Getting started with the X-CUBE-GNSS1 Global Navigation Satellite System software expansion for STM32Cube

Introduction

The X-CUBE-GNSS1 is an expansion software package for the STM32Cube allowing for easy integration with STM32 microcontrollers.

This document provides information on the examples that are provided as part of the X-CUBE-GNSS software package V7.1.0 (minimum-required version V6.15.0). The document describes the hardware setup, software configuration, and the procedure to verify the execution result. The hardware setup includes set-up details for the STM32 evaluation board and the GNSS evaluation board. The software setup describes any additional tools that would be required as part of running the program. As the result of the execution is printed on to the console of a serial terminal program, the variables holding the result are also described.

Information and documentation related to the Teseo-LIV3F, Teseo-VIC3DA, and Teseo-LIV4F GNSS modules, the X-NUCLEO-GNSS1A1, X-NUCLEO-GNSS2A1 and the X-NUCLEO-LIV4A1 expansion boards and the ST expansion software for GNSS are available on www.st.com. Links are provided at the end of this document in the reference section.

While it is not a prerequisite, it is helpful if the user has a basic understanding of the STM32CubeMX environment. Reference material for the STM32CubeMX can be found in the references section.

Note: Unless it is explicitly written, all settings and configuration steps that refer to the Teseo-LIV3F device and the X-NUCLEO-GNSS1A1 expansion board also apply to the Teseo-VIC3DA, Teseo-LIV4F device, and the X-NUCLEO-GNSS2A1, X-NUCLEO-LIV4A1 expansion boards.

Note: The example document covers examples that are already provided for certain STM32 EVBs. However, for a custom PCB or an STM32 EVB that is not part of the example implementation, the user can refer to the getting started guide placed under the documentation folder of the X-CUBE-GNSS1 software package for the user to implement any of the GNSS examples provided.

Related links

Visit the STM32Cube ecosystem web page on www.st.com for further information

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
CDB	Configuration Data Block
A-GNSS	Assisted global navigation satellite system
BSP	Board support package
BeiDou	China regional navigation satellite system
DRAW	Dead reckoning automotive way
Galileo	Europe global navigation satellite system
GLONASS	Global navigation satellite system
GNSS	Global navigation satellite system
GPS	Global positioning system
HAL	Hardware abstraction layer
I2C	Inter integrated circuit
IC	Integrated circuit
IP	Internet protocol
NMEA	National Marine Electronics Association - United States standards organization for marine equipment
QZSS	Quasi-zenith satellite system (used in the Asia-Oceania regions)
RTOS	Real-time operating system
TCP	Transmission control protocol
UART	Universal asynchronous receiver-transmitter
USB	Universal serial bus

2 X-CUBE-GNSS1 software expansion for STM32Cube

2.1 Overview

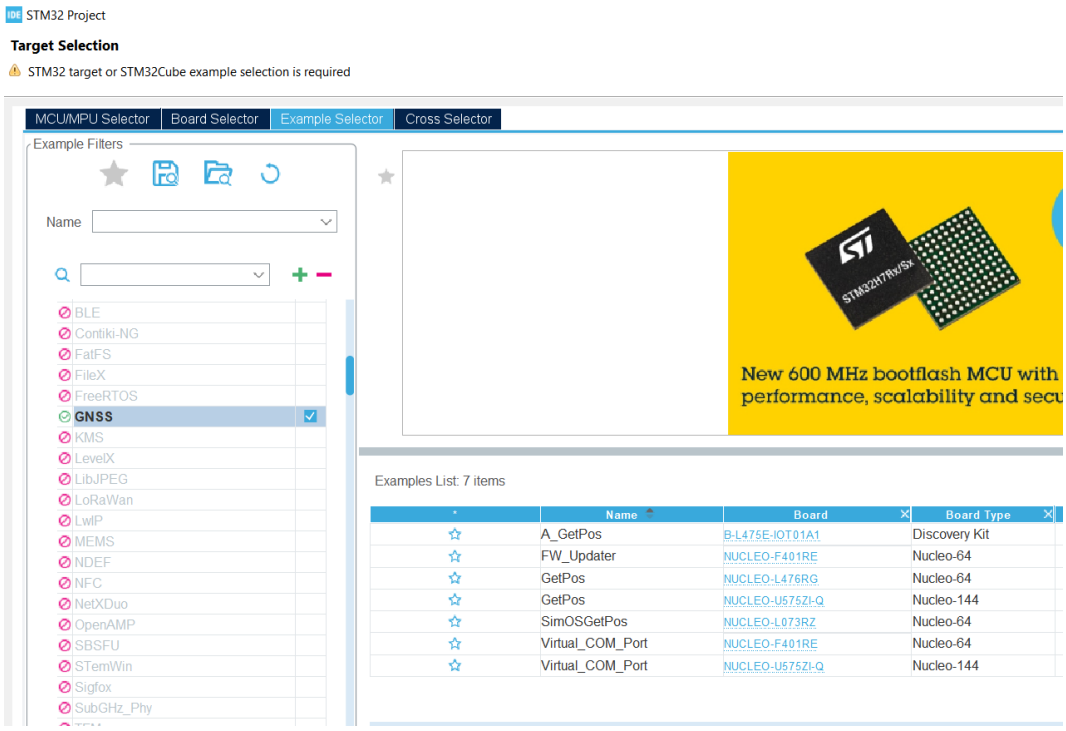
The X-CUBE-GNSS1 is a software package that expands the functionality provided by the STM32Cube.

The key features of the package are:

- Complete software to build applications using the Teseo-LIV3F or Teseo-VIC3DA or Teseo-LIV4F GNSS device.
- Middleware for the NMEA protocol and for assisted GNSS (A-GNSS) support.
- RTOS (Azure® RTOS ThreadX or FreeRTOS™) task scheduling to ensure a better asynchronous message parsing.
- Easy portability across different MCU families, thanks to the STM32Cube.
- Sample application to transmit GNSS data to a PC and for A-GNSS support.
- Free, user-friendly license terms.

The GNSS examples can be selected from the STM32CubeMX window or from STM32CubeIDE by clicking on “Create a New STM32 Project”. Examples are selected for a certain STM32 EVB. When selecting the GNSS example category, the following list of projects is generated. The list shows the application name, the commercial name of the STM32 EVB on which the example is implemented, and the type of STM32 board, that is, if it is a Nucleo or a Discovery board.

Figure 1. Example list of GNSS with associated EVBs



The table below shows a summary view of the projects based on the STM32 board, RTOS selection, and the peripheral being used to communicate with the Teseo module.

Table 2. Projects based on STM32 board

Project name	STM32 evaluation board	RTOS or Bare-Metal	Interface with Teseo module
A_GetPos	B-L475E-IOT01A1	FreeRTOS	I2C1
FW_Updater	NUCLEO-F401RE	Bare-Metal	USART1
GetPos	NUCLEO-F401RE	FreeRTOS	USART1
	NUCLEO-H563ZI	ThreadX	I2C1
	NUCLEO-L476RG	FreeRTOS	I2C1
	NUCLEO-U575ZI-Q	ThreadX	LPUART1
Virtual_COM_Port	NUCLEO-F401RE	FreeRTOS	USART1
	NUCLEO-H563ZI	ThreadX	LPUART1
	NUCLEO-L073RZ	FreeRTOS	I2C1
	NUCLEO-L476RG	FreeRTOS	USART1
	NUCLEO-U575ZI-Q	ThreadX	I2C1
SimOsGetPos	NUCLEO-F401RE	Bare-Metal	USART1
	NUCLEO-H563ZI	Bare-Metal	I2C1
	NUCLEO-L073RZ	Bare-Metal	I2C1
	NUCLEO-L476RG	Bare-Metal	I2C1
	NUCLEO-U575ZI-Q	Bare-Metal	I2C1

I²C settings

Teseo module I²C communication - 100 kHz.

The following table shows the baud rate at which the different STM32 evaluation boards communicate with the PC and the Teseo module, and the jumper settings' modifications needed.

Table 3. UART baud rate settings

Connection	Baud rate
PC	115200
X-NUCLEO-(LIV4A1, GNSS2A1)	115200
X-NUCLEO-GNSS1A1	9600

B-L475E-IOT01A1, NUCLEO-U575ZI-Q, NUCLEO-H563ZI

Table 4. Teseo module connections

Teseo module	Jumper modifications
X-NUCLEO-LIV4A1	Open: J14, J7, J5 Closed: J1, J2, J4, J6, J9 1-2: J12, J10, J11 2-3: J8, J3
X-NUCLEO-GNSS1A1	Closed: J2, J13, J5, J9, J12, J11, J14, J15 Open: J3, J4, J6, J7, J8, J10

Teseo module	Jumper modifications
X-NUCLEO-GNSS2A1	Closed: J11, J12, J14, J15 1-2: J29, J28, J23, J25 2-3: J27, J24, J26, J30

NUCLEO-F401RE, NUCLEO-L476RG, NUCLEO-L073RZ

Table 5. Teseo module connections

Teseo module	Jumper modifications
X-NUCLEO-LIV4A1	Open: J14, J7, J5 Closed: J1, J2, J4, J6, J9 1-2: J8, J12, J10 2-3: J11, J3
X-NUCLEO-GNSS1A1	Closed: J3, J4, J6, J9, J11, J12, J13, J14, J15 Open: J2, J5, J7, J8, J10
X-NUCLEO-GNSS2A1	Closed: J11, J12, J14, J15 1-2: J23, J25, J27, J29 2-3: J24, J26, J28, J30

2.2 Architecture

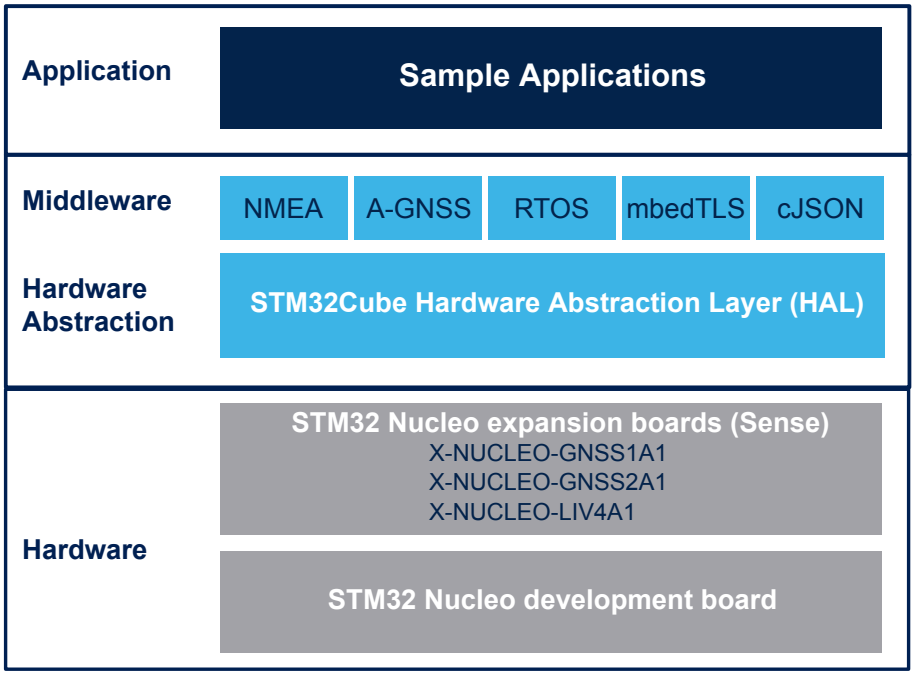
This software is a fully compliant expansion of STM32Cube architecture for the development of applications using the ST Teseo-LIV3F, Teseo-LIV4F, and Teseo-VIC3DA GNSS modules.

The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the GNSS expansion board and middleware components for NMEA protocol and A-GNSS support. The package is also based on the RTOS libraries to exploit the benefits of task scheduling for better asynchronous message parsing.

The software layers used by the application software to access and use the GNSS expansion board are:

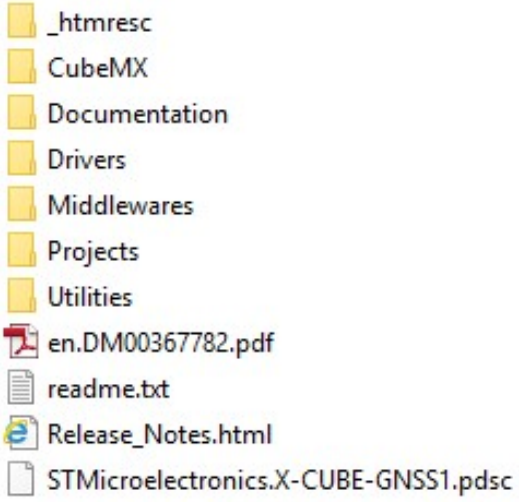
- **STM32Cube HAL layer**: consists of simple, generic, and multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries, and stacks. These generic and extension APIs are based on a common framework so that overlying layers like middleware can function without requiring specific microcontroller unit (MCU) hardware information. This structure improves library code reusability and ensures easy portability across other devices.
- **Board support package (BSP) layer**: provides software support for the STM32 Nucleo board peripherals, excluding the MCU. These specific APIs provide a programming interface for certain board-specific peripherals like LEDs, user buttons, etc., and can also be used to fetch individual board version information. It also provides support for initializing, configuring, and reading data.

Figure 2. **X-CUBE-GNSS1 software architecture**



2.3 Folder structure

Figure 3. **X-CUBE-GNSS1 package folder structure**



The following folders are included in the software package:

- **CubeMX**: includes configuration files for the STM32CubeMX tool.
- **Documentation**: contains a compiled HTML file generated from the source code, detailing the software components and APIs.
- **Drivers**: contains the HAL drivers, the board-specific drivers for each supported board or hardware platform (including the onboard components), and the CMSIS vendor-independent hardware abstraction layer for the Arm Cortex®-M processor series.
- **Middlewares**: contains a library for the NMEA protocol support (for example, for NMEA message parsing), the RTOS libraries for task scheduling management, an A-GNSS library for assisted GNSS support, and a cJSON library for JSON message parsing.

- **Projects**: contains the sample applications used to access the GNSS data provided for the NUCLEO-H563ZI, NUCLEOF401RE, NUCLEO-L476RG, NUCLEO-L073RZ, and NUCLEO-U575ZI-Q platforms and a sample application for A-GNSS provided for the B-L475E-IOT01A discovery kit for IoT nodes. All the applications are available under three development environments: IAR Embedded Workbench for Arm (IAR-EWARM), RealView Microcontroller Development Kit (MDK-ARM), and STM32CubeIDE.
- **Utilities**: contains a “PC_Software” folder with a Java utility to update the Teseo-LIV3F device firmware.

2.4 APIs

Detailed technical information about the APIs available to the user can be found in a compiled HTML file located inside the “Documentation” folder of the software package where all the functions and parameters are fully described.

2.5 Sample applications for X-CUBE-GNSS1

This section explains the internals of each of the sample applications provided in the X-CUBE-GNSS1. Below are the common steps for each of the applications.

General usage flow:

- Mount the X-NUCLEO-GNSSx board on the STM32 Arduino® connector and configure the jumper settings as described in Section 2.5.1 and Section 2.6: GNSS firmware upgrader utility. A readme file is also present in the application folder (readme.html) for the project.
  - Connect the STM32 Nucleo board using a USB cable to the PC.
  - Open any serial terminal program and set the serial port to 115200 (for X-NUCLEO-GNSS2A1 and X-NUCLEO-LIV4A1) or 9600/115200, 8, N, 1. Refer to Section 2.5.1 to choose the appropriate baud rate setting for the STM32 Nucleo board and X-NUCLEO-GNSS1A1 board.
  - Open the project (IAR, Keil®, or STM32CubeIDE), compile and run/debug.
  - The output for the project is displayed on the serial terminal.

2.5.1 Virtual_COM_Port

RTOS:

FreeRTOS for NUCLEO-F401RE, NUCLEO-L473RG and NUCLEO-L073RZ

ThreadX for NUCLEO-U575ZI-Q and NUCLEO-H563ZI

STM32:

NUCLEO-F401RE

NUCLEO-U575ZI-Q

NUCLEO-L476RG

NUCLEO-H563ZI

NUCLEO-L073RZ

Description:

This example emulates a COM port for the GNSS module using the STM32. NMEA data from the Teseo module is received by the STM32 which outputs the NMEA data over UART onto a serial terminal program.

Using RTOS (FreeRTOS™ or ThreadX) two threads are created,

1. **TeseoConsumerTask:** this thread is for reading the NMEA messages coming from the Teseo module over I²C or UART and prints the result on the console.
   In TeseoConsumerTask, the NMEA messages are stored in GNSS1A1_GNSS_Msg_t *gnssMsg data structure and its buffered NMEA message is printed on the console: gnssMsg->buf
   GNSS1A1_GNSS_Msg_t is defined in Drivers\BSP\GNSS1A1\gnss1a1_gnss.h.
   Below are the data structures corresponding to this message:

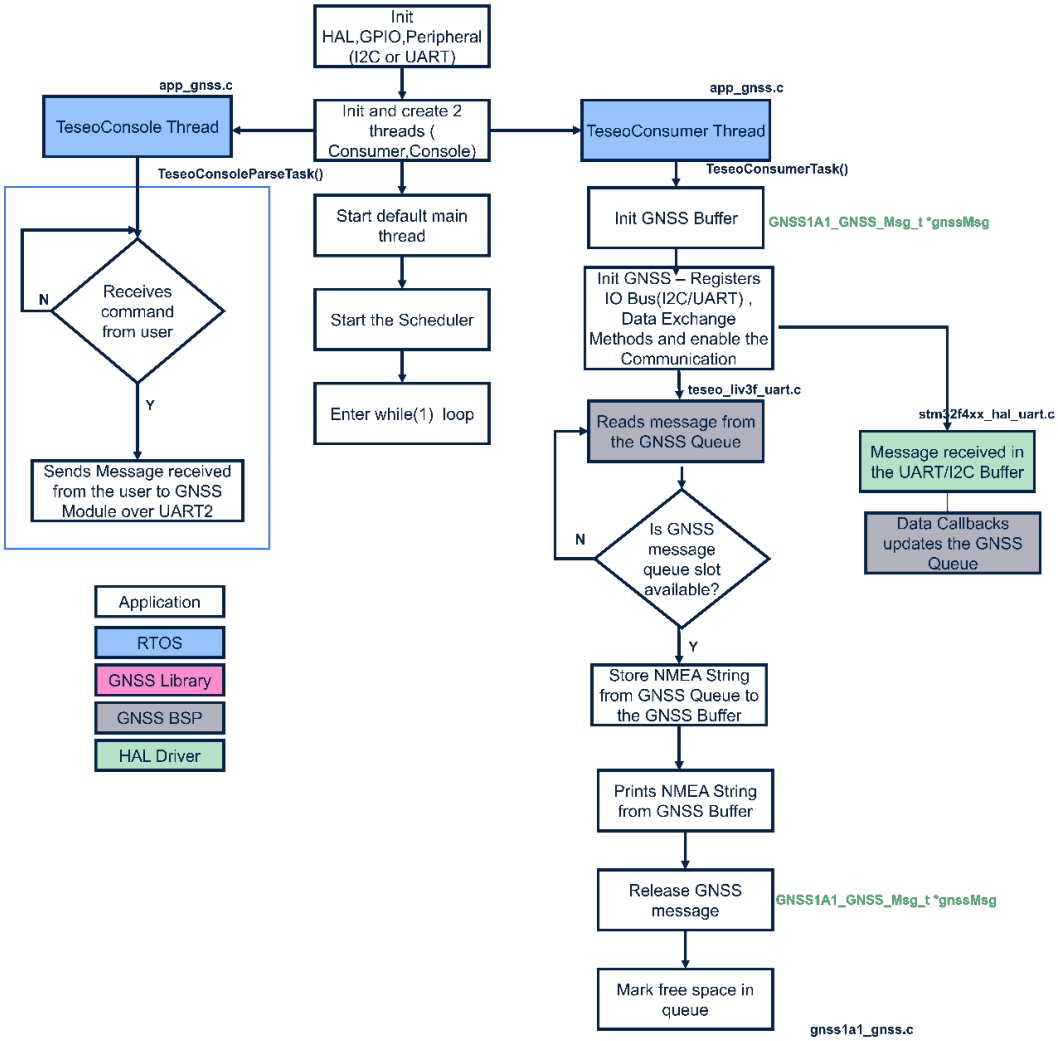
Table 6. Data structure GNSS1A1_GNSS_Msg_t

Data structure	Information
Buf	Message buffer
Len	Length of the message buffer

2. **ConsoleParseTask:** this thread is for reading input from the console. It is used in this example.

The flow chart below illustrates the implementation of a Virtual_COM_Port example.

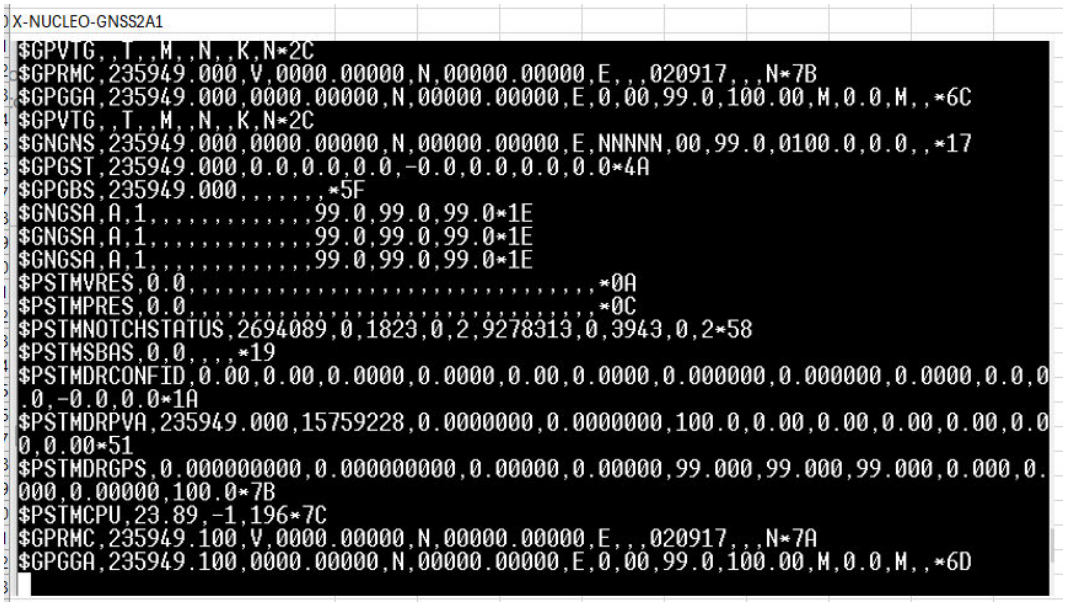
Figure 4. VCOM flow diagram



Result:

As soon as the board is powered up and flashed GNSS NMEA, data starts printing on the screen.

Figure 5. NMEA messages printed on the serial terminal console



2.5.2 GetPos

RTOS:

FreeRTOS for NUCLEO-F401RE and NUCLEO-L476RG

ThreadX for NUCLEO-U575ZI-Q and NUCLEO-H563ZI

STM32:

NUCLEO-L476RG

NUCLEO-U575ZI-Q

NUCLEO-F401RE

NUCLEO-H563ZI

Description:

GNSS NMEA data is continuously read over I²C or UART. The GNSS data structure is populated based on the GNSS NMEA messages received. Using the serial terminal program and selecting the appropriate selection, the user can view the data.

Using RTOS (FreeRTOS™ or ThreadX) two threads are created:

1. **TeseoConsumerTask:** this thread is for reading the NMEA messages over I²C or UART and parsing them into the respective data structure that the application can use directly. Inside TeseoConsumerTask NMEA messages are stored in GNSS1A1_GNSS_Msg_t *gnssMsg. These messages are then parsed using the LibNMEA library. LibNMEA library parses standard NMEA messages like xxGGA, GPGST, xxRMC, xxGSA, and xxGSV and stores the parsed values inside the respective NMEA data structures defined in the \LibGNSS\Inc\gnss_parser.h

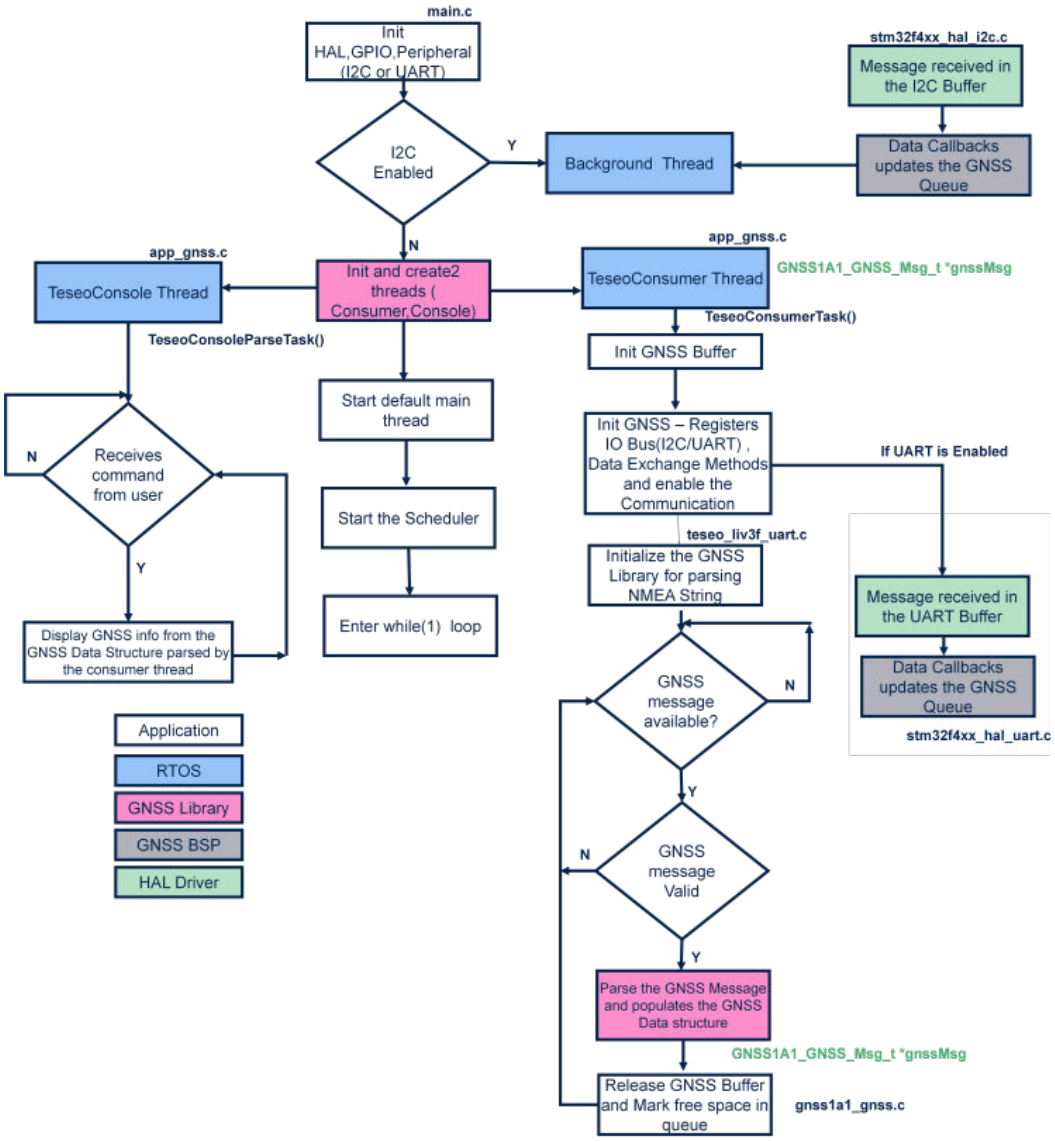
Figure 6. Typedef struct

```
typedef struct
{
  Debug_State debug;      /**< Debug status */
  GPGGA_Info_t gpgga_data; /**< $GPGGA Data holder */
  GNS_Info_t   gns_data;   /**< $--GNS Data holder */
  GPGST_Info_t gpgst_data; /**< $GPGST Data holder */
  GPRMC_Info_t gprmc_data; /**< $GPRMC Data holder */
  GSA_Info_t   gsa_data;   /**< $--GSA Data holder */
  GSV_Info_t   gsv_data;   /**< $--GSV Data holder */

  PSTMVER_Info_t pstmver_data;   /**< $PSTMVER Data holder */
  PSTMPASSRTN_Info_t pstmpass_data; /**< $PSTMPASSRTN Data holder */
  PSTMAGPS_Info_t pstmagps_data; /**< $PSTMAGPS Data holder */

  Geofence_Info_t geofence_data; /**< $PSTMGEOFENCE Data holder */
  Odometer_Info_t odo_data; /**< $PSTMODO Data holder */
  Datalog_Info_t datalog_data; /**< $PSTMDATALOG Data holder */
  PSTMSETPAROK_Info_t setpark_status_data; /**< Result from PSTMSETPAR command */
  PSTMGETPAR_Info_t pstmgetpar_data; /**< $PSTMGETPAR Data holder */
  OpResult_t result; /**< Operation result/status */
} GNSSParser_Data_t;
```

1. ***ConsoleParseTask:*** *this thread is for receiving the user inputs from the serial terminal program. Based on the user selection, the respective data being stored in the NMEA data structure is displayed on the terminal using the respective display methods: GNSS_DATA_GetGPYYY() defined in the gnss_data.c*

The flow chart below illustrates the implementation of GetPos example.

Figure 7. GetPos flow diagram



The sample application also includes three advanced features supported by the Teseo-LIV3F/Teseo-VIC3DA devices. These features can be enabled by the X-CUBE-GNSS GUI provided as part of the STM32CubeMX program.

- Open the project file, *.ioc in STM32 CubeMX (or CubeIDE)and navigate to X-CUBE-GNSS1 component. Under "Parameter Settings" enable the special functions, Geofencing, Odometer or Data logging".
- **Geofencing**: allows the Teseo-LIV3F/Teseo-VIC3DA receiver to report the status of Geofence based on a preconfigured circle.
- **Odometer**: provides information on the traveled distance using GNSS position.
- **Data logging**: allows the Teseo-LIV3F receiver to save the resolved GNSS position on the local flash memory to be retrieved on demand from the host.

The Teseo-LIV3F/Teseo-VIC3DA device sends the received GNSS data via UART (or I²C) to the STM32 microcontroller on the STM32 Nucleo board according to the NMEA 0183 version 4.0 protocol.

At compile time, the user can choose the interface to receive data (setting the macro-USE_I2C in the file gnss1a1_conf.h) and also enable a specific feature (setting the corresponding macro in the file teseo_liv3f_conf.h).

The following describes the various options:

Table 7. Options

Option	Option name	Feature
1	getpos	Displays latitude, longitude, altitude information for the current epoch. This information is from $GPGGA or $GNGGA message
2	lastpos	Displays latitude, longitude, altitude information from the previous epoch. This information is from $GPGGA or $GNGGA message
3	wakestatus	Reports the status of EXTINT pin
4	help	Description of the various options
5	debug	Enables/disables debug messages
6	track	Displays location, time, and DOP based for various epochs. The number of epochs to be tracked and the duration between each track are provided by the user
7	lasttrack	Displays location, time, and DOP for the last epoch
8	getfwver	Displays firmware version of the GNSS library in the Teseo module
9	getgnsmsg	Displays NMEA xxGNS message
10	getgpgst	Displays GPGST message
11	getgprmc	Displays GPRMC message
12	getgsamsg	Displays xxGSA message - shows satellites that are actively tracked
13	getgsvmsg	Displays xxGSV message - shows satellites that are actively in view
14	en-feature	Enables special features like Geofence, Odometer, or Datalog
15	conf-geofence	Configure Geofence circle. The current example provides two Geofence circle
16	req-geofence	Request status of Geofence
17	odometer-op	Sends command to start and stop odometer
18	datalog-op	Sends command to start and stop datalog
19	ext-help	Prints this extended help
20	set-cdb	Command to set the CDB Register
21	get-cdb	Command to get the CDB Register
22	save-cdb	Command to save the CDB Register
23	reset	Command for system reset
24	cdb-factoryreset	Command to restore factory setting parameters

The NMEA messages [8][9][10] that are parsed are xxGGA, xxGGA, xxGNS, xxRST and xxRMC, xxGSA and xxGSV. The definitions on the various data fields associated with the NMEA messages can be found in the software user manual for the respective module provided in the reference.

The result of each NMEA message parsed are contained in the NMEA data structures as shown below.

Table 8. Data structure and their information

Data structure	Information
Coords_t	Latitude and longitude information
Geoid_Info_t	Geoid
UTC_Info_t	Universal Time Coordinated (UTC)
GSV_SAT_Info_t	Satellites in view information
GPGGA_Info_t	Information from GPGGA or GNGGA NMEA message
GNS_Info_t	Multi-constellation fix information
GPGST_Info_t	Pseudo range and noise statistics
GSA_Info_t	Information of satellites in track
GSV_Info_t	Information on satellites in view
pstmver_data	Firmware information of GNSS module

All these data structures are defined in: Middlewares\ST\lib_gnss\LibNMEA\Inc\ NMEA_parser.h

Result:

As soon as the board is powered up, the menu options in the terminal window appears.

Figure 8. GetPos: menu

Select from the different options to obtain human readable information related to the acquired GNSS position, the satellites in view, the active satellites, and so on.

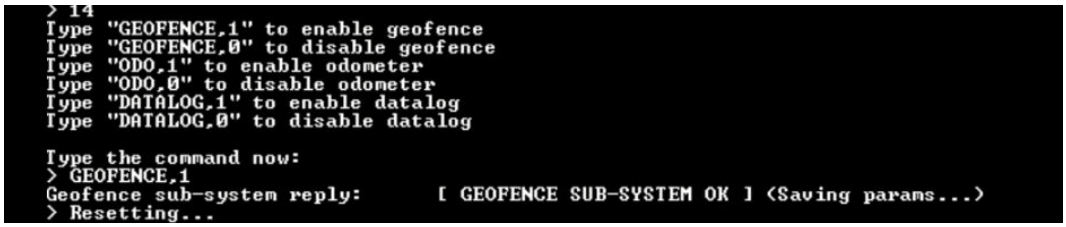
The figure below relates to menu selection option 1. Similarly, other options can be entered to get the Populated Data Fields.

Figure 9. GetPos: menu option 1

Geofence:

This feature enables the user to enable, config, and get the status of Geo-fencing. The commands below are used for this feature:

Figure 10. #14



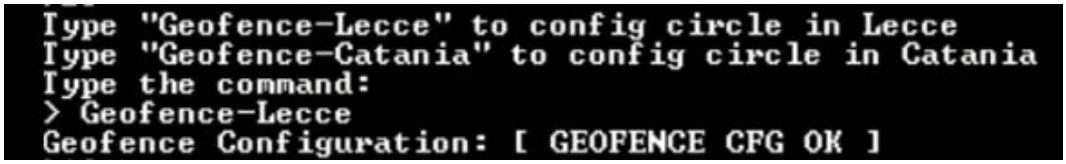
Type "GEOFENCE,1" to enable geofence:

NMEA message: $PSTMCFGGEOFENCE

Type "GEOFENCE,0" to disable geofence:

NMEA message: $PSTMCFGGEOFENCE

Figure 11. **#15**



Type "Geofence-Lecce" to config circle in Lecce.

Type "Geofence-Catania" to config circle in Catania.

Geofence-yyy is a Geo-fence configuration structure as shown below. In the GetPos example, there are two circles that are provided by default, Geofence_STLecce, Gefence_STCatania that are defined in gnss_feature_cfg_data.h. The user can modify it for a given location by modifying latitude, longitude, and radius (in meters).

GNSSGeofence_t Geofence_STLecce = {

.id = 5,

.enabled = 1,

.tolerance = 1,

.lat = 40.336055,

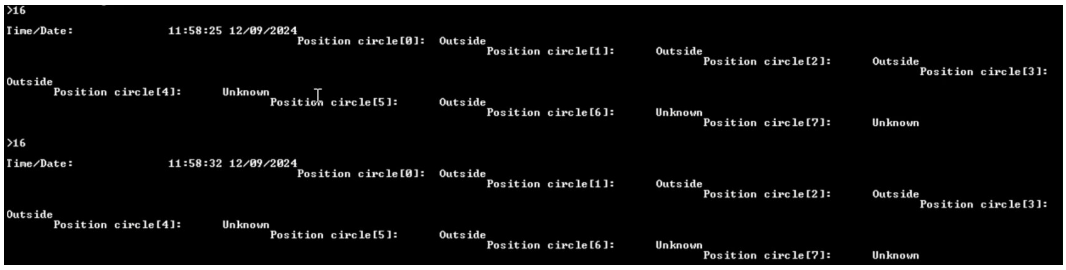
.lon = 18.120611,

.radius = 200,

.status = 0

};

Figure 12. **16 : req-geofence**



Once the Geofence is enabled and configured, the Geofence status can be retrieved by sending the below command:

It returns the following status based on the position of the moving body wrt the Geo Fence Circle configured above.

"Unknown", “Outside", "Boundary", "Inside"

NMEA message: PSTMGEOFENCEREQ and Status Message Parsed PSTMCFGGEOFENCE for Error, OK, and Status

Odometer:

The odometer provides information on the traveled distance using only positioning information. This feature is used to enable the odometer feature.

Type "ODO,1" to enable odometer.

NMEA message: $PSTMODOSTART

Type "ODO,0" to disable odometer.

NMEA message: $PSTMODOSTOP

To get the odometer status, the command below is sent.

NMEA message: $PSTMODOREQ

Datalog:

Data logging allows the GNSS Teseo module to save locally to the flash the resolved GNSS position to be retrieved on demand from the host. This feature can be enabled, disabled, and erased.

Type "DATALOG,1" to enable datalog.

NMEA message: $PSTMCFGLOG with Enable Data Log = 1

Type "DATALOG,0" to disable datalog.

NMEA message: $PSTMCFGLOG with Enable Data Log = 0

Type "CONFIG-DATALOG" to config datalog.

NMEA message: $PSTMLOGCREATE

Type "START-DATALOG" to start datalog.

NMEA message: $PSTMLOGSTART

Type "STOP-DATALOG" to stop datalog.

NMEA message: $PSTMLOGSTOP

Type "ERASE-DATALOG" to erase datalog.

NMEA message: $PSTMLOGERASE

Configuration Data Block (CDB) Register commands:

The GNSS software utilizes a CDB that holds the working parameters for the system. The parameters can be set, read or stored (in NVM) using the system configuration commands:

$PSTMSETPAR, $PSTMGETPAR and $PSTMSAVEPAR.

There is also a command to restore the factory setting parameters:

$PSTMRESTOREPAR.

NMEA message: $PSTMSETPAR will set/write the parameters of CDB.

NMEA message: $ PSTMGETPAR will read the parameters of CDB.

NMEA message: $ PSTMSAVEPAR will save the parameters of CDB.

NMEA message: $ PSTMRESTOREPAR will factory reset the parameters of CDB.

2.5.3 SimOSGetPos

RTOS:

NA

STM32:

NUCLEO-L073RZ

NUCLEO-L476RG

NUCLEO-F401RE

NUCLEO-H563ZI

NUCLEO-U575ZI-Q

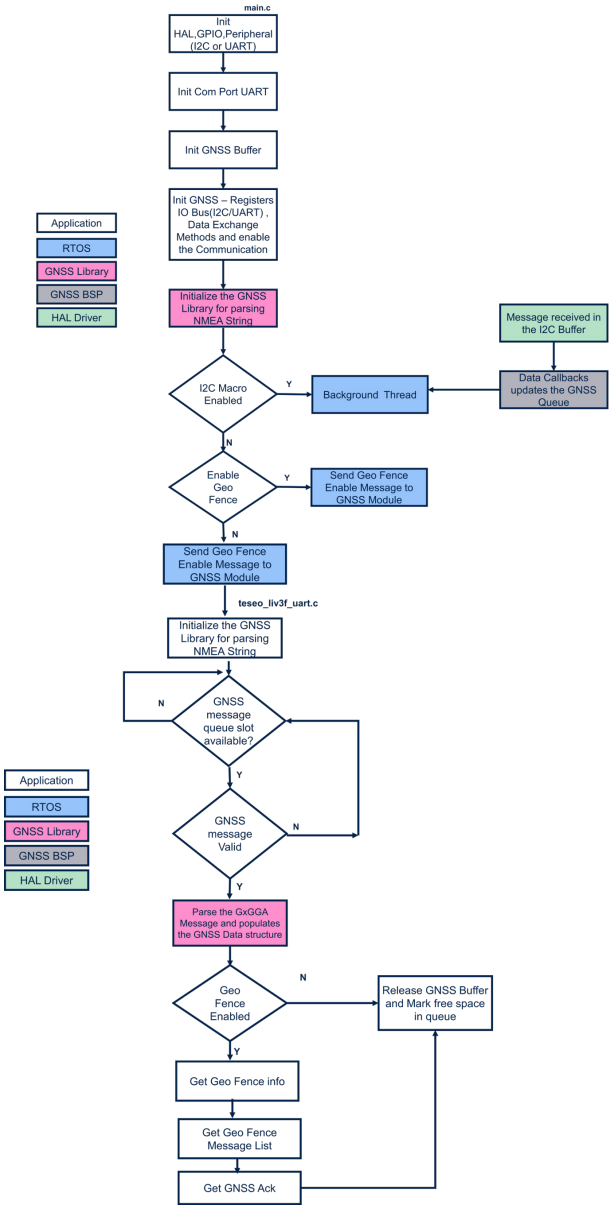
Description:

NMEA data is continuously read over I^2C or UART from the Teseo module. The application parses the NMEA data from the GPGGA or GNGGA sentence and it is parsed into integer values and displayed on the terminal console.

Open the project file, *.ioc in STM32 CubeMX (or CubeIDE)and navigate to X-CUBE-GNSS1 component. Under "Parameter Settings" enable the special function, Geofencing.

The geofence coordinates are configured for ST-Lecce. The coordinate information is found in gnss_feature_cfg_data.h. The user can manually edit this file to include the coordinates that needs to be geofenced.

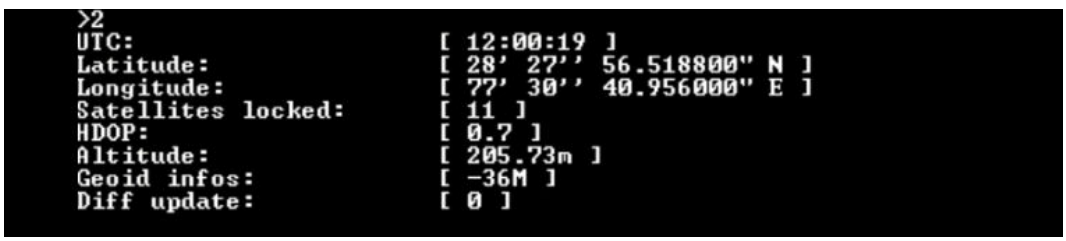
Figure 13. SimOS flow diagram



Result:

As soon as the board is powered up and flashed, GNSS NMEA data starts printing on the screen.

Figure 14. GNSS NMEA data on output screen



This data structures are defined in: Middlewares\ST\lib_gnss\LibNMEA\Inc\ NMEA_parser.h.

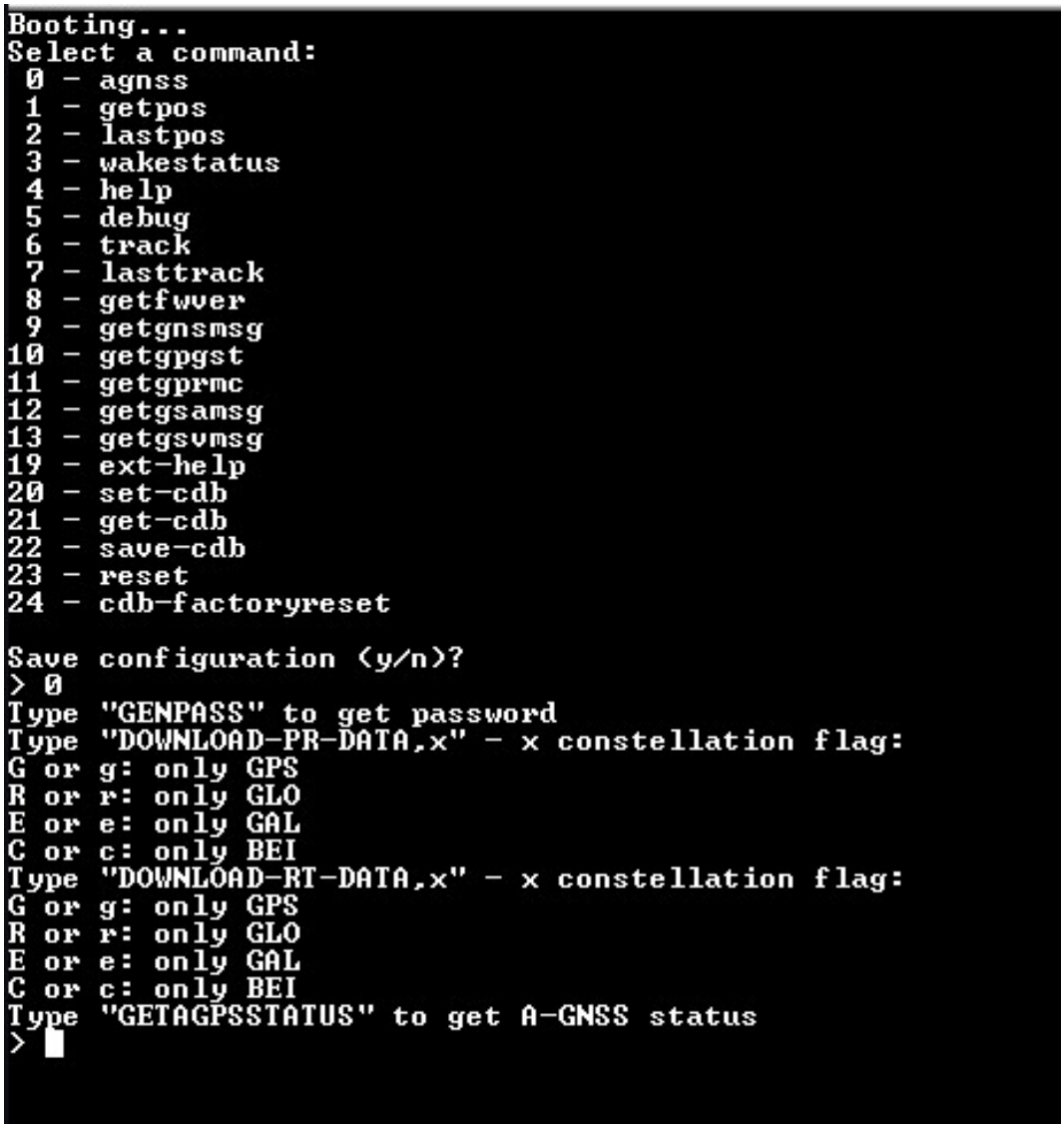
2.5.4 A_GetPos

A_GetPos is an extension of the GetPos application including the Assisted GNSS feature (both predictive and real-time) supported by the Teseo-LIV3F /Teseo-VIC3DA GNSS device.

The application has to access the Internet and is customized for the B-L475E-IOT01A discovery kit for IoT node providing the Wi-Fi connectivity (for further details on Assisted GNSS, refer to AN5160 and Section 3 of UM2399).

This application can be found in the package folder ROOT_DIR\Projects\B-L475-IOT01\Applications\A_GetPos.

Figure 15. A_GetPos menu option 0 and related sub-options



2.5.5 Firmware updater

RTOS:

NA

STM32:

NUCLEO-F401RE

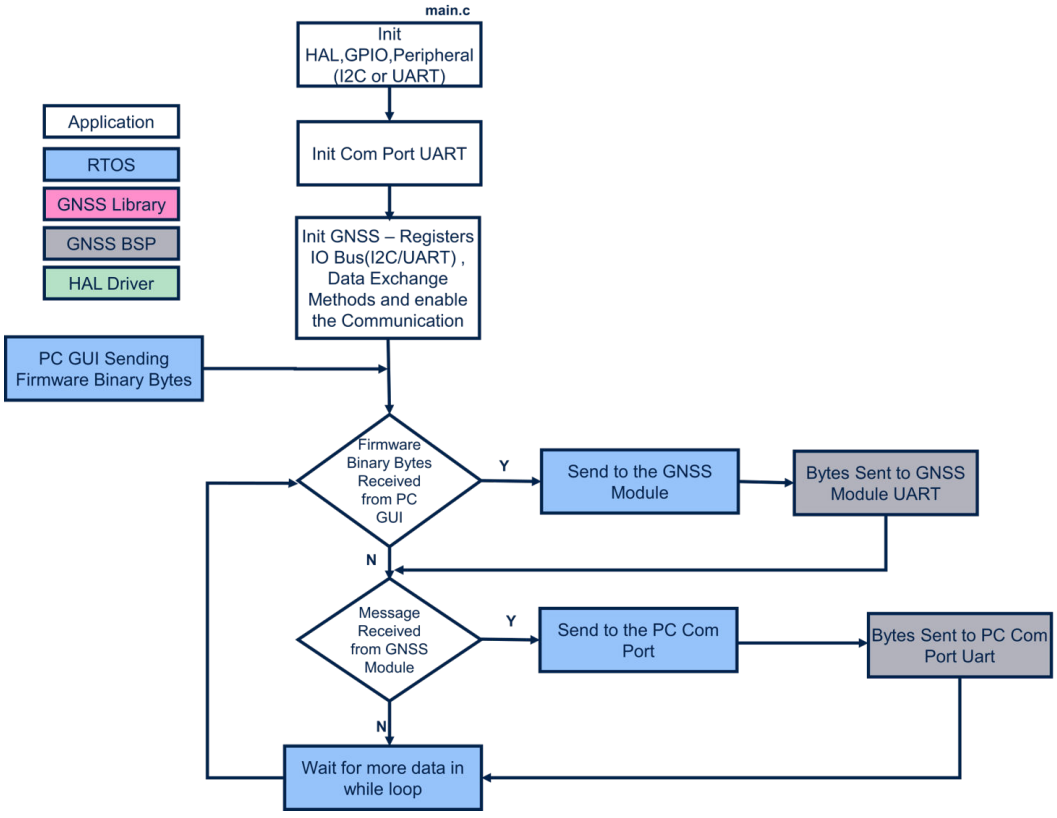
Description:

This example is used to flash the GNSS module present in the X-NUCLEO-GNSS1 board. The STM32 acts as a bridge between the PC and the module. Firmware utility runs on the PC and sends the firmware data over UART to the STM32 which then sends it to the GNSS module. The firmware update example running on the STM32 does not implement the handshaking needed between the bootloader and the Teseo module; the handshaking is instead implemented by the firmware upgrade Java utility mentioned in Section 2.6: GNSS firmware upgrader utility.

This example only applies to the X-NUCLEO-GNSS1A1 and X-NUCLEO-GNSS2A1 boards.

The flow chart below illustrates the implementation of a firmware update example.

Figure 16. Firmware updater flow diagram



Result:

The Teseo-LIV3F module is flashed using Teseo-Suite once the board is flashed. See the section below for more details.

2.6 GNSS firmware upgrader utility

The X-CUBE-GNSS1 software expansion for STM32Cube contains a Java utility to upgrade the Teseo-LIV3F GNSS device firmware to the latest available version. This utility is available in the ROOT_DIR\Utilities\PC_Software folder.

Step 1. Connect the X-NUCLEO-GNSS1A1 expansion board to the STM32 Nucleo development board.

Step 2. Copy the FW_Updater firmware to the Nucleo drive. The firmware can be found in the ROOT_DIR\Projects\STM32F401RE-Nucleo\Applications\FW_Updater folder on your PC.

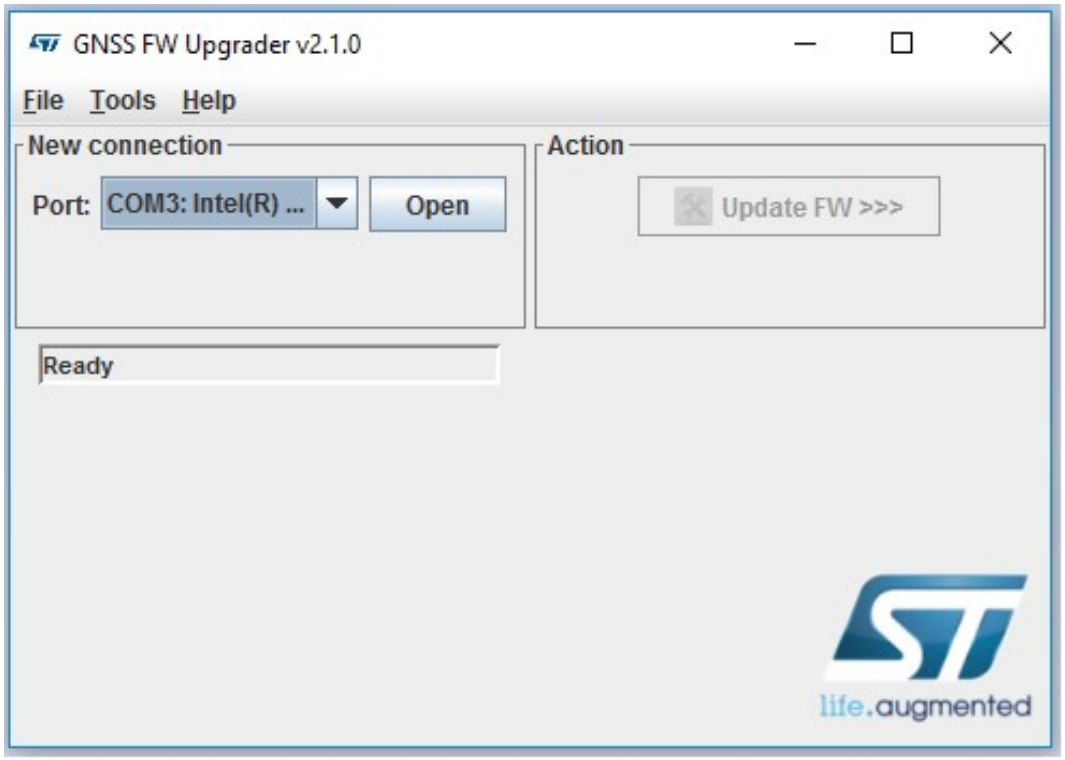
Step 3. Open the GNSS upgrader utility.

Step 4. Establish connection with the STM32 Nucleo board by selecting the serial port and clicking the Open button.

Step 5. A new firmware can also be loaded by clicking Tools > Advanced and pointing to the directory where the Teseo module binary is located.

*Note: Keep the [**Reset**] button on the STM32 Nucleo board pressed until the updating procedure starts.*

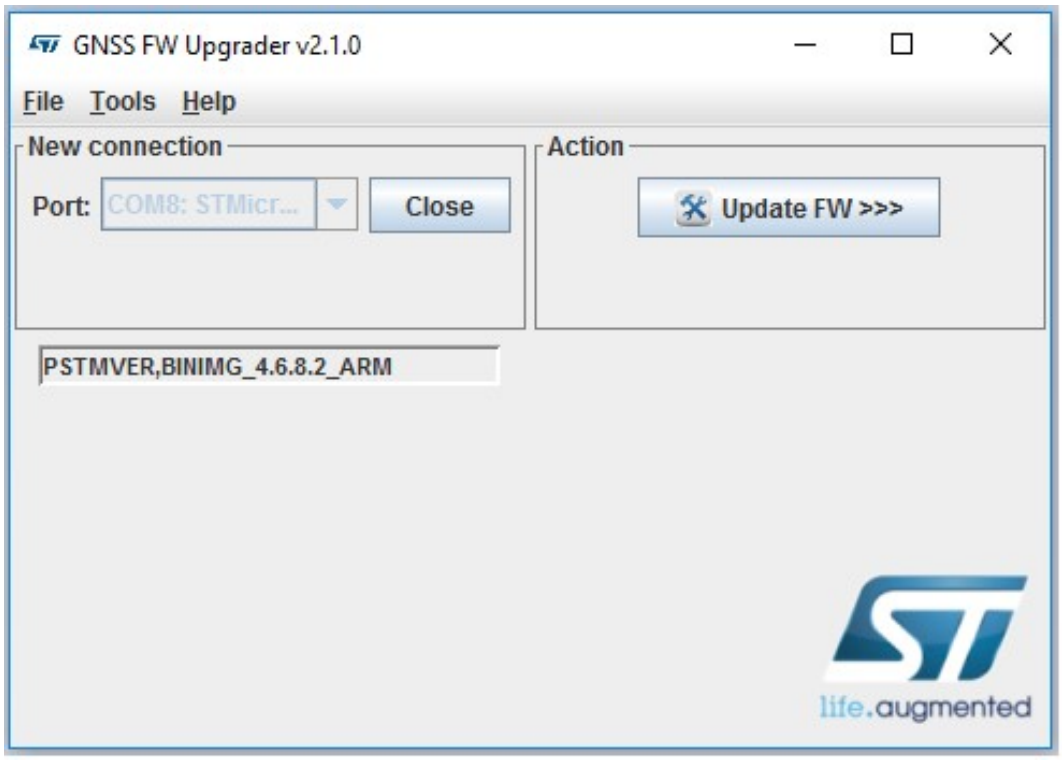
Figure 17. GNSS FW Upgrader: Open button



After a few seconds, the Teseo-LIV3F current firmware version is displayed in the GUI. A pop-up window indicates whether the firmware is the latest available version.

Step 6. If the current firmware is not the latest version, click on the **Update FW >>>** button to start the upgrade process.

Figure 18. GNSS FW Upgrader: upgrade process



Step 7. Wait for the upgrade to complete. A new pop-up should now confirm that the firmware is up to date. The GUI shows the new firmware version.

3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo

STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

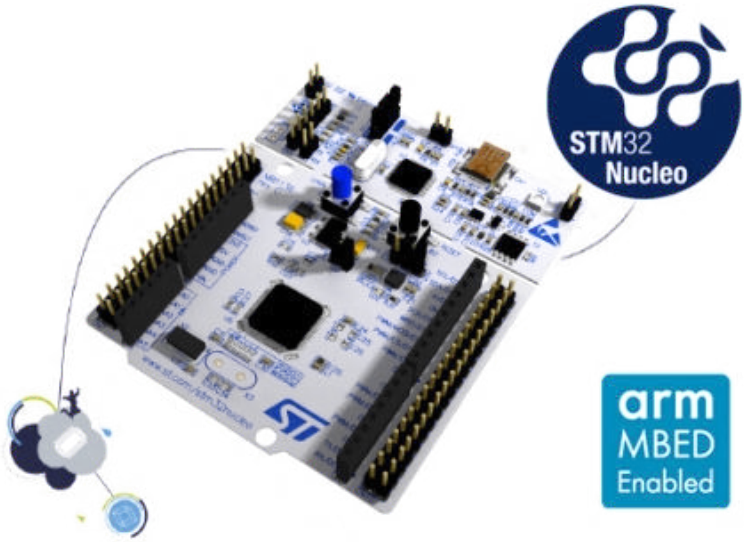
The Arduino connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples for different IDEs (IAR EWARM, Keil® MDK-ARM, STM32CubeIDE, Mbed™, and GCC/ LLVM).

All STM32 Nucleo users have free access to the Mbed™ online resources (compiler, C/C++ SDK, and developer community) at www.mbed.org to easily build complete applications.

Figure 19. STM32 Nucleo board



Hardware:

The following STM32 boards have been tested for the examples implemented below.

- **B-L475E-IOT01A1**: STM32L4 Discovery wireless low-power EVB with Bluetooth® Low Energy, NFC, sub-GHz, and Wi-Fi with an Arduino Uno V3 morpho connector.
- **NUCLEO-F401RE**: STM32 Nucleo-64 development board with STM32F401RE MCU, supports Arduino and ST morpho connectivity.
- **NUCLEO-L073RZ**: STM32 Nucleo-64 development board with STM32L073RZ MCU, supports Arduino and ST morpho connectivity.
- **NUCLEO-L476RG**: STM32 Nucleo-64 development board with STM32L476RG MCU, supports Arduino and ST morpho connectivity.
- **NUCLEO-U575ZI**: STM32 Nucleo-144 development board with STM32U575ZIT6Q MCU, SMPS, supports Arduino, ST Zio and morpho connectivity.
- **NUCLEO-H563ZI**: STM32 Nucleo-144 development board with STM32H563ZI MCU, supports Arduino, ST Zio and morpho connectivity.

GNSS device:

The following X-NUCLEO-GNSSx boards are compatible with the headers on STM32 evaluation boards.

X-NUCLEO-GNSS1A1: Teseo-LIV3F module.

X-NUCLEO-GNSS2A1: Teseo-VIC3DA module with integrated IMU for dead reckoning.

X-NUCLEO-LIV4A1: Teseo-LIV4F dual band L1 + L5 module.

Firmware:

X-CUBE-GNSS1

Utilities:

Teseo-Suite

Tera Term (or any serial terminal utility)

Note: Refer to Section 2.1 for appropriate jumper settings for the corresponding NUCLEO boards supported.

3.1.2 X-NUCLEO-GNSS1A1 expansion board

The X-NUCLEO-GNSS1A1 expansion board is based on the Teseo-LIV3F tiny GNSS module.

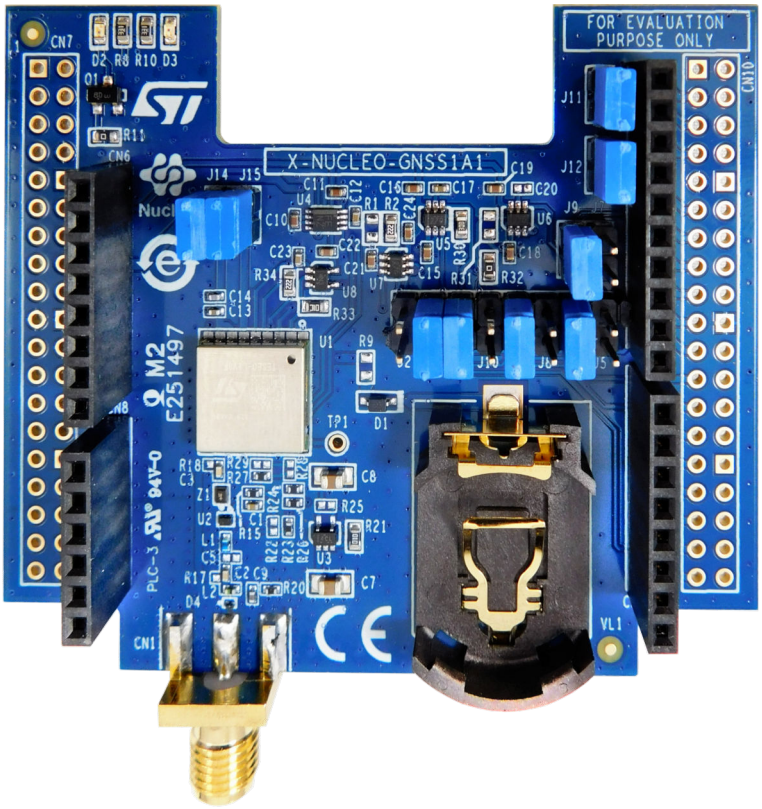
It represents an affordable, easy-to-use, global navigation satellite system (GNSS) module, embedding a TeseoIII single die standalone positioning receiver IC, usable in different configurations in your STM32 Nucleo project.

The Teseo-LIV3F is a compact (9.7x10.1 mm) module that provides superior accuracy thanks to the onboard 26 MHz temperature-compensated crystal oscillator (TCXO) and a reduced time-to-first fix (TTFF) with its dedicated 32 kHz real-time clock (RTC) oscillator.

The Teseo-LIV3F module runs the GNSS firmware (X-CUBE-GNSS1) to perform all GNSS operations including acquisition, tracking, navigation, and data output without external memory support.

The X-NUCLEO-GNSS1A1 expansion board is compatible with the Arduino UNO R3 connector and the ST morpho connector, so it can be plugged to the STM32 Nucleo development board and stacked with additional STM32 Nucleo expansion boards.

Figure 20. X-NUCLEO-GNSS1A1 expansion board



The following table shows the default jumper configuration for the X-NUCLEO-GNSS1A1 board (LIV3F).

Table 9. **X-NUCLEO-GNSS1A1 jumper default settings**

Signal	Arduino connector	STM32 Nucleo-64	Jumper	Configuration
I2C-SCL	D15	PB8	J11	Closed
I2C-SDA	D14	PB9	J12	Closed
Wakeup	D13	PA5	J9	Closed
Wakeup	D4	PB5	J7	Open
Reset	D9	PC7	J10	Open
Reset	D7	PA8	J13	Closed
PPS	D6	PB10	J6	Closed
PPS	D2	PA10	J8	Open
UART-RX	D8	PA9	J3 **	Closed
UART-TX	D2	PA10	J4	Closed
UART-RX	D1	PA2	J2 **	Open
UART-TX	D0	PA3	J5	Open

3.1.3 X-NUCLEO-GNSS2A1 expansion board

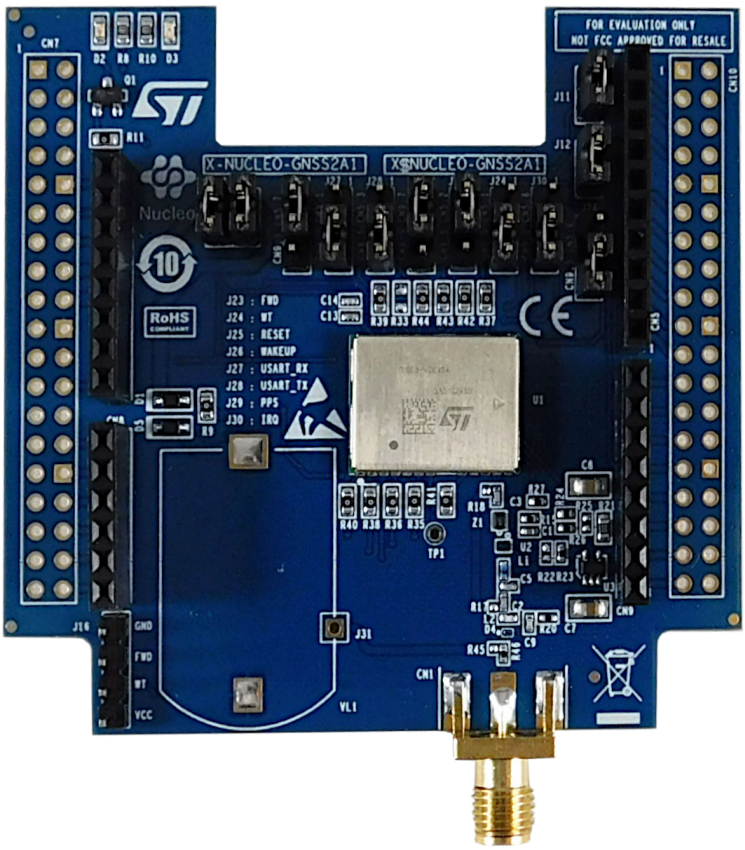
The X-NUCLEO-GNSS2A1 expansion board is based on the Teseo-VIC3DA tiny GNSS module. It represents an affordable, easy-to-use, global navigation satellite system (GNSS) module, which embeds a TeseoIII single die standalone positioning receiver IC, usable in different configurations in your STM32 Nucleo project.

The Teseo-VIC3DA is a compact (16.0 x 12.2 mm) module that provides superior accuracy thanks to the onboard temperature-compensated crystal oscillator (TCXO) and a reduced time-to-first fix (TTFF) with its dedicated real-time clock (RTC) oscillator.

The Teseo-VIC3DA module runs the GNSS firmware (X-CUBE-GNSS1) to perform all the GNSS operations including acquisition, tracking, navigation, and data output without any external memory support.

The X-NUCLEO-GNSS2A1 expansion board is compatible with the Arduino UNO R3 connector and the ST morpho connector. It can be plugged to an STM32 Nucleo development board and stacked with additional STM32 Nucleo expansion boards.

Figure 21. X-NUCLEO-GNSS2A1 expansion board



The following table shows the default jumper configuration for the X-NUCLEO-GNSS2A1 board (Teseo-VIC3DA).

Table 10. X-NUCLEO-GNSS2A1 jumper default settings

Signal	Arduino connector	STM32 Nucleo 64	Jumper	Configuration
I2C-SCL	D15	PB8	J11	Closed
I2C-SDA	D14	PB9	J12	Closed
VCC-VCC_IO			J14	Closed
V14Bat			J15	Closed
SYS_FWD	D4	PB5	J23	1-2
SYS_WHEELTICK	D8	PA9	J24	2-3
SYS_RESETn	D7	PA8	J25	1-2
SYS_WAKEUP	D13	PB5	J26	2-3
UART-RX	D8	PA9	J27 (1)	2-3
UART-TX	D2	PA10	J28(1)	2-3
SYS_PPS	D6	PB10	J29	1-2
SYS_IRQ	D9	PC7	J30	2-3

1. *Refer to Section 3.1: Hardware description for jumper setting for specific STM32 board.*

3.1.4 X-NUCLEO-LIV4A1 expansion board

The X-NUCLEO-LIV4A1 expansion board is based on the Teseo-LIV4F tiny dualband GNSS low-power and measurement engine modules.

It represents an affordable, easy-to-use, global navigation satellite system (GNSS) module, embedding the TESEO IV single die GNSS receiver IC working simultaneously on multiple constellations (GPS/Galileo/Glonass/ BeiDou/ QZSS) able to provide positioning (Teseo-LIV4F) in your STM32 Nucleo project.

The Teseo-LIV4F is a compact (9.7 x 10.1 mm) module that provides superior accuracy thanks to the onboard temperature-compensated crystal oscillator (TCXO) and a reduced time-to-first fix (TTFF) with its dedicated real-time clock (RTC) oscillator.

The Teseo-LIV4F module runs the GNSS firmware (X-CUBE-GNSS1) to perform all GNSS operations including acquisition, tracking, navigation, and data output without external memory support.

The X-NUCLEO-LIV4A1 expansion board is compatible with the Arduino UNO R3 connector and the ST morpho connector, so it can be plugged to the STM32 Nucleo development board and stacked with additional STM32 Nucleo expansion boards.

Figure 22. **X-NUCLEO-LIV4A1 expansion board**

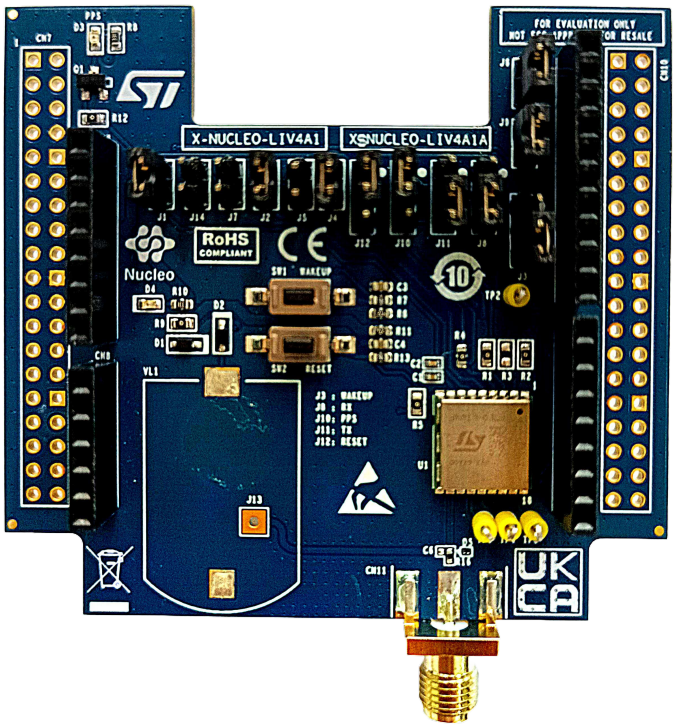
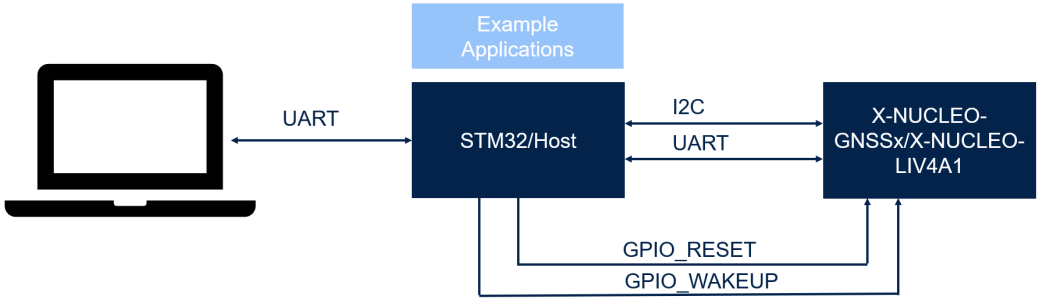


Table 11. **X-NUCLEO-LIV4A1 jumper default settings**

Signal	Arduino connector	STM32 Nucleo 64	Jumper	Configuration
I2C-SCL	D15	PB8	J6	Closed
I2C-SDA	D14	PB9	J9	Closed
VCC			J1	Closed
VBATT			J2	Closed
SYS_RESETn	D7	PA8	J12	1-2
SYS_WAKEUP	D13	PA5	J3	2-3
UART-RX	D0	PA2	J8	2-3
UART-TX	D1	PA3	J11	2-3
SYS_PPS	D6	PB10	J10	1-2
PRG			J4	Closed

3.2 Software setup

Figure 23. **X-CUBE-GNSS1 block diagram**



The following software components are required to set up a suitable development environment for creating applications for the STM32 Nucleo equipped with the X-NUCLEO-GNSS1A1, X-NUCLEO-GNSS2A1, or XNUCLEO-LIV4A1 expansion board:

- X-CUBE-GNSS1: an expansion for STM32Cube dedicated to GNSS applications development. The X-CUBE-GNSS1 firmware and related documentation is available on www.st.com.
- Development toolchain and Compiler: the STM32Cube expansion software supports the following environments:
  - IAR Embedded Workbench for Arm (IAR-EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM-STM32) toolchain + ST-LINK - STM32CubeIDE + ST-LINK

3.3 System setup guide

This section describes how to set up different hardware parts before developing and executing an application on the STM32 Nucleo board with the GNSS expansion board.

3.3.1 STM32 Nucleo and GNSS expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer.

The appropriate ST-LINK/V2-1 USB driver (STSW-LINK009) can be downloaded from www.st.com.

The GNSS expansion board X-NUCLEO-GNSS1A1, X-NUCLEO-GNSS2A1, or X-NUCLEO-LIV4A1 is easily connected to the STM32 Nucleo development through the Arduino UNO R3 extension connector and can interface with the external STM32 microcontroller on the STM32 Nucleo board either via UART or inter-integrated circuit (I²C) channels.

Figure 24. X-NUCLEO-GNSS2A1 expansion board connected to an STM32 Nucleo board

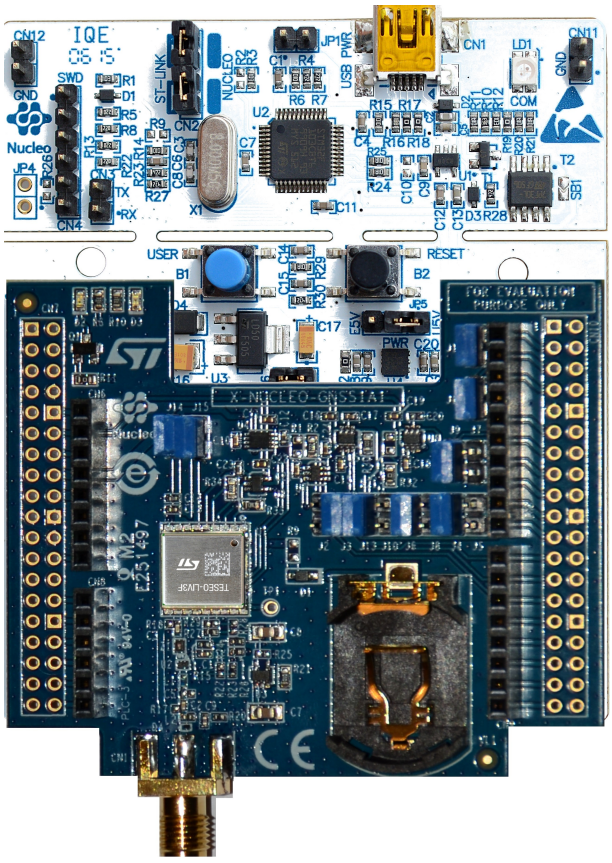
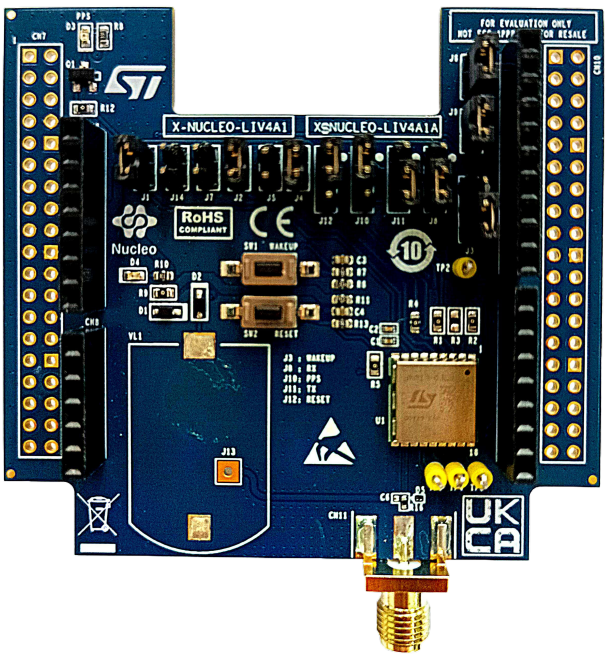


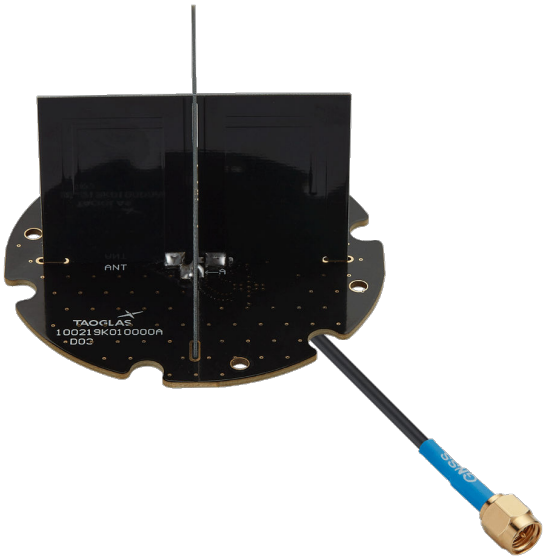
Figure 25. **X-NUCLEO-GNSS2A1 expansion board connected to an STM32 Nucleo board**

Figure 26. **X-NUCLEO-LIV4A1 expansion board**



The GPS/GLONASS/Beidou antenna included with the X-NUCLEO-GNSS1A1 or X-NUCLEO-GNSS2A1 expansion board must be connected to the antenna connector on the expansion board.

Figure 27. **X-NUCLEO-LIV4A1 expansion board**



Version needed when connecting X-NUCLEO-GNSSx module with a specific STM32 Nucleo board. Refer to Section 3.1: Hardware description for configuring the X-NUCLEO-GNSSxxx.

4 GNSS firmware upgrader utility setup

The GNSS firmware upgrader Java utility included in the software package is a graphical user interface that can be used to upgrade the Teseo-LIV3F on the X-NUCLEO-GNSS1A1 to the latest firmware version.

It has no particular minimum requirements; you can use any Linux, Microsoft, or OSX PC with:

- Java Runtime Environment (JRE) 6+ • 1 x USB port.

To use the firmware updater Java utility, make sure you have correctly set up your hardware and software.

The utility can be launched by simply clicking twice on the FWUPG.jar file, located in the “Utilities\PC_Software\FirmwareUpdaterTool” folder.

5 Known Limitations and Workaround

- For the GetPos application to properly work 15KB RAM is the minimum requirement.
- The following Preprocessor macros are required:
  - If using I2C, USE_HAL_I2C_REGISTER_CALLBACKS=1
    (GetPos, Virtual_COM_Port and SimOSGetPos applications).
  - When using UART, USE_HAL_UART_REGISTER_CALLBACKS=1
    (GetPos, Virtual_COM_Port and SimOSGetPos applications).
    When configuring a project from scratch, the preprocessor macros are enabled in STM32CubeMX GUI when configuring I2C/UART peripheral. This is described in Getting Started Guide in X-CUBE-GNSS installation repository.
- For STM32CubeIDE projects, the floating point for the printf function must be enabled.
  From the menu bar select: Project > Properties > C/C++ Build > Settings > Tool Settings > MCU Settings and then check Use float with print from newlib-nano.
- The FWUpdater application works only using the UART channel and tested with X-CUBE-GNSS1A1.
- For MDK-ARM projects, the Use MicroLib option in the project settings must be manually set.
- On dual-core STM32 series this expansion software can be used on both cores but exclusive.

6 References

[1] UM2334 – User Manual - *Getting started with the X-CUBE-GNSS1 Global Navigation Satellite System software expansion for STM32Cube.*

[2] UM2327 – User Manual - *Getting started with the X-NUCLEO-GNSS1A1 expansion board based on Teseo-LIV3F tiny GNSS module for STM32 Nucleo.*

[3] UM3020 – User Manual - *Getting started with the X-NUCLEO-GNSS2A1 expansion board based on the Teseo-VIC3DA dead-reckoning GNSS module for STM32 Nucleo*

[4] UM3296 – *User Manual - Getting started with the X-NUCLEO-LIV4A1 expansion board based on the Teseo-LIV4F Tiny GNSS dual-bands low power module for STM32 Nucleo*

[5] Teseo-LIV3F Data Sheet

[6] Teseo-LIV4F Data Sheet

[7] Teseo-VIC3DA Data Sheet

[8] Teseo-LIV4F GNSS Module - Software manual

[9] Teseo-LIV3 GNSS Module - Software manual

[10] Teseo-VIC3 software instructions

[11] STM32CubeMX MOOC

Revision history

Table 12. **Document revision history**

Date	Version	Changes
06-Dec-2017	1	Initial release.
25-Jan-2018	2	Updated Figure 5 GetPos: menu. Added Section 3.6 Virtual_COM_Port sample application.
11-May-2018	3	Updated Introduction, Section 3.1 Overview, Section 3.2 Architecture, Section 3.3 Folder structure, Section 3.5 GetPos sample application. Added Section 3.5.2 SimOSGetPos.
15-Oct-2018	4	Updated Introduction, Table 1. List of acronyms, Section 3.1 Overview, Figure 2. X-CUBE-GNSS1 software architecture, Section 3.3 Folder structure, Figure 8. GNSS FW Upgrader: Open button, Figure 9. GNSS FW Upgrader: upgrade process, Figure 11. X-NUCLEO-GNSS1A1 expansion board and Figure 12. X-NUCLEO-GNSS1A1 expansion board connected to an STM32 Nucleo board. Added Section 3.5.3 A_GetPos.
23-Apr-2019	5	Updated Figure 2. X-CUBE-GNSS1 software architecture, Figure 3. X-CUBE-GNSS1 package folder structure, Section 3.5 GetPos sample application, Section 3.5.2 SimOSGetPos, Section 3.5.3 A_GetPos, Section 3.6 Virtual_COM_Port sample application and Section 3.7 GNSS FW Upgrader utility.
14-Apr-2022	6	Minor text changes. Removed Section 2 What is STM32Cube? and Section 2.1 STM32Cube architecture, and replaced them with a link in the introduction.
21-Jun-2022	7	Updated introduction, Section 1 Acronyms and abbreviations, Section 2.1 Overview, Section 2.2 Architecture, Section 2.3 Folder structure, Section 2.5 GetPos sample application, Section 2.5.3 SimOSGetPos, Section 2.6 Virtual_COM_Port sample application, Section 3.2 Hardware setup, Section 3.3 Software setup, and Section 3.4.1 STM32 Nucleo and GNSS expansion board setup. Added references to NUCLEO-U575ZI-Q development board.
25-Jul-2022	8	Updated introduction, Section 1 Acronyms and abbreviations, Section 2.1 Overview, Section 2.2 Architecture, Section 2.5 GetPos sample application, Section 3.3 Software setup and Section 3.4.1 STM32 Nucleo and GNSS expansion board setup. Added Section 3.1.3 X-NUCLEO-GNSS2A1 expansion board, and Section 3.2.1 X-NUCLEOGNSS2A1 setup. Added references to X-NUCLEO-GNSS2A1 expansion board, Teseo-VIC3DA GNSS module, and NUCLEO-U575ZI-Q development board.
19-Mar-2024	9	Added refernces to Teseo-LIV4F. Updated *Section Introduction, Section 2.1: Overview, Section 2.2: Architecture, Section 2.6: Virtual_COM_Port sample application, Section 3.2: Hardware setup* and *Figure 16. GPS/GLONASS/Beidou antenna.* Added *Section 3.1.4: X-NUCLEO-LIV4A1 and Section 3.2.3: X-NUCLEO-LIV4A1 setup.*
07-Nov-2024	10	Updated Section Introduction, Section 1: Acronyms and abbreviations, Section 2: X-CUBE-GNSS1 software expansion for STM32Cube, Section 3: System setup guide and Section 4: GNSS firmware upgrader utility setup.
08-Oct-2025	11	Updated Section Introduction, Section 1: Acronyms and abbreviations, Section 2.1: Overview, Section 2.3: Folder structure, Section 2.5.1: Virtual_COM_Port, Section 2.5.2: GetPos, Section 2.5.3: SimOSGetPos, Section 2.5.4: A_GetPos, Section 3.1.1: STM32 Nucleo, Section 3.1.4: X-NUCLEO-LIV4A1 expansion board and Section 5: Known Limitations and Workaround.

Contents

List of tables

List of figures