

I²C DOLPHIN INTERFACE - Connecting Dolphin based modules to sensorics through I²C



I²C DOLPHIN INTERFACE

Table of content

1. INTRODUCTION	3
1.1. SHT 21 & STM 300 PACKAGE	3
1.2. I ² C BUS	4
1.3. REFERENCES.....	4
2. CONFIGURE I²C INTERFACE ON DOLPHIN	5
2.1. SCL PIN	5
2.2. SDA PIN.....	5
3. OPERATE I²C INTERFACE ON DOLPHIN	5
4. APPLICATION – SAMPLE SOURCE CODE	6
4.1. RADIO SETTINGS	7
4.2. ULTRA LOW POWER VS. LINE POWERED	7
5. GETTING STARTED – STEP BY STEP	7

I²C DOLPHIN INTERFACE

1. INTRODUCTION

In this application note we want to present a demonstration of how to connect an EnOcean Dolphin chip to a sensor through the I²C bus. In the demonstration we use a SHT 21 temperature / humidity sensor from Sensirion [7] and an EnOcean STM 300 Module [9].

This application note is the result of a joint project between EnOcean and Sensirion. This part describes how to implement and configure the I²C bus on Dolphin Chip. Please see the application note from Sensirion [8] (which is part of this package) to see the reference design specifics of connecting the SHT 21 humidity and temperature sensor operating with I²C interface to an EnOcean Module STM 300. For visualisation please see Figure 1.

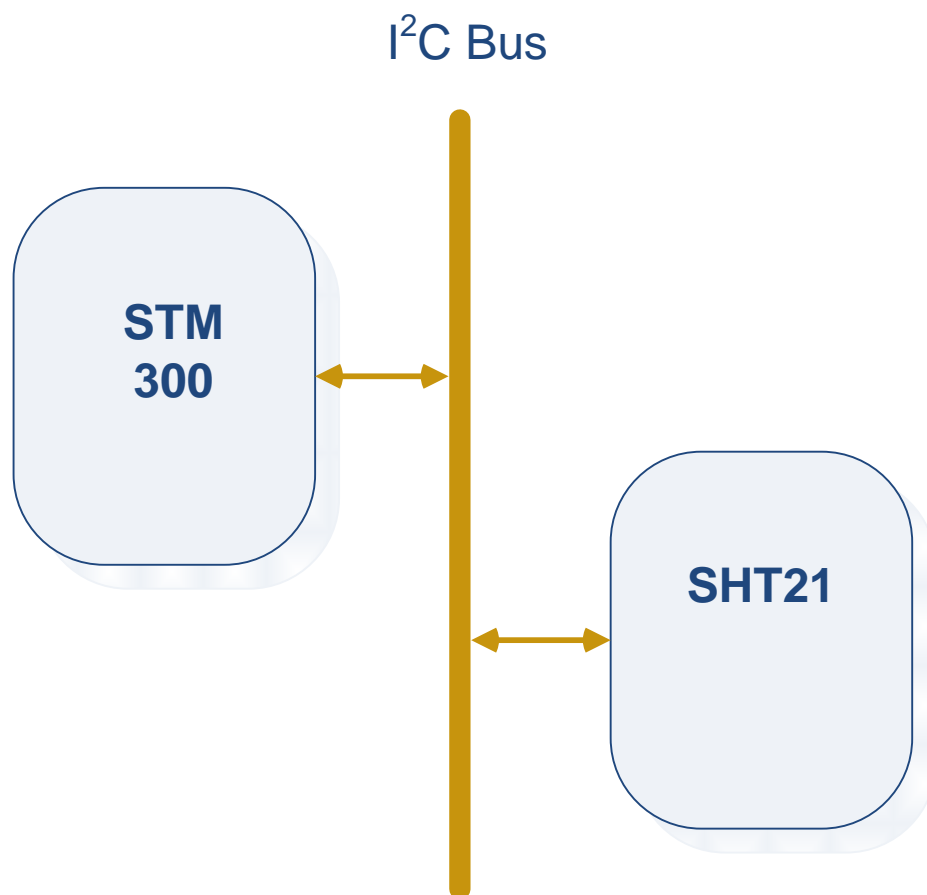


Figure 1 Overview of Application Architecture

Sample source code which is included in SHT 21 & STM 300 package [8] is intended for STM 300, but the routines operating the I²C bus can be easily ported to any other Dolphin based product. The sample source code also includes specific calls for the SHT 21 Sensor (configuration, value encoding). The architecture of the project enables to exclude files with these calls, so the project can be used as template for other applications with an EnOcean module and sensor operating with I²C interface.

1.1. SHT 21 & STM 300 Package

The SHT 21 & STM 300 package [8] includes:

- Application Note from EnOcean (AN508: "I²C Dolphin interface")

I²C DOLPHIN INTERFACE

- Application Note from Sensirion (AN: "SHT21 interfacing with EnOcean STM300")
- Sample Source Code
- Energy budget calculations – STM 300 with STH2x
- SHT 21 & STM 300 Extension Board Data (BoM, Schematic, Assembly, Gerber files)"

1.2. I²C Bus

Philips Semiconductors (now NXP Semiconductors) developed a simple bidirectional 2-wire bus for efficient inter-IC control. This bus is called the Inter-IC or I²C-bus. Only two bus lines are required:

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

The clock is provided by the master (in our demonstration the STM 300). The communication is serial, 8-bit oriented, bidirectional. The SDA line is either occupied by the master or one of the slaves (7-bit or 10 bit addressing of slaves). Data transfers can be made at:

- serial clock line (SCL)
- up to 100 kbit/s in the Standard-mode
- up to 400 kbit/s in the Fast-mode
- up to 1 Mbit/s in the Fast-mode Plus (Fm+)
- up to 3.4 Mbit/s in the High-speed mode

Please find a full specification of the I²C-bus on NXP Homepage [5].

Operating the I²C interface from Dolphin is emulated by software on GPIO pins [1]. It is not guaranteed (not confirmed by measurements) that this emulation is conform to the I²C physical specifications [5]. In the demonstration the STM 300 is a single master, routines to handle multi-master architecture are not implemented.

1.3. References

- [1] Dolphin Core Description - (<http://www.enocean.com/knowledge-base/>)
- [2] EnOcean Equipment Profiles 2.1 Specification - (<http://www.enocean.com/knowledge-base/>)
- [3] EnOcean Software Download - (<http://www.enocean.com/en/download/>)
- [4] EnOcean Support - (<http://www.enocean.com/support/>)
(support@enocean.com)
- [5] I²C Specification - (<http://www.standardics.nxp.com/support/i2c/>)
- [6] Keil Licensing options - (<http://www.enocean.com/en/knowledge-baseasicapi%3Aissue%3Akeillicensordering/>)
- [7] SHT 21 Product page - (www.sensirion.com/sht21)
- [8] SHT 21 – STM 300 Package - (<http://www.enocean.com/sht21-stm300/>)

I²C DOLPHIN INTERFACE

[9] STM 300 Product page –
(http://www.enocean.com/en/enocean_modules/stm-300/)

2. CONFIGURE I²C INTERFACE ON DOLPHIN

As described before, for the I²C communication 2 PINs are required:

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

It is allowed to use on Dolphin two pins from the GPIO_0 port [1] for the I²C communication. The user can define which of the four pins will be used. GPIO_0 has these four pins:

- SCSEDIO0
- SCLKDIO1
- WSDADIO2
- RSDADIO3

These pins are also used for the SPI interface. You can use only SPI or I²C at the same time. If using two pins for I²C communication the other two pins from GPIO_0 port can be used only as digital inputs and not as digital outputs.

2.1. SCL Pin

The SCL Pin is controlled by the master (STM 300). All clock pulses come from master. Slave can stretch the clock (put and hold it down) [5] to signalize he is busy. This feature is not implemented in the sample source code. After requesting measurements from slave there is a predefined wait period [7] before requesting the actual data.

In the sample source code the SCL pin can clock up to 130 kbit/s (Fast Mode). To reduce the rate you can add *nop()* – empty calls to the routine. You can find more information in chapter 3.

2.2. SDA Pin

The SDA pin changes direction (OUT/IN) as the direction of the communication changes (STM 300 -> SHT 21 / SHT 21->STM 300). To change the direction it is needed to change the GPIO_0 direction register.

Data value is sampled / read on both devices during the HIGH period of the clock. For every bit one clock pulse is generated. Therefore the value of the SDA line must be stable during the HIGH period of the clock. Only when the SCL is LOW the data on the SDA line can be changed.

3. OPERATE I²C INTERFACE ON DOLPHIN

In this document we do not focus on the I²C specifications, please see this reference [5] if you are not familiar with I²C interface.

Resulting from the specification we need in the Dolphin these basic routines:

- Set SCL high / low
- Set SDA high / low
- Set SDA as input / output
- Write / Read SDA value

These routines enable to operate I²C on Dolphin. You can find the I²C routines in **i2c.c** and **i2c.h**. These files can be also copied to other projects to enable I²C communication.

I²C DOLPHIN INTERFACE

Besides the basic routines also functions to receive and send a byte (8-bit with ACK) are implemented. They can be also found the I²C files. Please refer to the sample source code for detailed documentation.

4. APPLICATION – SAMPLE SOURCE CODE

The application has simple functionality. Please refer to the Figure 2. It is set that the STM 300 wakes up every 20 seconds. This period is set one time only after powering up or reset. Then it powers the SHT 21 sensor and requests humidity and temperature values. While the sensor is starting up or performing measurements the STM 300 goes to sleep until the sensor values are ready. The duration of these sleep periods is defined by the SHT 21 specification [7]. Please refer to Figure 3.

The measured values are then scaled according to the EnOcean Equipment Profiles 2.1 (EEP) standard [2]. Then they are transmitted via a radio data telegram.

After a reset from wake button (Learn Button) a radio learn telegram that includes the EEP number (A5-04-01 - Temperature and Humidity Sensor) and manufacturer ID (0x7FF - Multiuser ID) is transmitted. The EEP tells the receiver how to interpret the future sensor data.

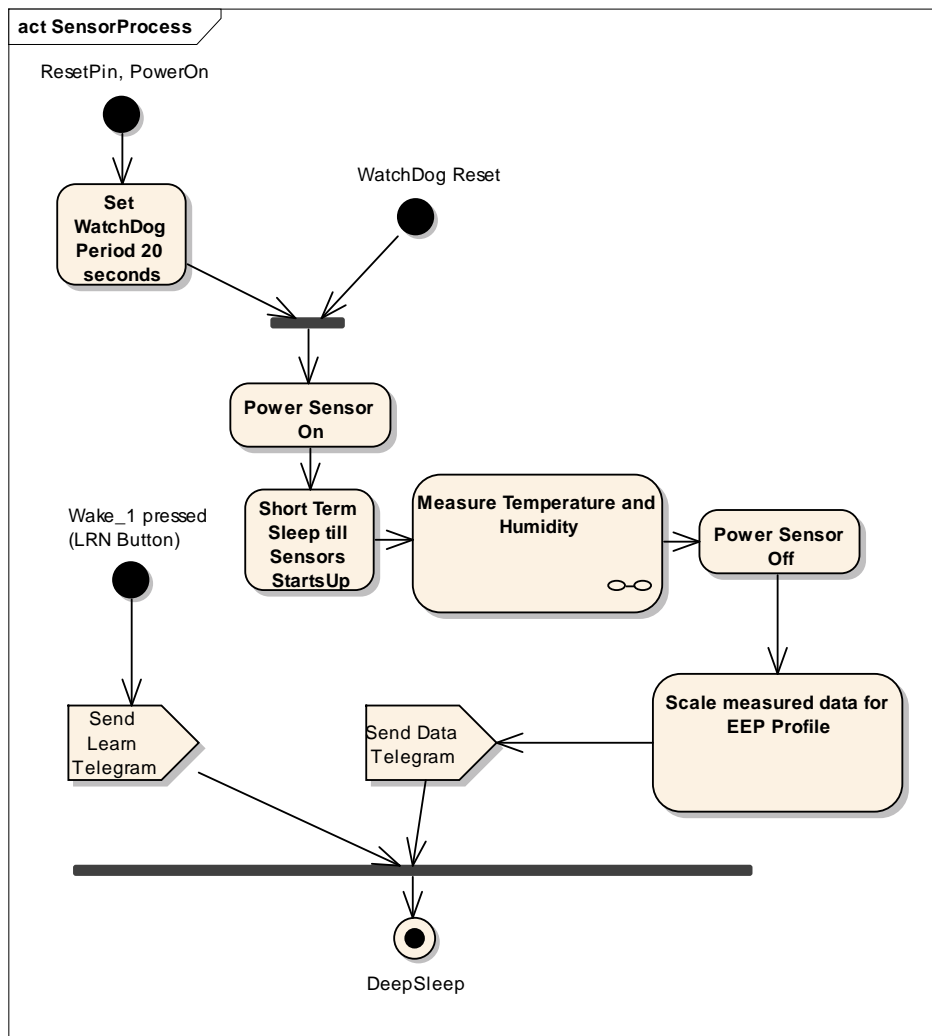


Figure 2 Sensor activity diagram

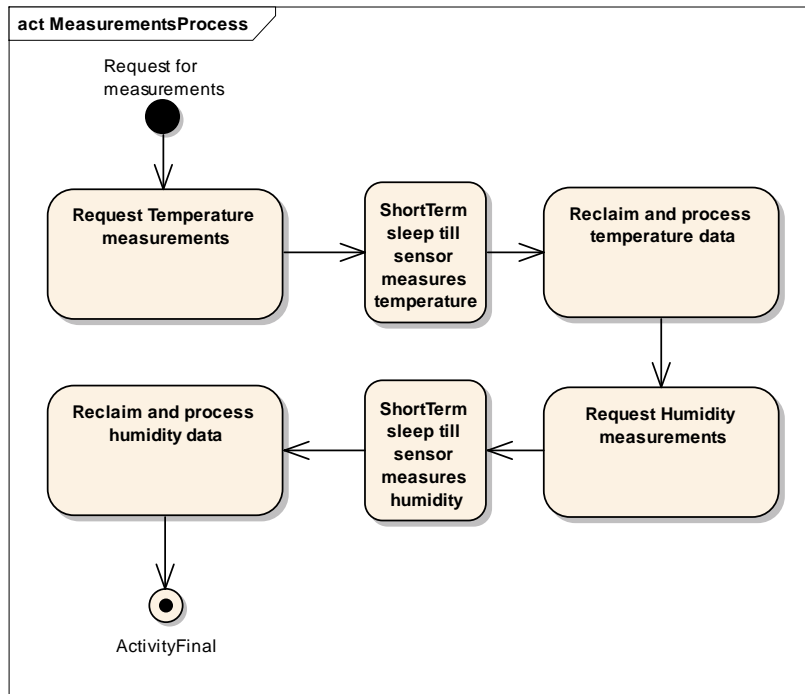
I²C DOLPHIN INTERFACE

Figure 3 Measurements activity diagram

4.1. Radio Settings

The sample source code can run on every Dolphin based EnOcean module. To port it to different modules you have to specify a radio settings assembler file. The radio settings file is specific for:

- Module type (e.g. STM)
- Module step code (e.g. DA)
- Module frequency (e.g. 868.3 MHz)

Please choose a radio settings file according to the module you want to use. As default file the radio settings for STM, 868.3 MHz, Step Code: DA are linked.

4.2. Ultra Low Power vs. Line Powered

Application example is designed for energy autarkic – ultra low power applications. Such applications are mostly sensors with an energy harvester. Sample source code project architecture changes by line powered and ultra low power application. Please refer to the DolphinAPI manual for more details [3].

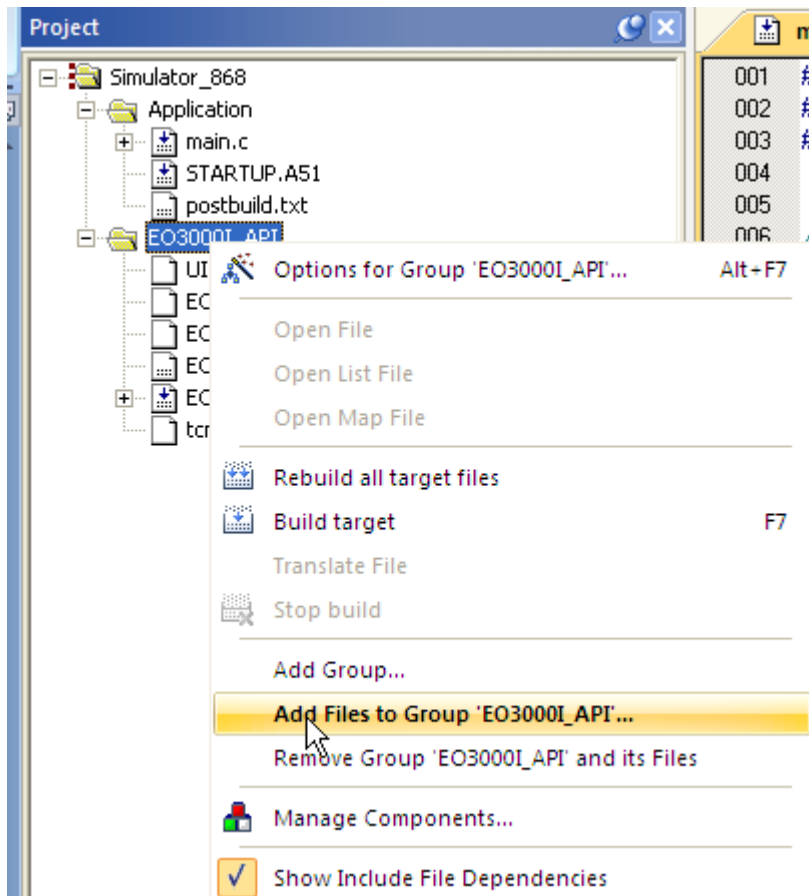
The I²C routines are available also in Line Powered applications. But please consider restrictions mentioned in chapter 2.

5. GETTING STARTED – STEP BY STEP

In this chapter we describe how to use the attached sample source code project. We explain how to install, configure and download the firmware to the module. Please follow these steps:

I²C DOLPHIN INTERFACE

1. Before we can start please install the newest DolphinAPI and DolphinStudio. You can download it from the EnOcean software download centre [3].
2. To change and compile the sample source code you need a KEIL C51 compiler and a licence. This is a third party tool so please download it directly from KEIL. Please read our knowledge base article about the licensing options [6].
3. Download and copy the application note sample source code [8] project to the example folder of the DolphinAPI. (e.g. when you installed DolphinAPI to C:\Program Files\EnOcean\DolphinAPI then the examples folder will be C:\Program Files\EnOcean\DolphinAPI\EO3000I_API\Examples) The DolphinAPI files are already linked to the project. You need to copy the project to the examples folder to match the default file paths.
4. Open the KEIL Project file. (*.uvproj)
5. Find the radio settings assembler file corresponding to your module. You can find the latest radio settings file in the DolphinAPI installations folder (e.g. C:\Program Files\EnOcean\DolphinAPI\EO3000I_API\EO3000I_LIB\RadioCFG).
6. Link the radio settings file to the project workspace in Keil.



Please note that only one radio settings file can be linked to the project. Before adding new radio settings files please exclude the previous.

7. Determine which pins are going to be used for SDA, SCL. By default these are SDA = SCSEDIO0 and SCL = SCLKDIO1.

I²C DOLPHIN INTERFACE

8. If needed, change the communication pins. The pins are determined by defines in `i2c.c`:

```
#define sda_pin SCSEDIO0
#define scl_pin SCLKDIO1
```

Change SCSEDIO0 and SCLKDIO1 to the corresponding pins you have chosen. Please consider restrictions mentioned in chapter 2.
9. Connect your module to the PC through an EOP (EnOcean Programmer). You will probably also use an EVA board.
10. Compile and download the firmware to the module.
11. Please prepare the SHT 21 sensor and extension board (layout) as described in the Sensirion application note.

For further assistance please contact EnOcean Support [4].

Disclaimer

The information provided in this document describes typical features of the EnOcean radio 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.