) and one pin for the serial clock line (SCL). Any two pins of the GPIO_0 port [1] can be used. Table 1 lists the four pins of this port together with the assignment that is used in the sample source code.

I²C DOLPHIN INTERFACE

Table 1 Pins of the GPIO_0 port

Pin	Use in the sample source code
SCSEDIO0	serial data line (SDA)
SCLKDIO1	serial clock line (SCL)
WSDADIO2	not used, available as digital input
RSDADIO3	not used, available as digital input

The pins of the GPIO_0 port are also used by the SPI interface. Only one of the two interfaces, SPI or I²C, can be used at a time. If two pins are used for I²C communications, the two remaining pins of the port can be used as digital inputs only, not as digital outputs.

2.1 SCL Pin

The SCL pin is controlled by the master. All clock pulses are generated by the STM 300. A slave may stretch the clock, which means that it holds the clock line low, to signal that it is busy [5]. This function is not implemented in the sample source code. Instead, a fixed waiting time is observed after a measurement has been requested, before the result is read from the sensor [7].

In the sample source code, the clock rate reaches up to 130 kbit/s, which corresponds to Fast mode. The clock rate is reduced by adding empty `nop()` calls to the routine. Chapter 3 describes the routines.

2.2 SDA Pin

The SDA pin changes its direction whenever the direction of the data transfer between the STM 300 and the SHT21 changes. The direction is changed in the direction register of the GPIO_0 port.

Both devices sample the data while the clock is high. One clock pulse is generated for every bit. The level of the SDA line must therefore be stable during the high phase of the clock. The data on the SDA line may only change while the clock is low.

3 Operating the I²C Interface on Dolphin

This application note does not describe the I²C specification itself. Reference [5] contains the complete description of the bus.

The following basic routines are required on the Dolphin chip:

- set SCL high or low
- set SDA high or low
- set SDA as input or as output
- write or read the value of SDA

These routines are sufficient to operate the I²C bus on Dolphin. They are contained in the files `i2c.c` and `i2c.h`, which can be copied into other projects.

I²C DOLPHIN INTERFACE

In addition to the basic routines, functions for sending and receiving one byte including the acknowledge bit are implemented. These functions are contained in the same files. The sample source code contains detailed documentation of all functions.

4 Application and Sample Source Code

The function of the application is simple. Figure 2 shows the complete sequence. The STM 300 wakes up every 20 seconds. This wake-up period is set once after power-on or after a reset. After every wake-up the module switches the supply of the SHT21 sensor on and requests the temperature and the humidity value. While the sensor starts up and performs the measurement, the STM 300 returns to sleep. The length of these sleep periods is defined in the specification of the SHT21 [7]. Figure 3 shows the measurement sequence.

The measured values are then scaled according to the EnOcean Equipment Profiles (EEP) [2] and transmitted in a radio data telegram.

When the learn button on the WAKE_1 pin is pressed, a learn telegram is transmitted. The learn telegram contains the EEP number A5-04-01 (temperature and humidity sensor) and the manufacturer ID 0x7FF (multi-user ID). The EEP tells the receiver how the sensor data that follows must be interpreted.

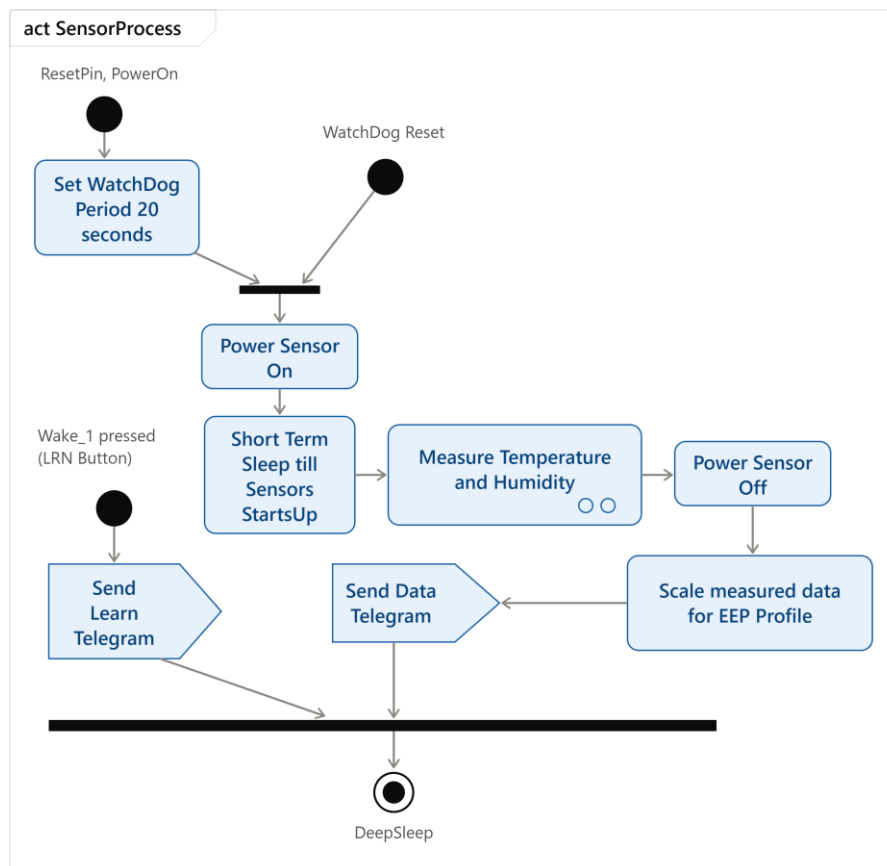


Figure 2 Sensor activity diagram

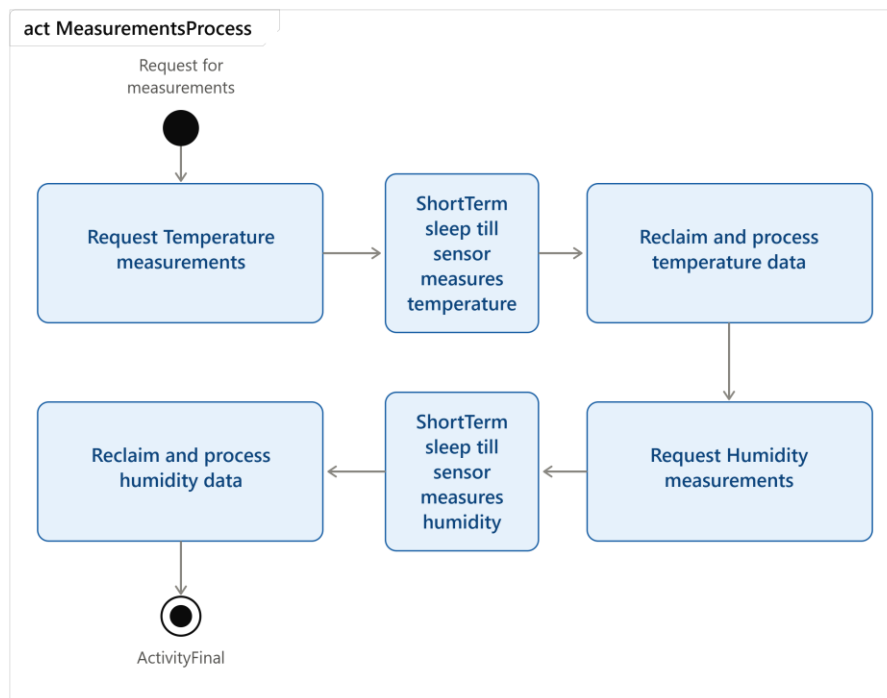
I²C DOLPHIN INTERFACE

Figure 3 Measurement activity diagram

4.1 Radio Settings

The sample source code runs on every Dolphin based EnOcean module. A radio settings assembler file must be selected for the module that is used. This file is specific to:

- the module type, for example STM
- the module step code, for example DA
- the module frequency, for example 868.3 MHz

The radio settings for the STM module, 868.3 MHz, step code DA are linked by default.

4.2 Ultra-Low Power and Line-Powered Operation

The example is designed for energy autarkic, ultra-low power applications, which are typically sensors that are supplied by an energy harvester. The structure of the sample source code project is different for line-powered applications and for ultra-low power applications. The DolphinAPI manual [3] describes both structures.

The I²C routines can also be used in line-powered applications. The restrictions described in chapter 2 apply in this case as well.

I²C DOLPHIN INTERFACE

5 Getting Started Step by Step

This chapter describes how the sample source code project is used. It explains how the firmware is installed, configured, and downloaded to the module.

1. Install the latest version of DolphinAPI and DolphinStudio. Both are available in the EnOcean software download area [3].
2. Obtain a Keil C51 compiler and a licence. The compiler is required to change and compile the sample source code. It is a third-party tool and must be ordered directly from Keil [6].
3. Copy the sample source code project into the examples folder of the DolphinAPI installation, for example C:\Program Files\EnOcean\DolphinAPI\EO3000I_API\Examples. The DolphinAPI files are already linked to the project. The project must be stored in the examples folder so that the default file paths are valid.
4. Open the Keil project file (*.uvproj).
5. Select the radio settings assembler file for the module that is used. The radio settings files are stored in the DolphinAPI installation folder, for example C:\Program Files\EnOcean\DolphinAPI\EO3000I_API\EO3000I_LIB\RadioCFG.
6. Link the radio settings file to the project workspace in Keil, as shown in Figure 4.

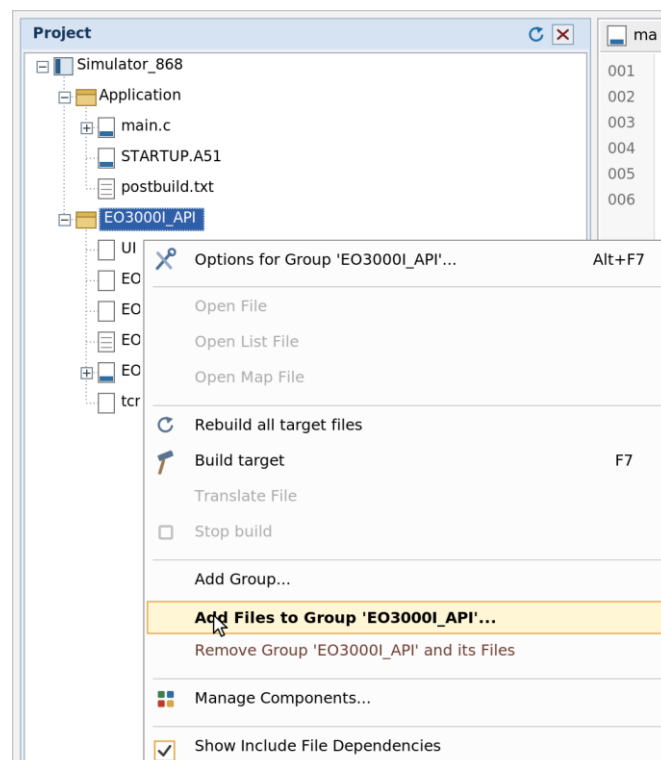


Figure 4 Linking the radio settings file in the Keil project workspace

Note: Only one radio settings file may be linked to the project. Exclude the file that is currently linked before a new file is added.

I²C DOLPHIN INTERFACE

7. Define the pins that are used for SDA and SCL. The default assignment is listed in Table 1.
8. Change the communication pins if this is required. The pins are defined in the file i2c.c:

```
#define sda_pin    SCSEDIO0  
#define slc_pin    SCLKDIO1
```

Replace SCSEDIO0 and SCLKDIO1 with the pins that have been selected and observe the restrictions described in chapter 2.

9. Connect the module to the PC with an EnOcean programmer. An evaluation board is normally used for this connection.
10. Compile the firmware and download it to the module.
11. Prepare the SHT21 sensor and the extension board as described in the application note from Sensirion [8].

The EnOcean support team [4] provides further assistance.

I²C DOLPHIN INTERFACE

DISCLAIMER

The information provided in this document describes typical features of the EnOcean radio transmission system and should not be misunderstood as specified operating characteristics. No liability is assumed for errors and / or omissions. We reserve the right to make changes without prior notice. For the latest documentation visit the EnOcean website at **www.enocean.com**.