AN5203
Application note

Teseo-LIV3F - I2C Positioning Sensor

Introduction

Teseo-LIV3F supports two interfaces: UART and I2C and on both it provides the NMEA messages.

The standard NMEA over I2C interface is a mirror of NMEA over UART interface; this interface is designed for Navigation solution platform where the GNSS continuously fills the I2C buffer with satellites and position data and the external microprocessor has to continuously pull the Teseo-LIV3F solution to flush the I2C buffer.

There are applications where the main-host does not care on NMEA stream at all and position awareness is required only when specific events happens. This is the so called *I2C Positioning Sensor* where Host reads the real-time position data, like any other sensor over I2C (i.e.: an accelerometer), when needed without flushing the whole I2C internal buffer.

This document describes how configure and interact with the Teseo-LIV3F through I2C bus using the Teseo-LIV3F as *I2C Positioning sensor*.

Contents

List of tables

List of figures

1 Design description

I2C Positioning sensor is a platform solution achievable with Teseo-LIV3F standard firmware.

In this document, it is described:

- how to configure the Teseo-LIV3F firmware
- the runtime commands to interact with Teseo-LIV3F using the I2C bus

to achieve the I2C Positioning sensor with Teseo-LIV3F.

1.1 Configure the firmware

The Teseo-LIV3F firmware configuration needs to be modified to support I2C Positioning Sensor.

The I2C interface has its own message-list bitmap (CDB-ID 231 and CDB-ID 232), this 64bits bitmap specifies which messages are continuously sent on the internal I2C buffer and ready to be sent when an I2C read operation is triggered by the host.

Set the I2C-message-list to zero (CDB-ID 231 and CDB-ID 232) to avoid the GNSS continuously fills the I2C internal buffer with satellite and positioning data.

Take care that a resetted i2c-message-list affects only the autonomously NMEA messages while the message sent in replay of a commands are still generated.

1.2 The $PSTMNMEAREQUEST command

The I2C Positioning Sensor on Teseo-LIV3F bases on the command: `$PSTMNMEAREQUEST`.

As reported in the Teseo-LIV3F – Software User Manual the command `$PSTMNMEAREQUEST` forces the Teseo-LIV3F to replay with NMEA messages according to the input message list as specified in the CDB-ID 231 and CDB-ID 232.

Synopsis:

```
$PSTMNMEAREQUEST,<msglist_l>,<msglist_h>*<checksum><cr><lf>
```

This means that the main-host can force the Teseo-LIV3F to replay with defined NMEA message based on the 64bits bitmap input the host provides.

2 Prepare the demo platform

The demo application, shown in these pages, is built using the external micro STM32, hosted in a NUCLEO-F401RE, and the Teseo-LIV3F, hosted on X-NUCLEO-GNSS1A1; when an events happens (the blue-button on the NUCLEO-F401RE is pressed) the STM32 will request the message $GPGLL to the Teseo-LIV3F.

2.1 Hardware pre-requisite

The following hardware is required in this demo:

- A X-NUCLEO-GNSS1A1 as Teseo-LIV3F solution
- A NUCLEO-F401RE as host processor
- USB-Cable to connect the NUCLEO-F401RE to the PC
- FTDI-UART2USB cable

2.2 Software pre-requisite

From *www.st.com* download and install:

- the TESEO-SUITE
- the STM32CUBE-MX
- the Atollic-TRUE Studio

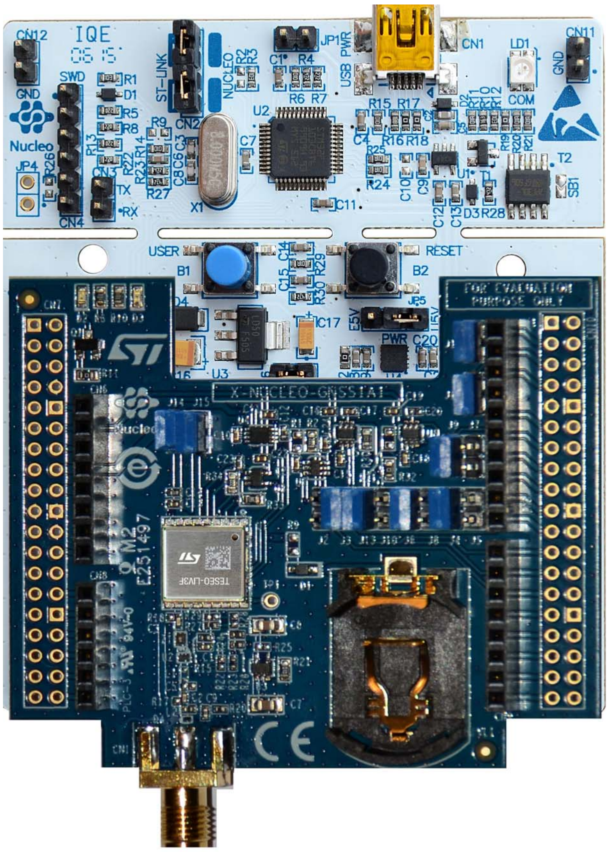
Also a Terminal-Console on PC is required to print-out information coming from STM32.

2.3 Configure the firmware on the X-NUCLEO-GNSS1A1

The Teseo-LIV3F configuration on top of X-NUCLEO-GNSS1A1 can be modified using the TESEO SUITE PC Tool.

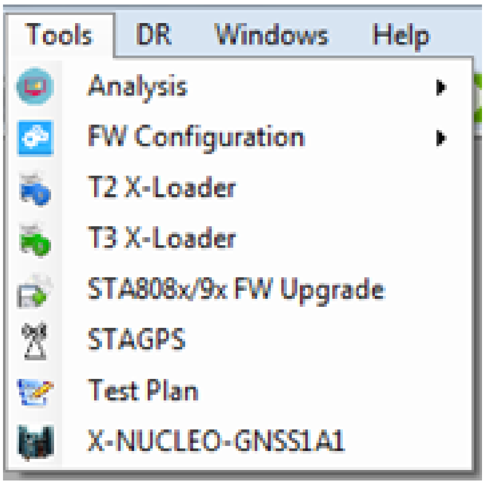
Connect the X-NUCLEO-GNSS1A1 on top of NUCLEO-F401RE as show in *Figure 1*.

Figure 1. X-NUCLEO-ON-TOP-OF NUCLEO



1. Install the STM32 firmware
   a) Select *Tool->X-NUCLEO-GNSS1A1* In the menu as in *Figure 2*

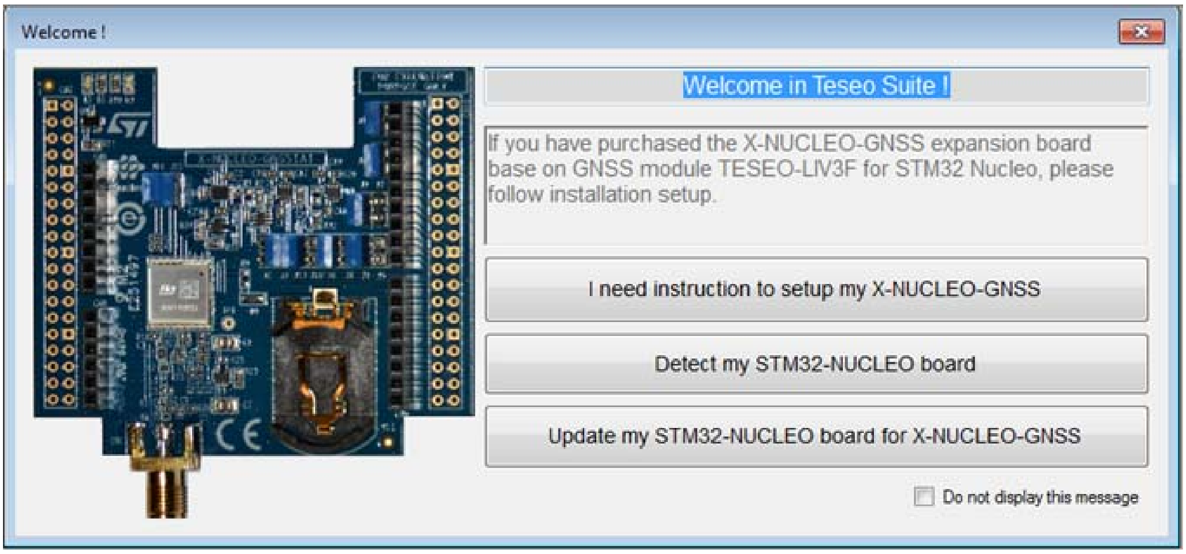
Figure 2. Tool->X-Nucleo-GNSS1A1 path in the Menu



b) The X-NUCLEO-GNSS1A1 panel will appear as in *Figure 3*

c) Push the ‘Update my STM32-NUCLEO board for X-NUCLEO-GNSS

Figure 3. X-NUCLEO-GNSS1A1 panel



Welcome !

Welcome in Teseo Suite !

If you have purchased the X-NUCLEO-GNSS expansion board base on GNSS module TESEO-LIV3F for STM32 Nucleo, please follow installation setup.

I need instruction to setup my X-NUCLEO-GNSS
Detect my STM32-NUCLEO board
Update my STM32-NUCLEO board for X-NUCLEO-GNSS
□ Do not display this message

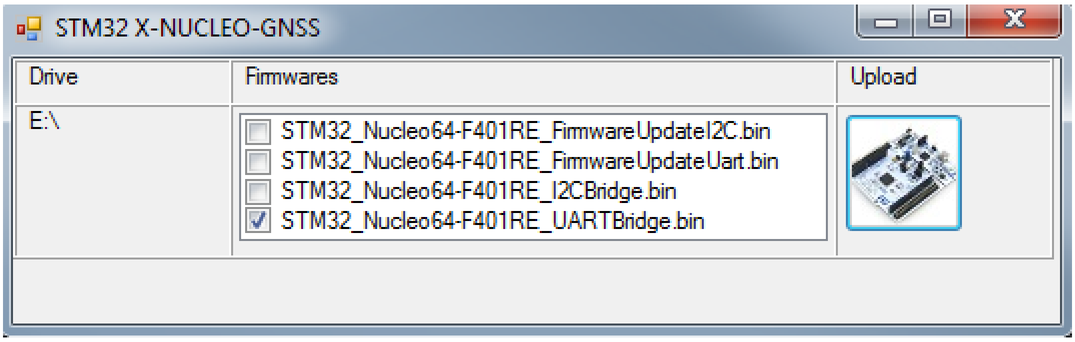
d) The *STM32 X-Nucleo-GNSS* panel will appear as in *Figure 4*

e) Select the firmware ‘

f) *STM32_Nucleo64-F401RE_UARTBridde.bin’*

g) Push the *Upload* button

Figure 4. STM32 X-Nucleo-GNSS



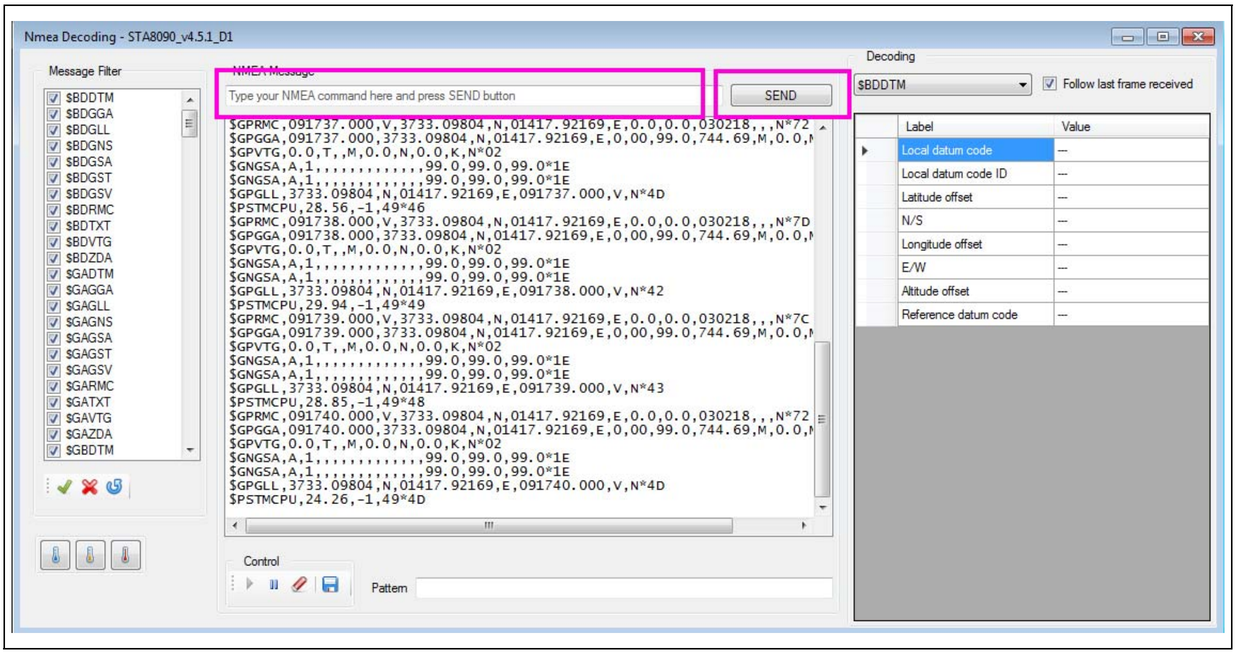
STM32 X-NUCLEO-GNSS

Drive	Firmwares	Upload
E:\	□ STM32_Nucleo64-F401RE_FirmwareUpdateI2C.bin □ STM32_Nucleo64-F401RE_FirmwareUpdateUart.bin □ STM32_Nucleo64-F401RE_I2CBridge.bin ☑ STM32_Nucleo64-F401RE_UARTBridge.bin	

2. Open the TESEO-SUITE PC Tool and
3. Enable an UART/COM connection against the NUCLEO-F401RE.

Using the *NMEA Decoding Panel* (shown in *Figure 5*) on the TESEO-SUITE send the following commands:

Figure 5. NMEA Decoding Panel of TESEO-SUITE



```
$GPRMC,091737.000,V,3733.09804,N,01417.92169,E,0.0,0.0,030218,,,N*72
$GPGGA,091737.000,3733.09804,N,01417.92169,E,0,00,99.0,744.69,M,0.0,M
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GPGLL,3733.09804,N,01417.92169,E,091737.000,V,N*4D
$PSTMCPU,28.56,-1,49*46
$GPRMC,091738.000,V,3733.09804,N,01417.92169,E,0.0,0.0,030218,,,N*7D
$GPGGA,091738.000,3733.09804,N,01417.92169,E,0,00,99.0,744.69,M,0.0,M
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GPGLL,3733.09804,N,01417.92169,E,091738.000,V,N*42
$PSTMCPU,29.94,-1,49*49
$GPRMC,091739.000,V,3733.09804,N,01417.92169,E,0.0,0.0,030218,,,N*7C
$GPGGA,091739.000,3733.09804,N,01417.92169,E,0,00,99.0,744.69,M,0.0,M
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GPGLL,3733.09804,N,01417.92169,E,091739.000,V,N*43
$PSTMCPU,28.85,-1,49*48
$GPRMC,091740.000,V,3733.09804,N,01417.92169,E,0.0,0.0,030218,,,N*72
$GPGGA,091740.000,3733.09804,N,01417.92169,E,0,00,99.0,744.69,M,0.0,M
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,,,,,,,,,,,,,99.0,99.0,99.0*1E
$GPGLL,3733.09804,N,01417.92169,E,091740.000,V,N*4D
$PSTMCPU,24.26,-1,49*4D
```

Label	Value
Local datum code	—
Local datum code ID	—
Latitude offset	—
N/S	—
Longitude offset	—
E/W	—
Altitude offset	—
Reference datum code	—

```
$PSTMCFGMSGL,3,1,0,0

$PSTMSETPAR,1227,1,2

$PSTMSAVEPAR
```

This set of commands will:

- Reset the i2c message list
- Disable the eco-ing message
- Save the configuration on flash

Now the Teseo-LIV3F is configured to support I2C Positioning Sensor.

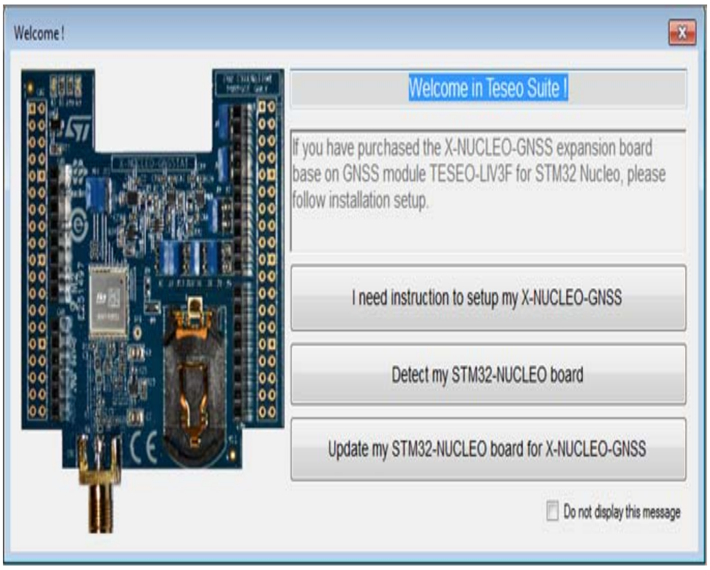
2.4 Prepare the demo platform

The demo is based on:

- NUCLEO-F401RE: which is a prototyping platform of STM32 microcontroller
- X-NUCLEO-GNSS1A1: which is the prototyping platform of Teseo-LIV3F

Connect the X-NUCLEO-GNSS1A1 on top of NUCLEO-F401RE as show in *Figure 6*:

Figure 6. Setup HW demo platform



Welcome !

Welcome in Teseo Suite !

If you have purchased the X-NUCLEO-GNSS expansion board base on GNSS module TESEO-LIV3F for STM32 Nucleo, please follow installation setup.

I need instruction to setup my X-NUCLEO-GNSS
Detect my STM32-NUCLEO board
Update my STM32-NUCLEO board for X-NUCLEO-GNSS
□ Do not display this message

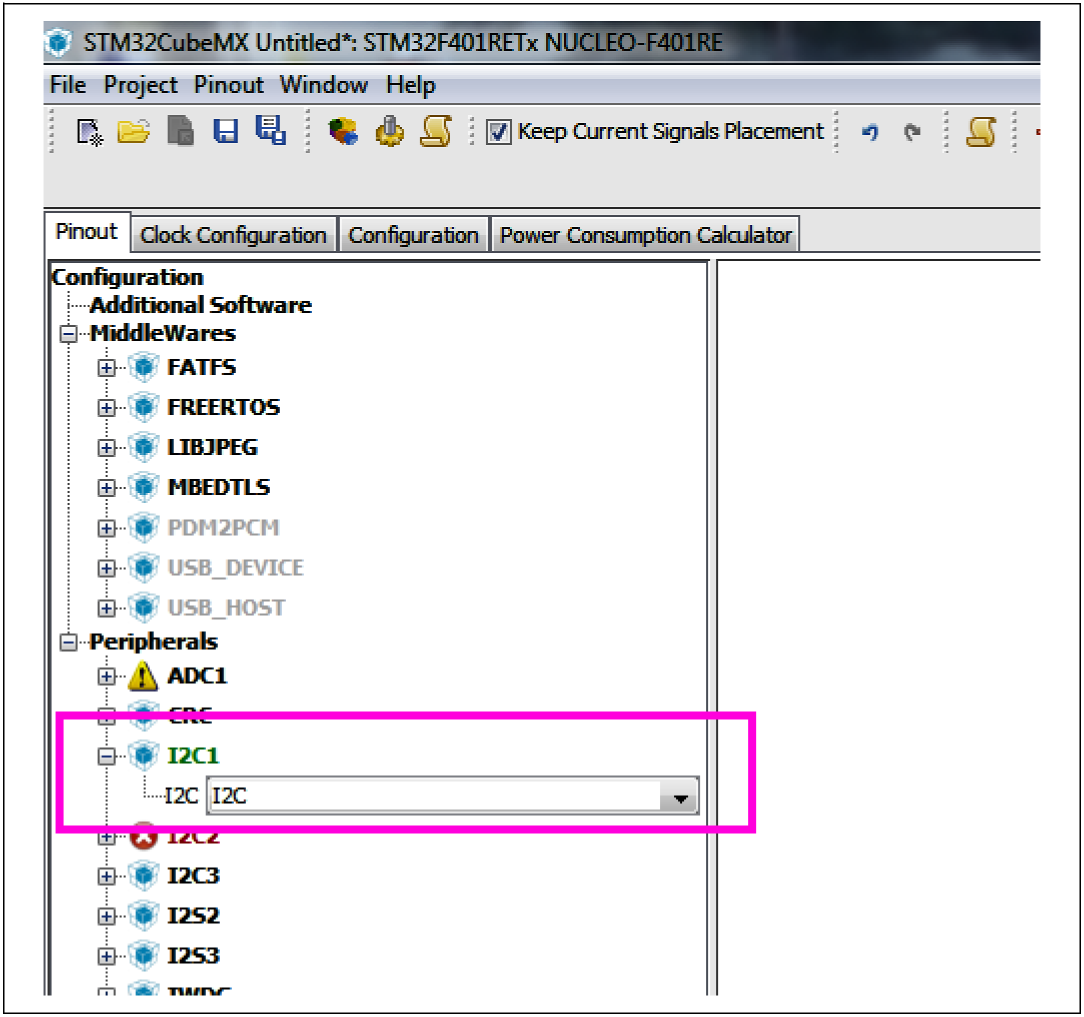
Connect the NUCLEO-F401RE to the PC using the USB-Cable.

3 Configure a STM32Cube-MX project

Create a new project for STM32 on NUCLEO-F401RE

Enable the I2C-1 bus, this will be the I2C communication channel between STM32 and Teseo-LIV3F as shown in *Figure 7*

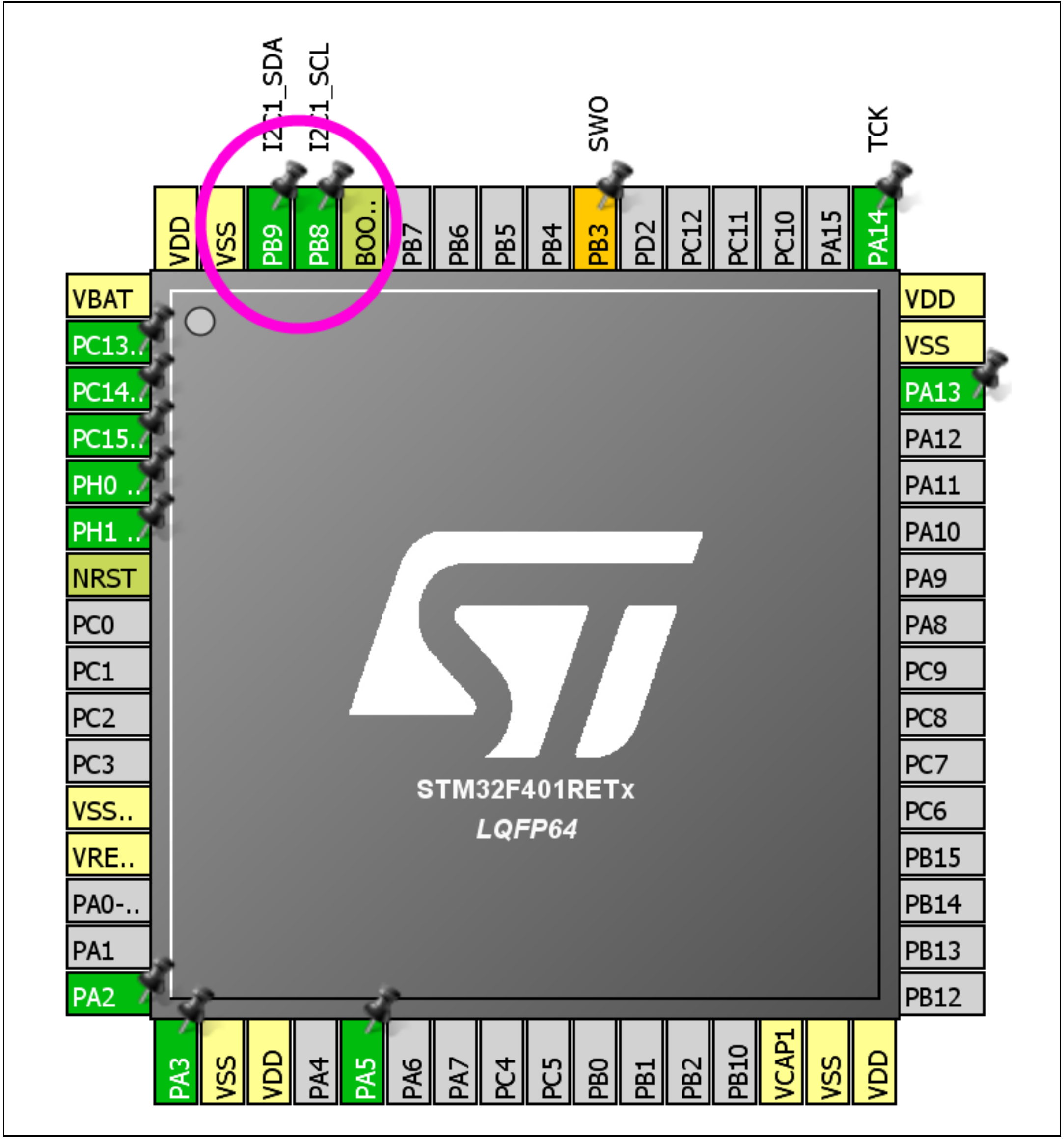
Figure 7. Enable the I2C-1 bus on NUCLEO-F401RE



I2C I2C

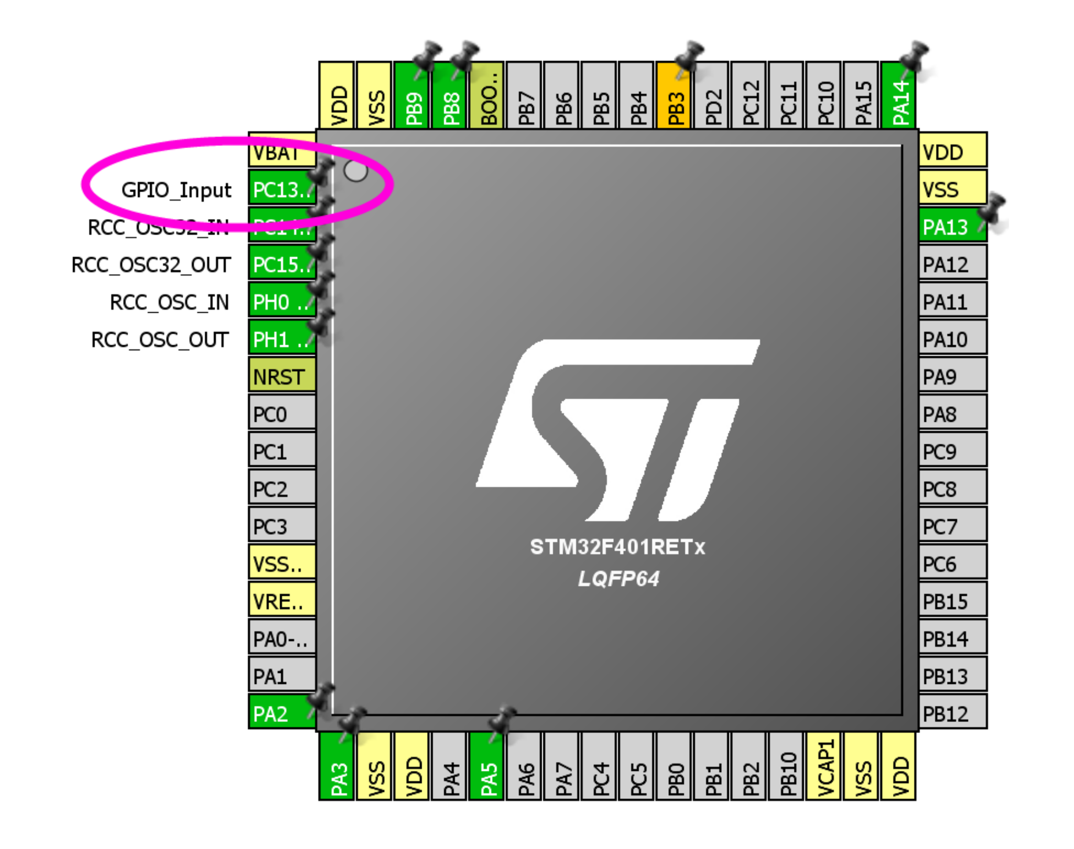
Route the I2C bus on right pins PB8 and PB9 as shown in *Figure 8*.

Figure 8. I2C pins configuration on NUCLEO-F401RE



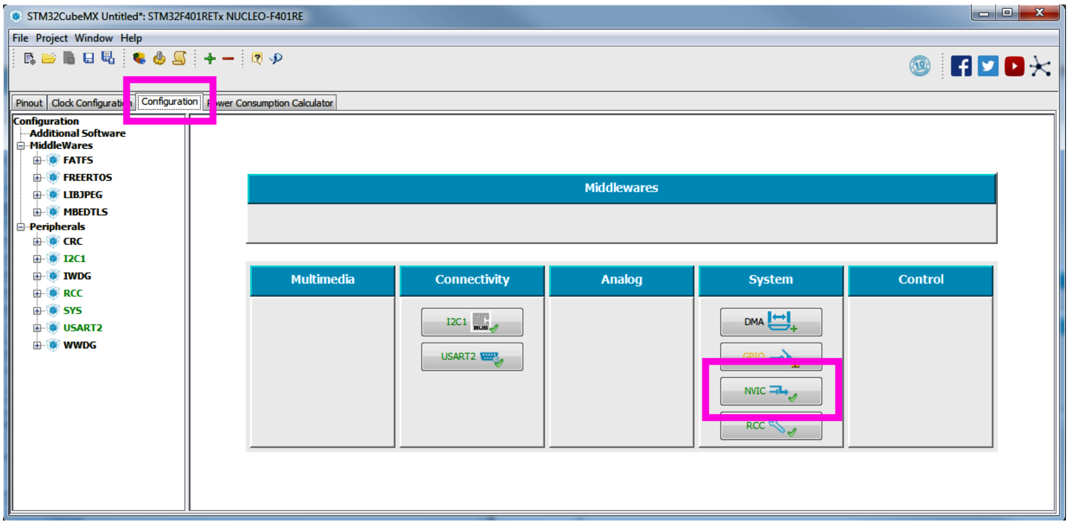
Configure the Blue-Button as GPIO_EXTI13 as shown in *Figure 9*.

Figure 9. Blue button configuration on NUCLEO-F401RE



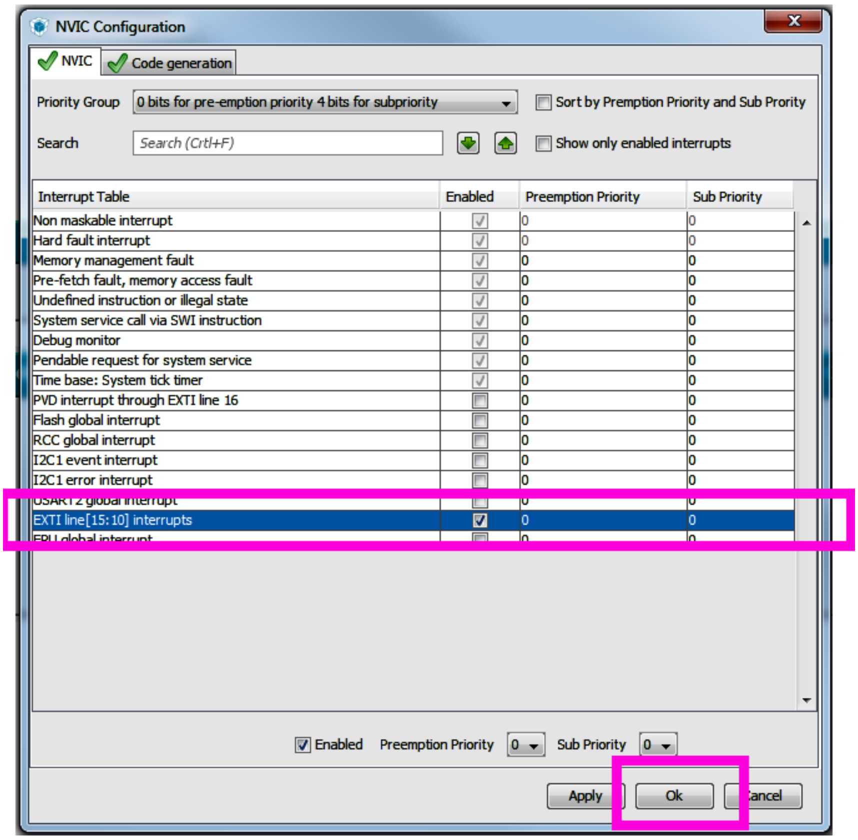
In the configuration TAB press the NVIC button to configure the NVIC shown in *Figure 10*.

Figure 10. STM32-CubeMX configuration TAB



Enable the Blue-Button (EXTI-interrupts) and confirm the configuration pressing the OK button as shown in *Figure 11*.

Figure 11. Configure EXTI interrupt



NVIC Configuration

NVIC | Code generation

Priority Group: 0 bits for pre-emption priority 4 bits for subpriority

□ Sort by Premption Priority and Sub Prority

Search: Search (Crtl+F)

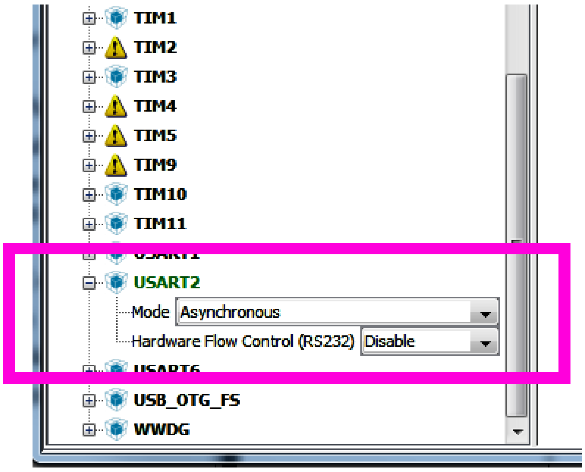
□ Show only enabled interrupts

Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	☑	0	0
Hard fault interrupt	☑	0	0
Memory management fault	☑	0	0
Pre-fetch fault, memory access fault	☑	0	0
Undefined instruction or illegal state	☑	0	0
System service call via SWI instruction	☑	0	0
Debug monitor	☑	0	0
Pendable request for system service	☑	0	0
Time base: System tick timer	☑	0	0
PVD interrupt through EXTI line 16	□	0	0
Flash global interrupt	□	0	0
RCC global interrupt	□	0	0
I2C1 event interrupt	□	0	0
I2C1 error interrupt	□	0	0
USART2 global interrupt	□	0	0
EXTI line[15:10] interrupts	☑	0	0
FPU global interrupt	□	0	0

☑ Enabled Preemption Priority 0 Sub Priority 0

Enable the USART-2 to enable communication between STM32 and the Terminal-Console running on PC as shown in *Figure 12*.

Figure 12. Enable the USART-2 on NUCLEO-F401RE

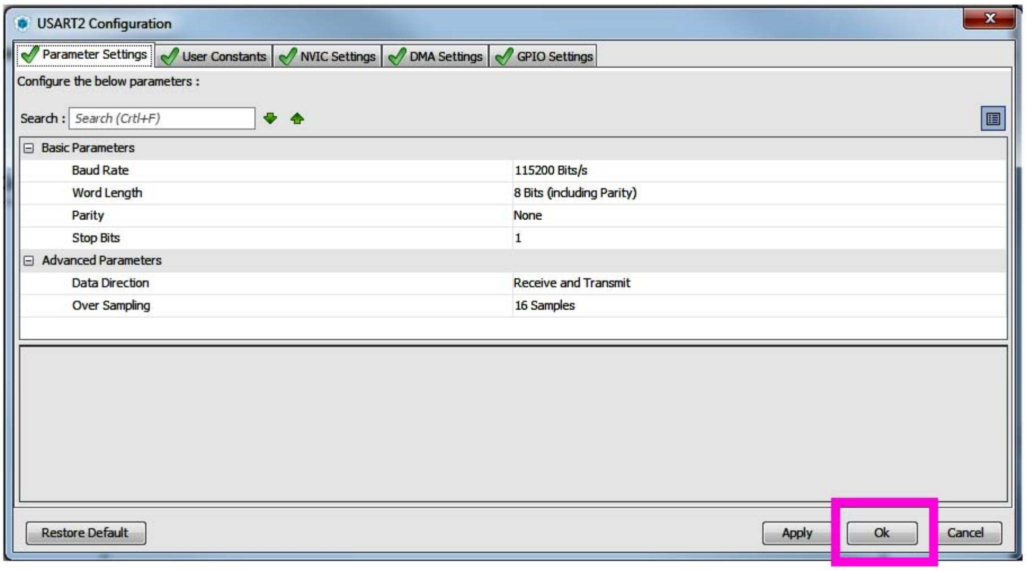


Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Configure the USART-2 as shown in *Figure 13*.

Figure 13. USART-2 configuration on NUCLEO-F401RE



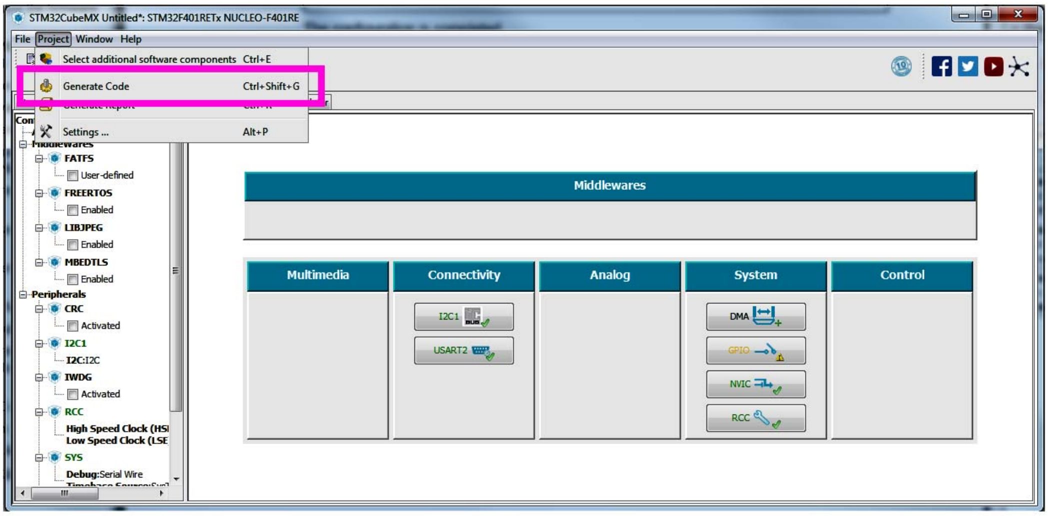
Configure the below parameters :

Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

The configuration is completed.

The skeleton C-Code can generate as shown in *Figure 14*.

Figure 14. Generate the C-Code



The project can now be opened with an IDE: Atollic TRUE STUDIO in this case.

4 Finalize the C-Code in the project

STM32Cube-MX during the Code Generation prepared all the skeleton C- Code to configure and initialized:

- the I2C bus;
- the UART port against the PC
- the Blue-button interrupt

In this project very few C-Code needs to be added.

4.1 Utility function

Just added an utility function to print-out information in a Terminal on PC

```
void Console_Write(uint8_t *string){
  HAL_UART_Transmit(&huart2, string, strlen((char *)string), 1000);
}
```

4.2 Blue Button call-back

Added the callback called on a Blue-Button pressed event:

```
static volatile _read = 0;
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
       _read = 1;
}
```

4.3 Prepare ad-hoc commands for the required message

In this demo, the host will request only the $GPGLL message. This message is identified in the message-list with the number 0x100000 (inline with the CDB-ID 231-232 description where the message $GPGLL is identified as bit 20 in the message-list-low).

The related NMEA command is defined as:

```
static const char *gpgll_msg = "$PSTMNMEAREQUEST,100000,0\n\r";
```

4.4 Main C-code

The final C-code in the main function could be like:

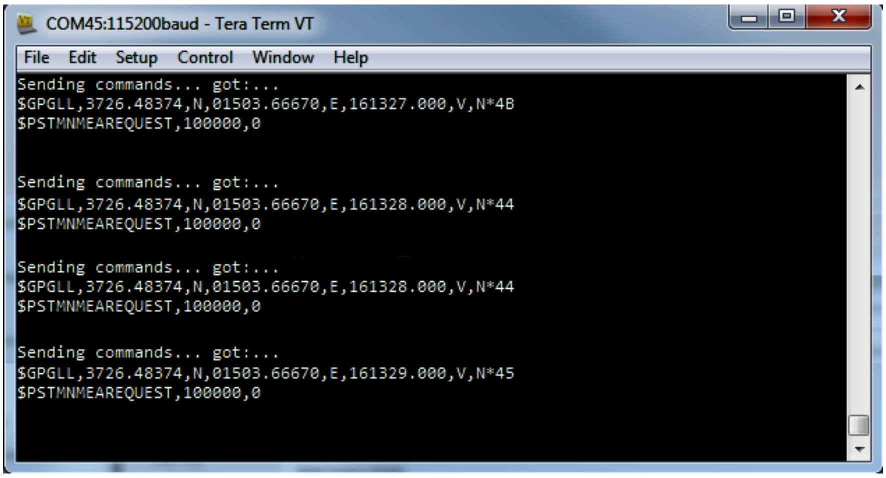
```
while (1) {
HAL_I2C_DeInit(&hi2c1);
HAL_I2C_Init(&hi2c1);
4 if (_read == 1) {
5    _read = 0;
6    Console_Write("Sending commands...");
7    HAL_I2C_Master_Transmit(&hi2c1, 0x3A << 1, (uint8_t
     *)gpgll_msg, strlen(gpgll_msg), 2000);
8    Console_Write(" got:... \r\n");
9    for (read_buf[180-1] = 0; read_buf[180-1] != 0xff;) {
10     HAL_I2C_Master_Receive(&hi2c1, 0x3A << 1, read_buf, 180,
2000);
11     for (i = 0; i < 180; ++i)
12       if (read_buf[i] != 0xff)
13         HAL_UART_Transmit(&huart2, &read_buf[i], 1, 1000);
14     }
15      Console_Write("\r\n");
16    }
17  }
```

Where in detail:

1. in line-4 the application waits the Blue-button event happens.
2. In line-7 the main-processor send to the Teseo-LIV3F on I2C-bus the `$PSTMNMEAREQUEST` command to request a *$GPGLL* message
3. In line-10 the main processor read the data sent by Teseo-LIV3F on I2C-bus
4. In lines 11-13 the received data are sent on the PC-Terminal console

Results can be inspect on a Terminal console on PC as shown in *Figure 15*.

Figure 15. Terminal console result



```
Sending commands... got:...
$GPGLL,3726.48374,N,01503.66670,E,161327.000,V,N*4B
$PSTMNMEAREQUEST,100000,0


Sending commands... got:...
$GPGLL,3726.48374,N,01503.66670,E,161328.000,V,N*44
$PSTMNMEAREQUEST,100000,0

Sending commands... got:...
$GPGLL,3726.48374,N,01503.66670,E,161328.000,V,N*44
$PSTMNMEAREQUEST,100000,0

Sending commands... got:...
$GPGLL,3726.48374,N,01503.66670,E,161329.000,V,N*45
$PSTMNMEAREQUEST,100000,0
```

Said that main host has only to parser the $GPGLL message to obtain the required information.

5 Conclusion

In these pages, it has been shown how Teseo-LIV3F can be used as I2C Positioning Sensor where positioning data or any kind of information are provided to the main-macro on demand based on the main-macro needs.

Host has only to raise well defined $PSTMNEAREQUEST command to query the Teseo-LIV3F

This kind of solution can simplify the final application design because the main-macro does not need pulling continuously the Teseo-LIV3F and, when needed, with a simple command it can required the data needed.

Appendix A Acronyms

Table 1. Acronyms

Keyword	Definition
Accuracy	Deviation of a GPS-based calculated position from the true position
ADC	Analogue to Digital Converter
Almanac	Contains the information about all available satellites, their orbit data and time of their clocks.
ANF	Adaptive Notch Filter
Azim	Azimuth - Angular distance from a reference
Bank Swap	Exchanging two memory banks for storage of data
BAUD rate	Transmission Rate Measure for the effective transmission of data content. (may differ from Bits/sec).
BeiDou	China’s regional navigation satellite system
Checksum	Calculated from the transmitted characters of a message by “ex-OR”ing the 8 bit character values excluding delimiters $ and *
CN0	Carrier to Noise Ratio - Identifies the quality of a received signal
Cold Start	Start Condition for a GPS system having no position nor time. Almanac and Ephemeris is not available, too.
BeiDou	China’s global navigation satellite system (also known as Beidou-2, BD2)
Dead Reckoning	Sensor based process to determine the movement of a mobile unit, utilizing Gyro, Odometer and Wheel Pulses.
Delimiter (within NMEA 0183)	ASCII “$” to indicate Address Field ASCII “,” to indicate Data Field ASCII “*” to indicate Checksum Field
DGPS	Differential GPS - GPS Augmentation System providing the accurate location of a Reference Station to reduce system errors.
EGNOS	European Geostationary Navigation Overlay System
Elev	Elevation - Angle between a high level or non-earth bound point and the horizontal plane of the viewer.
Ephemeris	Ephemeris Data is transmitted by each satellite and contains current and predicted satellite position.
FDA	Failure Detection Algorithm - Specific Algorithm to detect failures in position calculation
FDE	False Detection Exclusion
GALILEO	Europe’s global navigation satellite system
GDOP	Geometric Dilution Of Position - Quality value representing all geometry based error factors in a system.
GNSS	Global Navigation Satellite System - Satellite based system to calculate the position of the Teseo on the earth surface.
GPS	Global Positioning System - United States Satellite Navigation System

Table 1. Acronyms (continued)

Keyword	Definition
GPS Library	STMicroelectronics C-Library containing all GPS relevant Functions
Gyro	Gyroscope - Sensor to determine rotational movements
HDOP	Horizontal Dilution Of Precision - Quality value representing all 2D plane geometry based error factors in a system.
Hot Start	Start Condition for a GPS System having position, time, Almanac and Ephemeris already available. High time accuracy is required.
IMU	Inertial Measurement Unit
Lat	Lattitude - Angular difference of a given position to the Equator. Values include 0°-90° either North or South
Lat-Ref	Lattitude Reference - Reference if a Latitude value is North or South
Long	Longitude - Angular difference to a “reference” Longitude indicated as “000”. Values include 0°… 180° either West or East.
Long-Ref	Longitude Reference - Reference if a Longitude value is East or West of the “000” Meridian.
NMEA	National Marine Electronics Association - United States Standards Organisation For Marine Equipment
NMEA 0183	National Marine Electronics Association - Standard for Interfacing Marine Electronics Devices
NVM	Non Volatile Memory - Any type of memory that conserves data in the absence of regular supply voltage (includes battery buffered memories)
Proprietary Message	Messages within the scope of NMEA0183 which are not standardized. They start with $P and a 3 character identifier.
PRN	Pseudo Random Number - Satellite Specific 1023 Bit Number used for Spread Spectrum Modulation
RAIM	Teseo Autonomous Integrity Monitoring
RF	Radio Frequency - High Frequency for Reception with a RF-Teseo
RS232	IEEE Standard - Physical Layer Standard for Data Transmission
Sat-ID	Satellite Identifier - Satellite specific Number used to generate the corresponding PRN code
SBAS	Satellite Based Augmentation System - GPS enhancement system based on geostationary satellites.
SPS	Standard Positioning Service
Static Position Filtering	Algorithm to detect that the GPS Teseo doesn´t move and position output is kept stable.
UTC	Universal Time Coordinated
WAAS	Wide Area Augmentation System - American GPS Augmentation System delivering accurate Ionosphere Data
Warm Start	Start Condition for a GPS system having current Almanac, position and time availability. Ephemeris are not available. Time needs to be available with reasonable accuracy (some seconds).

Table 1. Acronyms (continued)

Keyword	Definition
2D Fix	Fix based on the use of 3 satellites
3D Fix	Fix based on the use of 4 satellites

Revision history

Table 2. Document revision history

Date	Revision	Changes
28-Aug-2018	1	Initial release.
10-Jun-2019	2	Updated *Section 2.3: Configure the firmware on the X-NUCLEO-GNSS1A1*.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to *www.st.com/trademarks*. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.