UM2334
User manual

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

Introduction

The X-CUBE-GNSS1 expansion package for STM32Cube runs on STM32 and includes drivers for the *Teseo-LIV3F* global navigation satellite system (GNSS) device as well as middleware for the NMEA protocol support.

It is built on top of STM32Cube software technology for easy portability across different STM32 microcontrollers.

The software comes with sample implementations for the drivers running on the *X-NUCLEO-GNSS1A1* expansion board, when connected to a *NUCLEO-F401RE* board.

Contents

List of tables

List of figures

1 Acronyms and abbreviations

Table 1: List of acronyms

Acronym	Description
BSP	Board support package
BeiDou	China’s regional navigation satellite system
Galileo	Europe’s 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 What is STM32Cube?

2.1 What is STM32Cube?

STMCube™ represents the STMicroelectronics initiative to make developers' lives easier by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

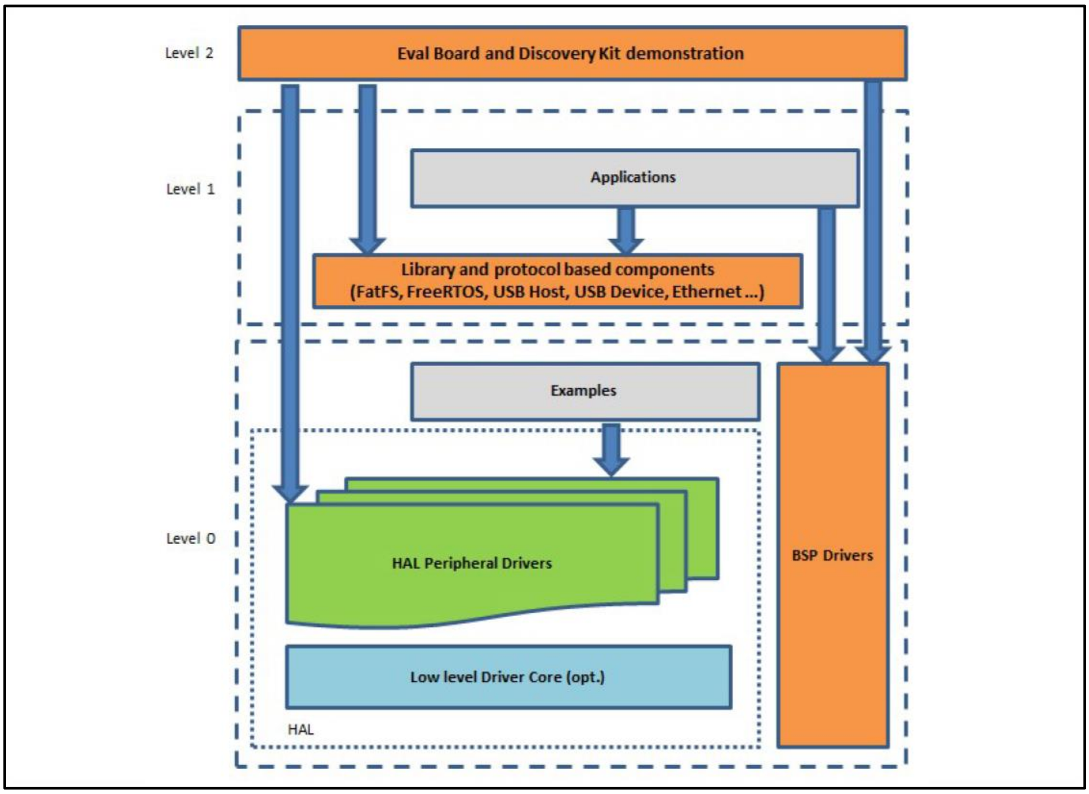
STM32Cube version 1.x includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32CubeF4 for the STM32F4 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer software, ensuring maximized portability across the STM32 portfolio
  - a consistent set of middleware components such as RTOS, USB, TCP/IP and graphics
  - all embedded software utilities with a full set of examples

2.2 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with one another, as described in the diagram below.

Figure 1: Firmware architecture


Level 2
Eval Board and Discovery Kit demonstration
Level 1
Applications
Library and protocol based components
(FatFS, FreeRTOS, USB Host, USB Device, Ethernet...)
Examples
Level 0
HAL Peripheral Drivers
BSP Drivers
Low level Driver Core (opt.)
HAL

Level 0: This level is divided into three sub-layers:

- Board Support Package (BSP): this layer offers a set of APIs relative to the hardware components in the hardware boards (Audio codec, IO expander, Touchscreen, SRAM driver, LCD drivers. etc…); it is based on modular architecture allowing it to be easily ported on any hardware by just implementing the low level routines. It is composed of two parts:

  - Component: is the driver relative to the external device on the board and not related to the STM32, the component driver provides specific APIs to the external components of the BSP driver, and can be ported on any other board.
  - BSP driver: links the component driver to a specific board and provides a set of easy to use APIs. The API naming convention is BSP_FUNCT_Action(): e.g., BSP_LED_Init(), BSP_LED_On().

- Hardware Abstraction Layer (HAL): this layer provides the low level drivers and the hardware interfacing methods to interact with the upper layers (application, libraries and stacks). It provides generic, multi-instance and function-oriented APIs to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I²C, UART, etc.) it provides APIs for peripheral initialization and configuration, data transfer management based on polling, interrupt or DMA processes, and communication error management. The HAL Drivers APIs are split in two categories: generic APIs providing common, generic functions to all the STM32 series and extension APIs which provide special, customized functions for a specific family or a specific part number.
- Basic peripheral usage examples: this layer houses the examples built around the STM32 peripherals using the HAL and BSP resources only.

Level 1: This level is divided into two sub-layers:

- Middleware components: set of libraries covering USB Host and Device Libraries, STemWin, FreeRTOS, FatFS, LwIP, and PolarSSL. Horizontal interaction among the components in this layer is performed directly by calling the feature APIs, while vertical interaction with low-level drivers is managed by specific callbacks and static macros implemented in the library system call interface. For example, FatFs implements the disk I/O driver to access a microSD drive or USB Mass Storage Class.
- Examples based on the middleware components: each middleware component comes with one or more examples (or applications) showing how to use it. Integration examples that use several middleware components are provided as well.

Level 2: This level is a single layer with a global, real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and basic peripheral usage applications for board-based functions.

3 X-CUBE-GNSS1 software expansion for STM32Cube

3.1 Overview

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

The key features of the package are:

- Complete software to build applications using Teseo-LIV3F GNSS device
- Middleware for the NMEA protocol support
- Sample application to transmit GNSS data to a PC
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

The package includes a sample application that the developer can use to start experimenting with the code. Java applications to update the *Teseo-LIV3F* and the *STM32 Nucleo* board firmware are included in the package.

3.2 Architecture

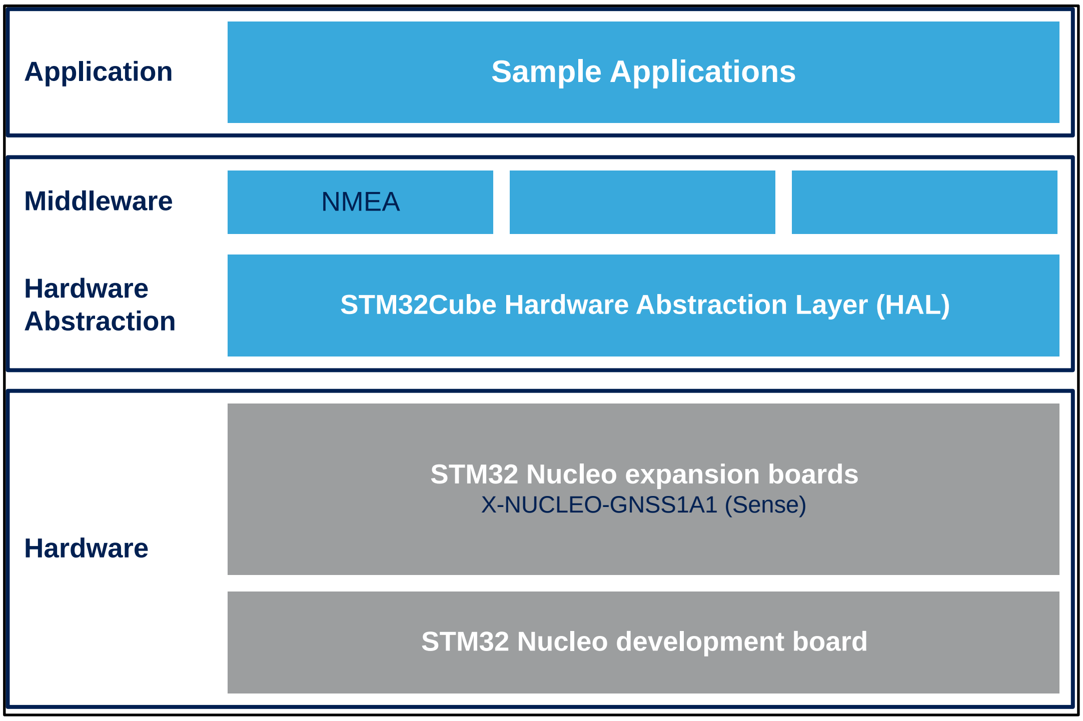
This software is a fully compliant expansion of *STM32Cube* architecture for the development of applications using the ST *Teseo-LIV3F* GNSS module.

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 some middleware components for NMEA protocol support.

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

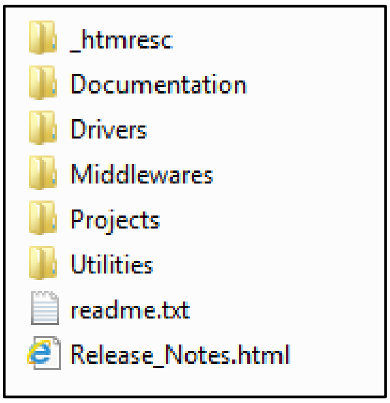
- **STM32Cube HAL layer**: consists of simple, generic and multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework so that overlying layers like middleware can function without requiring specific microcontroller unit (MCU) hardware information. This structure improves library code reusability and guarantees 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


Application
Sample Applications
Middleware
NMEA
Hardware Abstraction
STM32Cube Hardware Abstraction Layer (HAL)
Hardware
STM32 Nucleo expansion boards
X-NUCLEO-GNSS1A1 (Sense)
STM32 Nucleo development board

3.3 Folder structure

Figure 3: X-CUBE-GNSS1 package folder structure



The following folders are included in the software package:

- **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 on-board 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 (e.g., for NMEA message parsing).
- **Projects**: contains the sample applications used to access the GNSS data; they are provided for the *NUCLEO-F401RE* platforms under three development environments (IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM), and System Workbench for STM32 (*SW4STM32*).

- **Utilities**: contains a “PC_Software” folder with a Java utility to update the *Teseo-LIV3F* device firmware.

3.4 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.

3.5 GetPos sample application

An example application using the *X-NUCLEO-GNSS1A1* expansion board with a *NUCLEO-F401RE* board is included in the package (in folder ROOT_DIR\Projects\STM32F4xx-Nucleo\Applications\GetPos). Ready to be built projects are available for multiple IDEs.

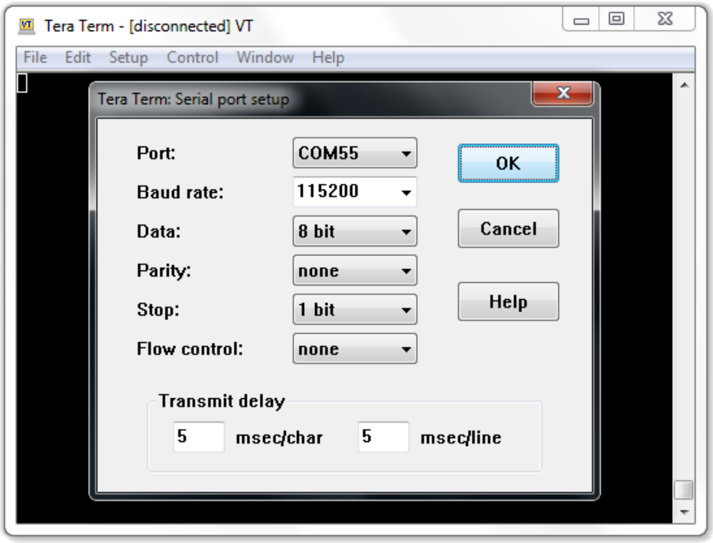
In this application, real time GNSS data received by the *Teseo-LIV3F* device can be displayed through a serial connection and a serial terminal on a PC.

The Teseo-LIV3F device sends the received GNSS data via UART to the STM32 microcontroller on the *STM32 Nucleo* board according to the NMEA 0183 Version 4.0 protocol.

3.5.1 GetPos usage procedure

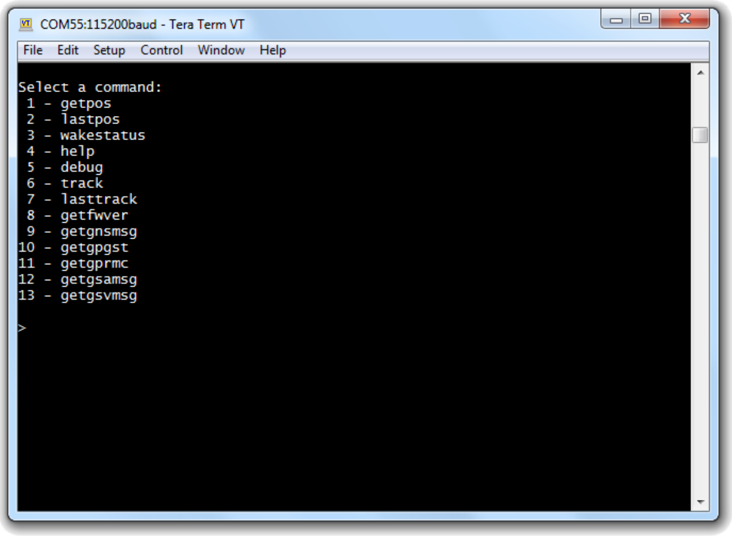
The following instructions show how to use the GetPos sample application to return information relating to a GNSS position.

1. Open a serial connection between the STM32 Nucleo and the X-NUCLEO-GNSS1A1 board and your PC with the correct COM port for your PC and the following remaining parameters:


Tera Term - [disconnected] VT
File Edit Setup Control Window Help
Tera Term: Serial port setup
OK
Cancel
Help

Port: COM55
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none
Transmit delay
5 msec/char 5 msec/line

2 Push the Reset button to show the menu options in the terminal window.

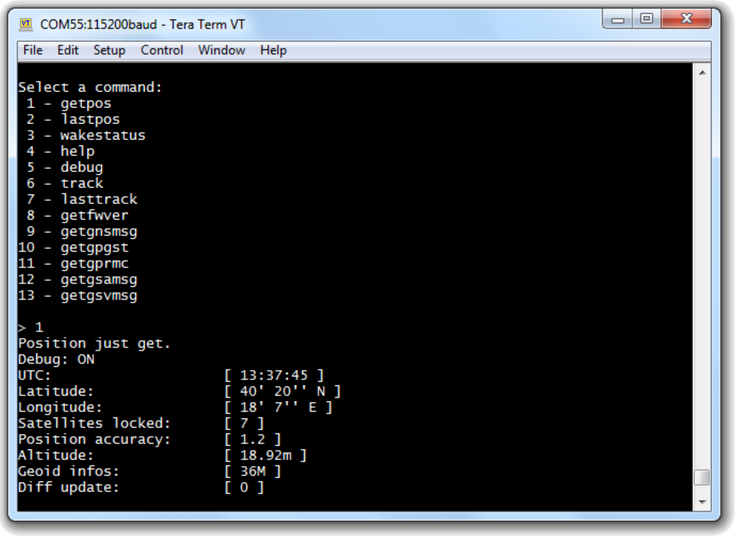

COM55:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Select a command:
 1 - getpos
 2 - lastpos
 3 - wakestatus
 4 - help
 5 - debug
 6 - track
 7 - lasttrack
 8 - getfwver
 9 - getgnsmsg
10 - getgpgst
11 - getgprmc
12 - getgsamsg
13 - getgsvmsg

>
```

3 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


COM55:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Select a command:
 1 - getpos
 2 - lastpos
 3 - wakestatus
 4 - help
 5 - debug
 6 - track
 7 - lasttrack
 8 - getfwver
 9 - getgnsmsg
10 - getgpgst
11 - getgprmc
12 - getgsamsg
13 - getgsvmsg

> 1
Position just get.
Debug: ON
UTC:                    [ 13:37:45 ]
Latitude:               [ 40' 20'' N ]
Longitude:              [ 18' 7'' E ]
Satellites locked:      [ 7 ]
Position accuracy:      [ 1.2 ]
Altitude:               [ 18.92m ]
Geoid infos:            [ 36M ]
Diff update:            [ 0 ]
```

3.6 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.

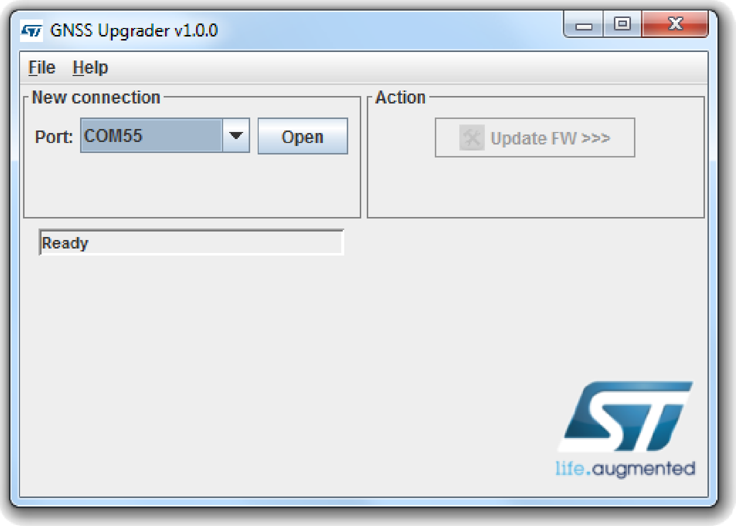
1 Connect the *X-NUCLEO-GNSS1A1* expansion board to the *STM32 Nucleo* development board.

2 Copy the Virtual_COM_Port firmware to the Nucleo drive.

You will find the firmware in the ROOT_DIR\Projects\STM32F4xx-Nucleo\Applications\Virtual_COM_Port folder on your PC.

3 Run the GNSS Upgrader utility

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

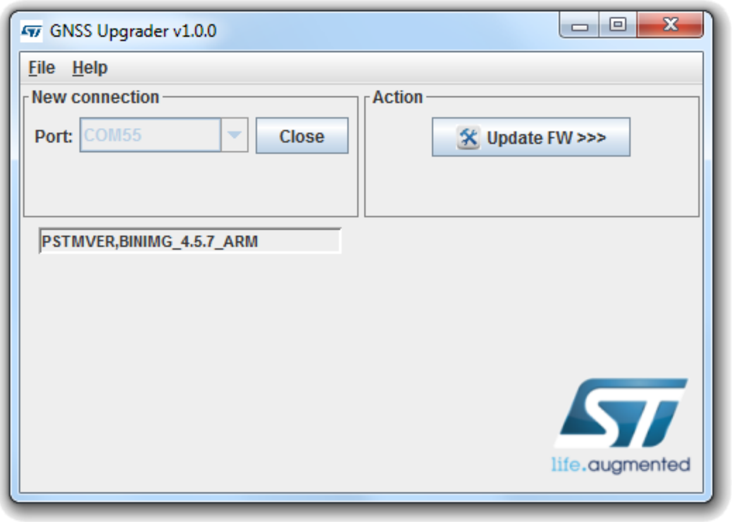


Ready

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.

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



PSTMVER,BINIMG_4.5.7_ARM

6 Wait for the upgrade to complete

A new pop-up should now confirm that the firmware is up to date.

The GUI will show the new firmware version.

3.7 System setup guide

3.7.1 Hardware description

3.7.1.1 STM32 Nucleo platform

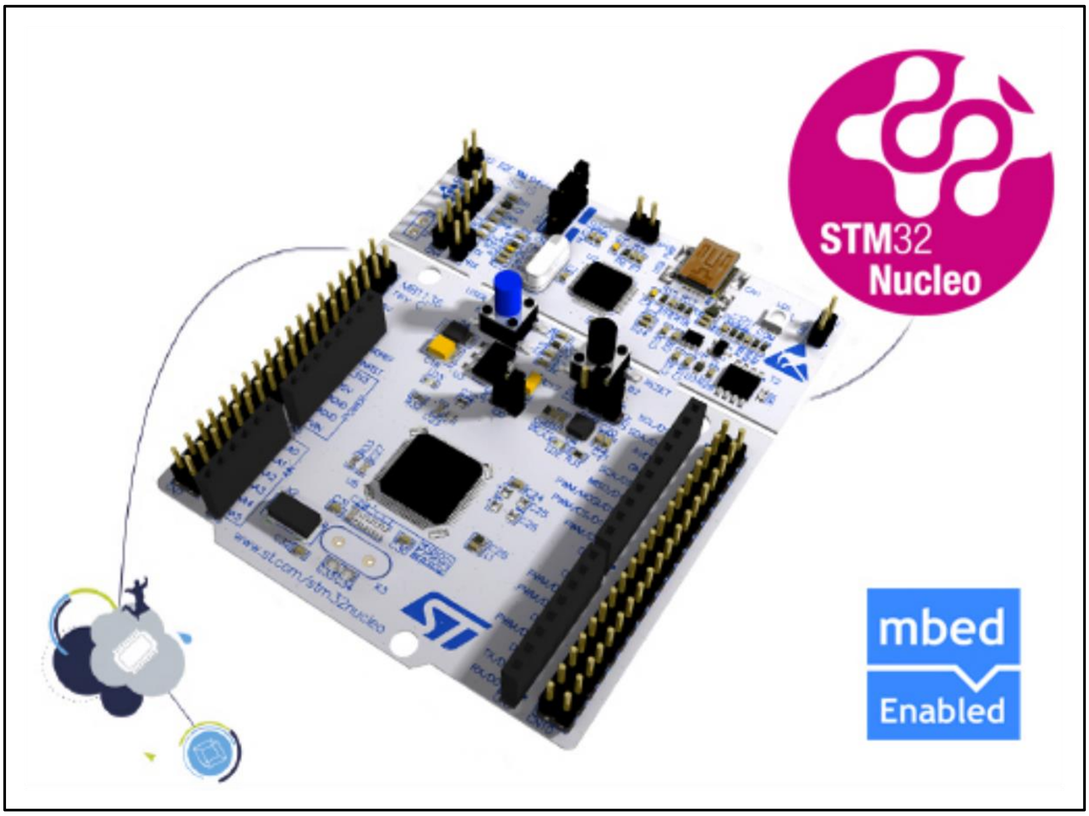
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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.

Figure 4: STM32 Nucleo board

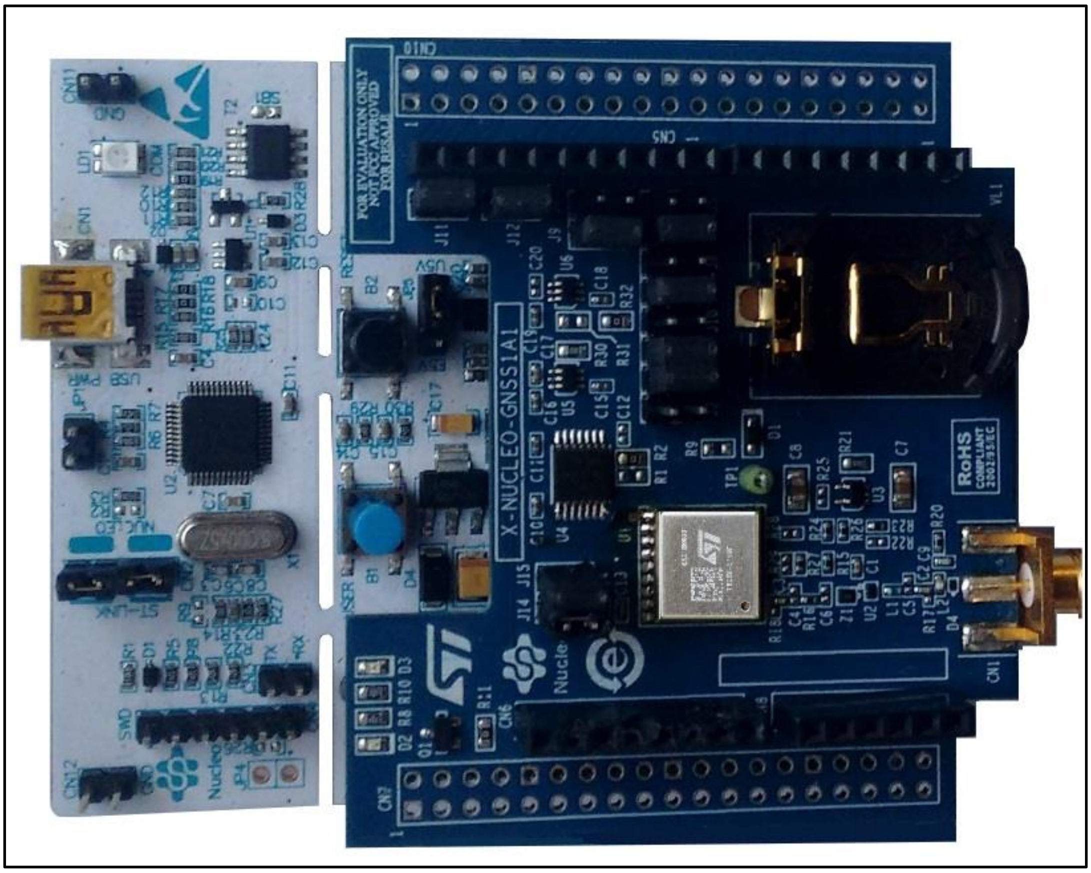


Information regarding the STM32 Nucleo board is available at *www.st.com/stm32nucleo*

3.7.1.2 X-NUCLEO-GNSS1A1 expansion board

The X-NUCLEO-GNSS1A1 must be plugged onto an *STM32 Nucleo* board through the Arduino™ UNO R3 connectors as shown in the figure below..

Figure 5: X-NUCLEO-GNSS1A1 expansion board connected to an STM32 Nucleo board



The X-NUCLEO-GNSS1A1 can be connected to any STM32 Nucleo board, even though complete testing has been performed on the *NUCLEO-F401RE* board.

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

Figure 6: GPS/GLONASS/Beidou antenna

3.7.2 Software description

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* 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 tool-chain and Compiler: The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - System Workbench for STM32 (*SW4STM32*) + ST-LINK

3.7.3 Hardware and software setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

3.7.3.1 Hardware setup

The following hardware components are needed:

1. One *STM32 Nucleo* development platform (suggested order code: *NUCLEO-F401RE*)
2. One GNSS expansion board (order code: *X-NUCLEO-GNSS1A1*)
3. One GPS/GLONASS/Beidou antenna to be connected to the X-NUCLEO-GNSS1A1 (bundled with the GNSS expansion board)
4. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

For the X-NUCLEO-GNSS1A1 proper operations, the following jumper settings must be used:

- J2 open
- J3 closed
- J4 closed
- J5 open
- J6 closed
- J7 closed
- J8 open
- J9 closed
- J10 open
- J11 closed
- J12 closed
- J13 closed
- J14 closed
- J15 closed

3.7.3.2 Software setup

This section lists the minimum requirements needed to set up the SDK, run the sample applications and customize them.

3.7.3.2.1 Development tool-chains and compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software and follow the system requirements and setup information described by the selected IDE provider.

3.7.3.2.2 PC Utility

The firmware upgrader Java utility for PC has no particular minimum requirements, you can use any Linux, Microsoft or OSX PC with:

- Java Runtime Environment (JRE)
- 1 x USB port

3.7.3.3 System setup guide

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

3.7.3.3.1 STM32 Nucleo and GNSS expansion board setup

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

You can download the appropriate ST-LINK/V2-1 USB driver (*STSW-LINK009*) from www.st.com.

The GNSS expansion board *X-NUCLEO-GNSS1A1* is easily connected to the STM32 Nucleo development through the Arduino UNO R3 extension connector. The X-NUCLEO-GNSS1A1 expansion board is capable of interfacing with the external STM32 microcontroller on the STM32 Nucleo board either via UART or Inter-Integrated Circuit (I2C) channels.

3.7.3.3.2 GNSS Firmware Upgrader utility setup

The Firmware Updater 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.

In order 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.

4 Revision history

Table 2: Document revision history

Date	Version	Changes
06-Dec-2017	1	Initial release.

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. 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.