
Getting started with X-LINUX-GNSS1 package for developing GNSS applications on Linux OS

Introduction

X-LINUX-GNSS1 is an STM32 MPU OpenSTLinux software expansion package that runs on the Arm Cortex®-A7-based core of the STM32MP1 microprocessor on the STM32MP157F-DK2 discovery kit to demonstrate GNSS-based applications.

X-LINUX-GNSS1 includes user space application, a device tree for the Teseo-LIV3F global navigation satellite system (GNSS) device, a library for NMEA protocol support and POSIX thread for task scheduling to ensure better asynchronous message parsing.

The software contains various application modules to retrieve the NMEA GNSS data and upload it to DSH-ASSETTRACKING.

The source code can be ported to any Linux platform.

1 X-LINUX-GNSS1 overview

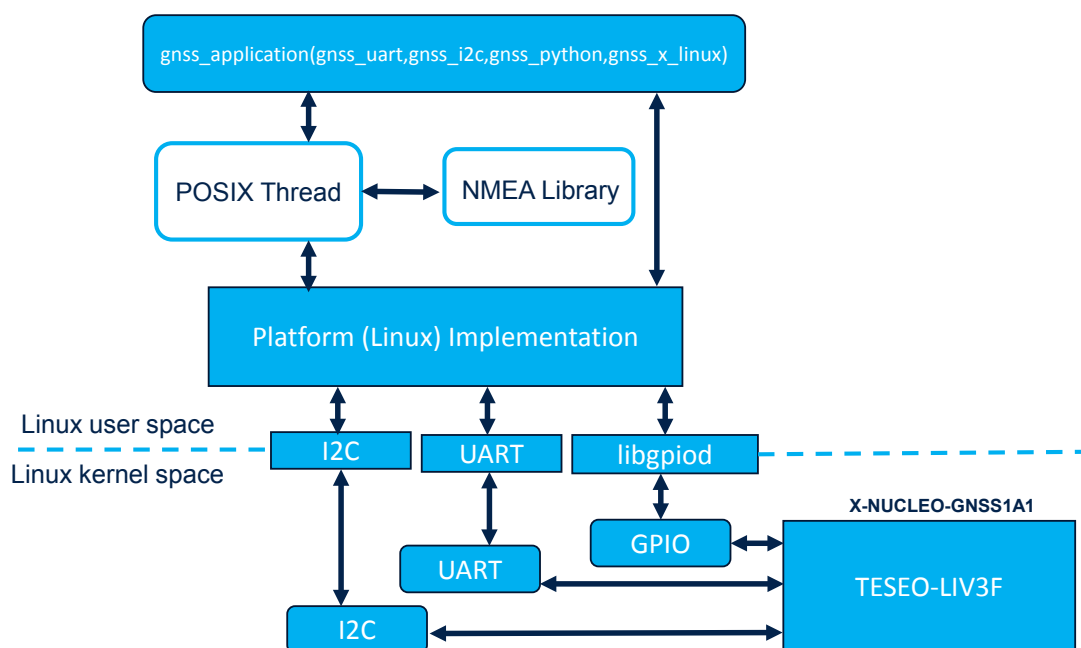
The X-LINUX-GNSS1 software provides a user space application running on STM32MP157F-DK2 for the X-NUCLEO-GNSS1A1 expansion board based on the Teseo-LIV3F tiny global navigation satellite system (GNSS) module.

The software package contains the following modules:

1. gnss_app (x-linux-gnss)
2. C utility(gnss_uart and gnss_i2c)
3. A Python utility (gnss_pynmea2.py)

Each software module can run independently to acquire the GNSS NMEA data from the X-NUCLEO-GNSS1A1 over UART and I²C.

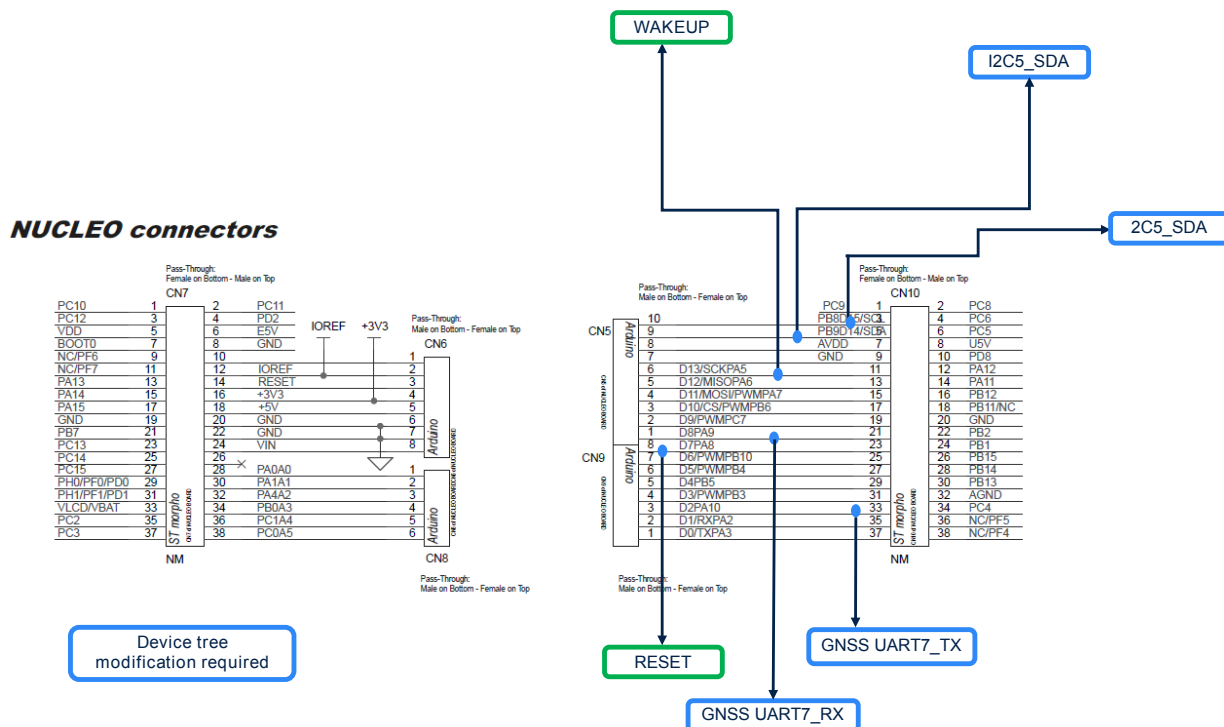
Figure 1. X-LINUX-GNSS1 user space application architecture



The X-NUCLEO-GNSS1A1 device tree has been modified to configure the UART7 and I2C5 on the Arduino connector. For UART, the underlying dev/ttySTM2 is enabled, whereas /dev/i2c-1 is enabled for I²C.

The X-LINUX-GNSS1 software interacts with the lower layer peripheral drivers (I²C and UART) through the user space application. It uses `termios` for UART and file descriptor reading for the I²C peripheral.

The software also exploits POSIX thread to run two parallel tasks (Consumer Task and Console Task). The Consumer Task acquires the NMEA data, parse them and populates the NMEA data structure. The Console Task reads the input from the user application and provides the information from the populated NMEA data structure, such as position, speed, elevation, etc., based on the provided inputs.

Figure 2. X-NUCLEO-GNSS1A1 hardware connections with required tree modification


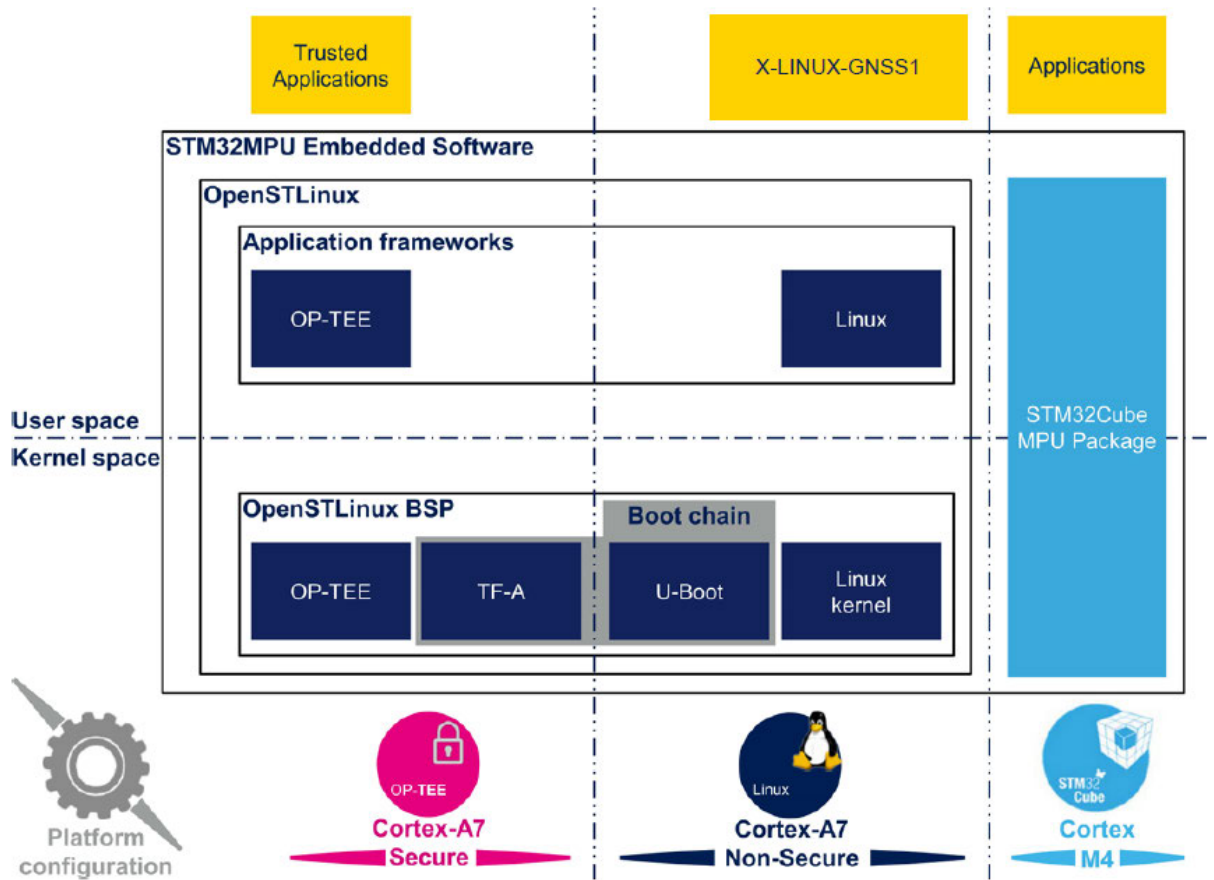
1.1 Features

- Standalone applications to read the NMEA data over UART and I²C
- Complete software to build applications on Linux using [Teseo-LIV3F](#) and [Teseo-VIC3DA](#) GNSS module
- Middleware for the NMEA protocol
- POSIX thread task scheduling to ensure better asynchronous message parsing
- Easy portability across different Linux platforms
- Application example to retrieve and parse GNSS data and send them to [DSH-ASSETTRACKING](#) for live tracking
- Python example to read the NMEA data over UART

1.2 Architecture

The software package runs on the ARM Cortex-A7 core of the [STM32MP157F-DK2](#). The [X-LINUX-GNSS1](#) interacts with the lower layers libraries and SPI lines exposed by the Linux software framework.

Figure 3. X-LINUX-GNSS1 application architecture in Linux environment



1.3 Software package structure

The X-LINUX-GNSS_V1.0.0 release package contains Linux user application C examples, a Python example, the device tree and a Yocto layer recipe.

You can run any application independently in the Application folder to retrieve the GNSS NMEA data.

Figure 4. X-LINUX-GNSS1_V1.1.0 release package structure - top level

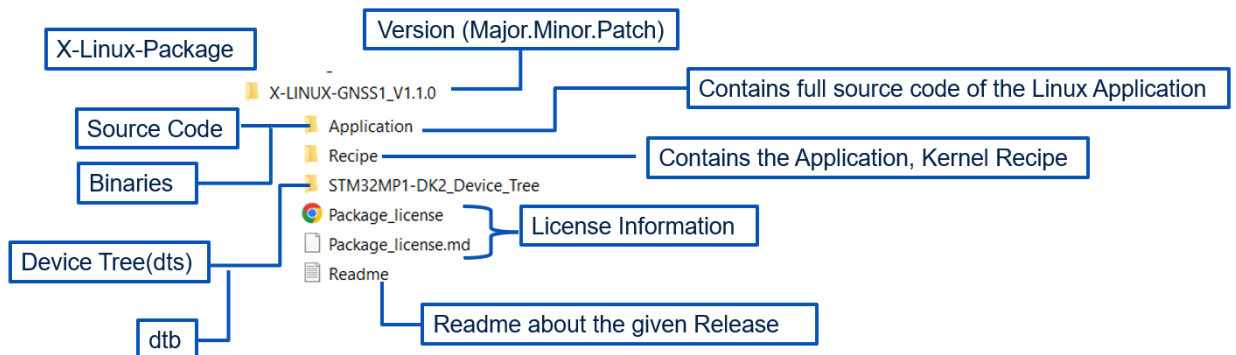
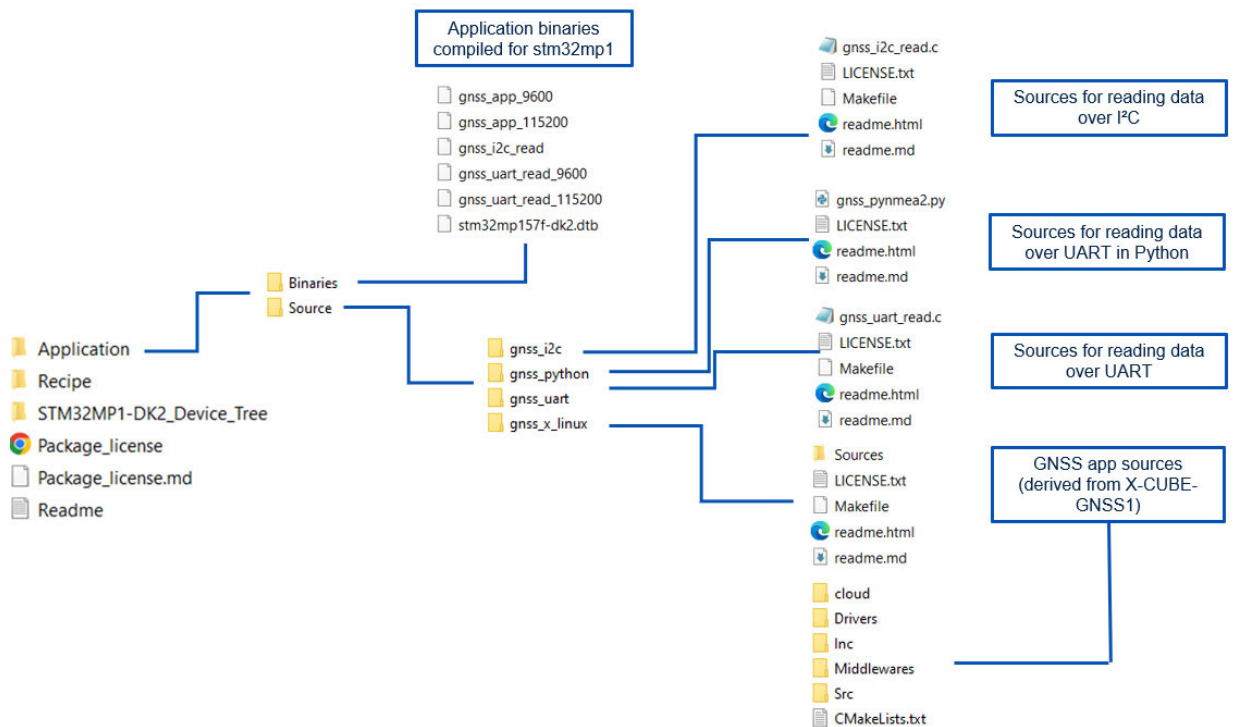
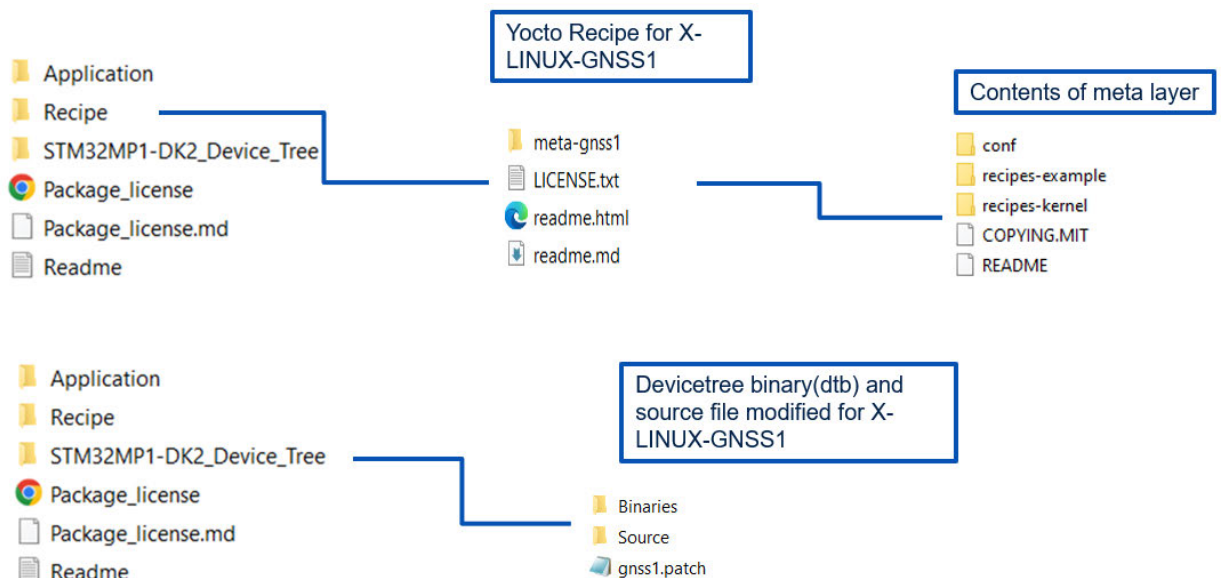


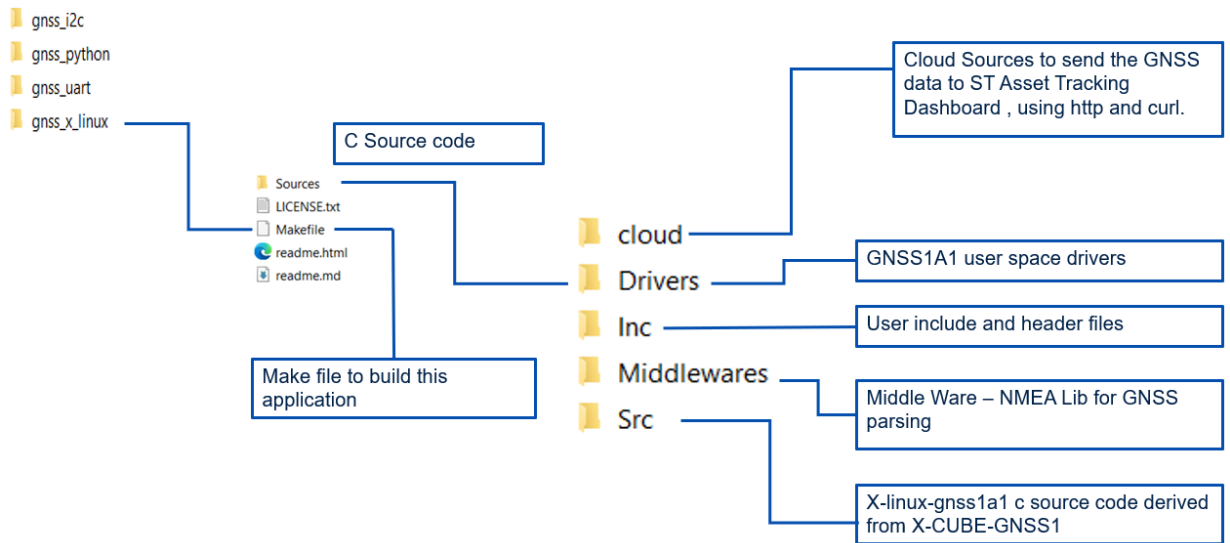
Figure 5. X-LINUX-GNSS_V1.1.0 release package structure - application folders

Figure 6. X-LINUX-GNSS_V1.1.0 release package structure - GNSS other folders


1.3.1

gnss_app

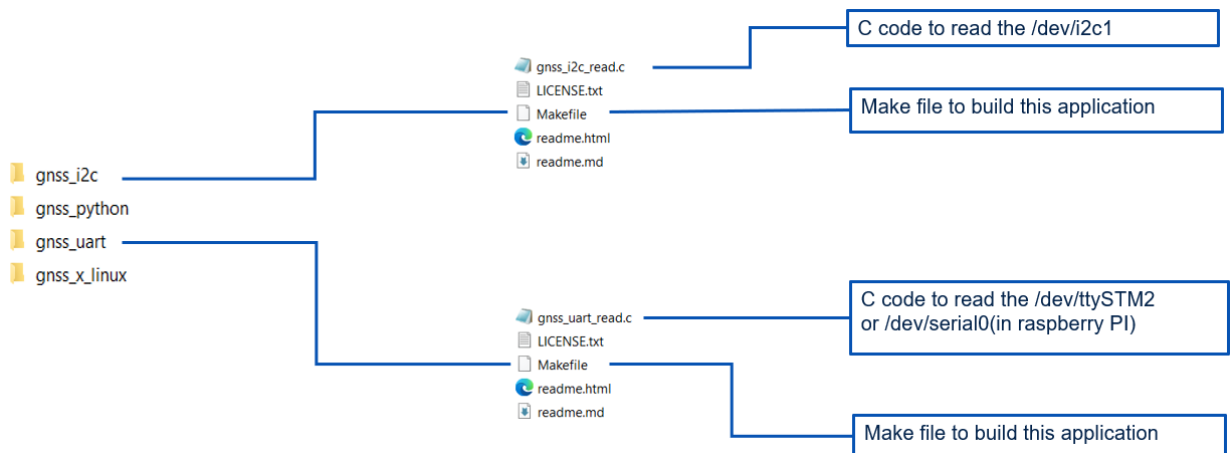
This application accesses the GPS data over UART (/dev/ttySTM2) and I²C (/dev/i2c-1) interface. The settings to enable UART and I²C are provided separately in the device tree file folder.

Through the gnss_app you can upload the data to the cloud ([DSH-ASSETTRACKING](#)).

Figure 7. gnss_app folder structure


1.3.2 C utility

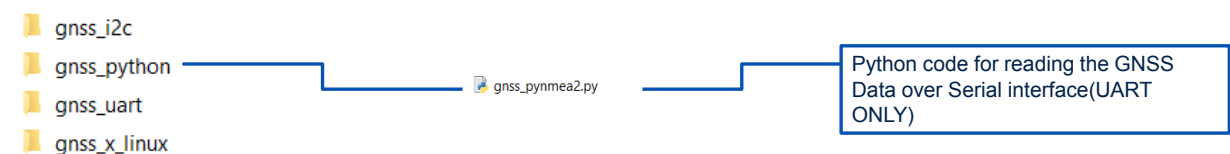
This Linux user space C application reads data from UART (/dev/ttySTM2) and I²C (/dev/i2c-1) interface.

Figure 8. C utility - gnss-uart and gnss-i2c


1.3.3 Python code application

This application is a basic Python code to read data from UART.

Important: You need to install `pyserial` and `pyneea2` library before using this application.

Figure 9. Python code application structure


2 Hardware setup

The [X-LINUX-GNSS1](#) is compatible with the [X-NUCLEO-GNSS1A1](#) expansion board which can be directly plugged on the [STM32MP157F-DK2](#) discovery kit Arduino connectors.

Step 1. Set the X-NUCLEO-GNSS1A1 jumpers as shown in the figures and the tables below.

Figure 10. X-NUCLEO-GNSS1A1 expansion board jumper settings

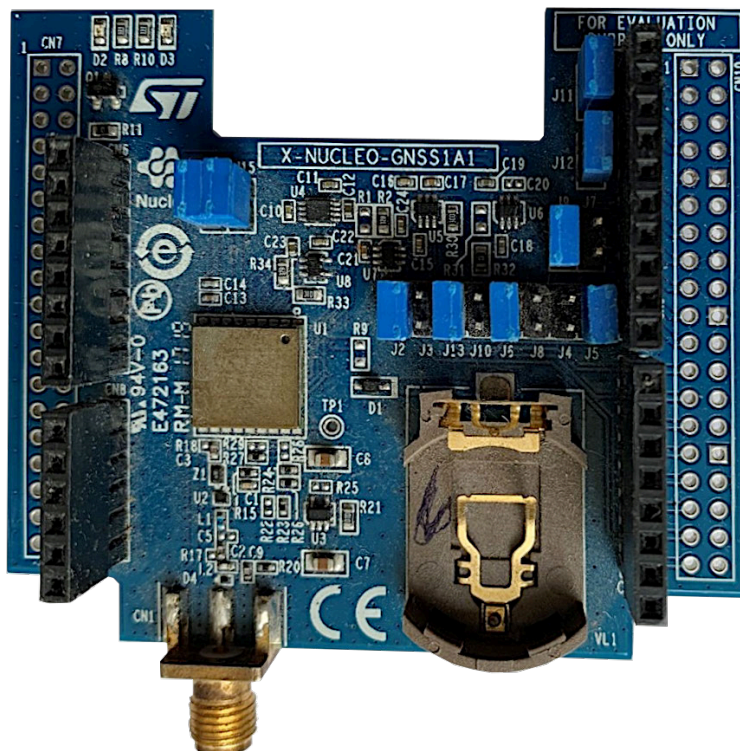


Figure 11. X-NUCLEO-GNSS2A1 expansion board jumper settings

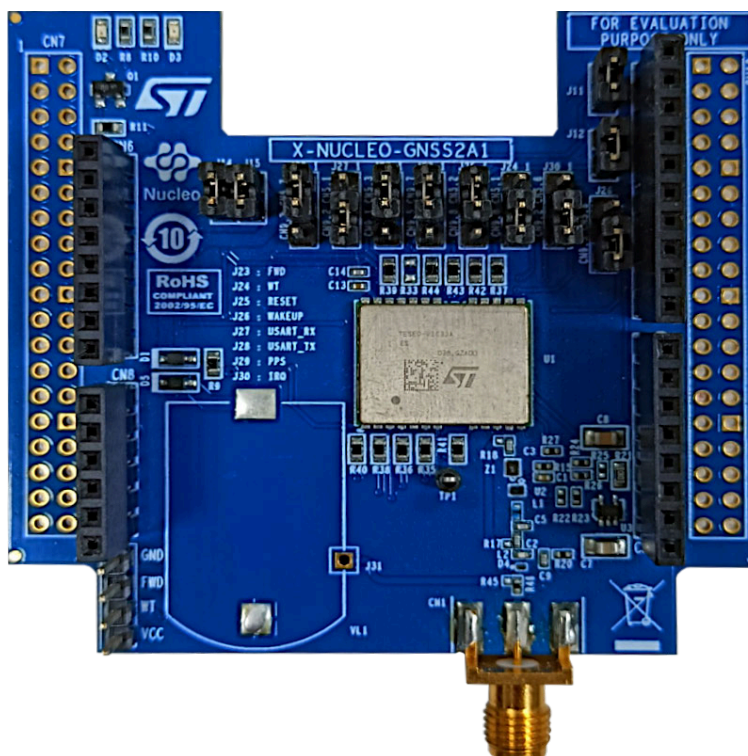


Table 1. X-NUCLEO-GNSS1A1 jumper configuration

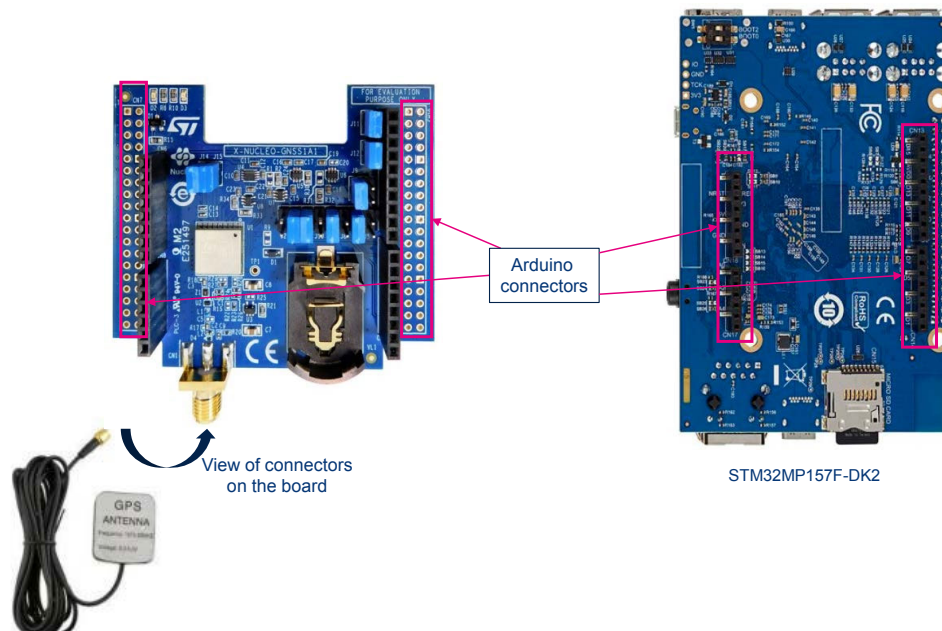
Signal	Arduino connector	Discovery kit	Jumper	Configuration
I2C-SCL	D15	PB8	J11	Closed
I2C-SDA	D14	PB9	J12	Closed
Wakeup	D13	PA5	J9	Closed
Wakeup	D4	PB5	J7	Open
Reset	D9	PC7	J10	Open
Reset	D7	PA8	J13	Closed
PPS	D6	PB10	J6	Closed
PPS	D2	PA10	J8	Open
UART-RX	D8	PA9	J3	Open
UART-TX	D2	PA10	J4	Open
UART-RX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

Table 2. X-NUCLEO-GNSS2A1 jumper configuration

Signal	Jumper	Configuration
I2C-SCL	J11	Closed
I2C-SDA	J12	Closed
VCC-VCC_IO	J14	Closed
V14Bat	J15	Closed
SYS_FWD	J23	1-2
SYS_WHEELTICK	J24	2-3
SYS_RESETn	J25	1-2
SYS_WAKEUP	J26	2-3
UART-RX	J27	2-3
UART-TX	J28	1-2
SYS_PPS	J29	1-2
SYS_IRQ	J30	2-3

Step 2. Plug the expansion board onto the discovery kit Arduino connectors.

Figure 12. X-NUCLEO-GNSS1A1 and STM32MP157F-DK2 connection



Step 3. Connect the GPS/GLONASS/Beidou antenna provided with X-NUCLEO-GNSS1A1.

Tip: Keep the antenna outdoor for better reception.

The STM32MP157F-DK2 is powered by the USB Type-C cable.

3 Software setup

This section describes the software setup required to build, flash, transfer, and run the GNSS application.

3.1 PC/laptop requirements

For the software setup, you need:

- a Linux® PC/laptop running Ubuntu® 18.04 or 20.04
- detailed instructions at https://wiki.st.com/stm32mpu/wiki/PC_prerequisites

3.2 Installing the SDK

The software package contains the binaries you can transfer using scp command.

Install the SDK to help you build a customized application, by following the instructions at https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157x-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Install_the_SDK.

3.3 Downloading the kernel sources (developer package)

Downloading the kernel sources is required to build the device tree. The software package already contains the binaries (dtb) which can be transferred using scp command. For the complete guide on how to download the kernel sources, see https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157x-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Modify,_rebuild_and_reload_the_Linux%C2%AE_kernel#

KERNEL SOURCE PATH

```
= ~/STM32MPU_workspace/STM32MP15-Ecosystem-v3.0.0/Developer-Package/stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31/sources/arm-ostl-linux-gnueabi/linux-stm32mp-5.10.10-r0/linux-5.10.10$
```

3.4 Downloading the distribution package

This is required to build the recipes and create STM32MP1 images which have GNSS application and device tree settings embedded (see https://wiki.st.com/stm32mpu/wiki/STM32MP1_Distribution_Package).

3.5 Connecting to the discovery kit

To transfer the built binaries (application, device trees) to the STM32MP157F-DK2 discovery kit from your PC/laptop, you can transfer the binaries either by hotspot (https://wiki.st.com/stm32mpu/wiki/How_to_configure_a_wlan_interface_on_hotspot_mode) or via Wi-Fi connectivity (https://wiki.st.com/stm32mpu/wiki/How_to_setup_wifi_connection).

4 Building and running the example

The code can be built using simple Makefile utility for the starter package or using bitbake for the distribution package. For Python, no building/compiling is required but it is dependent on pyserial and pynmea2 package which needs to be installed.

4.1 Using Makefile (for starter package)

Step 1. Download the X-LINUX-GNSS1 package.

Step 2. Create a directory named “gnss”.

```
$mkdir gnss
$cd gnss
```

Step 3. Download or clone the package (X-LINUX-GNSS_V1.1.0.tar.xz) from www.st.com and extract it.

```
$tar xvf X-LINUX-GNSS_V1.1.0.tar.xz
```

You will get the X-LINUX-GNSS1_V1.1.0 folder.

Figure 13. Cloning the package



4.1.1 How to build the gnss_app

Step 1. Modify the device tree or copy it from the folder provided.

Step 2. Copy the dts file in the directory: X-LINUX-GNSS1_V1.1.0/ STM32MP1-DK2_Device_Tree/Source to the kernel source directory at <KERNEL SOURCE PATH>/ arch/arm/boot/dts/.

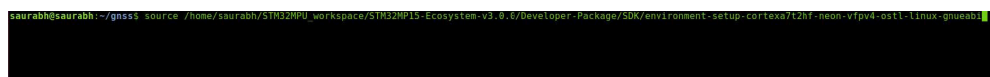
Step 3. Download the kernel sources as described in Section 3.3 .

```
$cd path-to/X-LINUX-GNSS1_V1.1.0/Application/Source/gnss_x_linux/Sources
$cp stm32mp157f-dk2.dts <KERNEL SOURCE PATH>/ arch/arm/boot/dts
```

Step 4. Source the path of the SDK (previously downloaded and installed).

```
$source <SDK PATH>/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
```

Figure 14. Sourcing the SDK path



Step 5. Build the device tree.

```
$cd <KERNEL SOURCE PATH>
$make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- menuconfig
$make arch=ARM menuconfig
$make ARCH=arm uImage vmlinux dtbs LOADADDR=0xC2000040 (Optional)
$make ARCH=arm modules (Optional)
```

Step 6. Once the dtbs are built, copy them to the STM32MP157F-DK2 via hotspot or Wi-Fi.

```
$scp <KERNEL SOURCE PATH>/arch/arm/boot/dts/stm32mp157f_dk2.dtb root@192.168.72.1:/boot
```

Step 7. Build the gnss_app.

```
$cd gnss
```

Step 8. Build the gnss_app. cd to the gnss_x_linux location.

```
$cd path to/X-LINUX-GNSS1_V1.1.0/Application/Source/gnss_x_linux
```

Figure 15. STM32MP157F-DK2 - building the application

```

saurabh@saurabh:~/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux$ make
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/gnss_lib_config.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/gnss_utils.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/main.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/app_gnss.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/cloud/cloud_comm_config.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/cloud/cloud_comm_https.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/cloud/cloud_helper.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/GNSS1A1/gnss1a1_gnss.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f_uart.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f_queue.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f_i2c.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Middlewares/ST/lib_gnss/LibGNSS5/Src/gnss_parser.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Middlewares/ST/lib_gnss/LibGNSS5/Src/gnss_data.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Middlewares/ST/lib_gnss/LibNMEA/Src/NMEA_parser.o
Linking gnss_app
saurabh@saurabh:~/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux$

```

The executable (gnss_app) will be built in the same location of Makefile.

Figure 16. gnss_app created

```

saurabh@saurabh:~/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux$ make
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/gnss_lib_config.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/gnss_utils.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/main.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Src/app_gnss.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/cloud/cloud_comm_config.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/cloud/cloud_comm_https.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/cloud/cloud_helper.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/GNSS1A1/gnss1a1_gnss.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f_uart.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f_queue.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f_i2c.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Drivers/BSP/Components/teseo_liv3f/teseo_liv3f.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Middlewares/ST/lib_gnss/LibGNSS5/Src/gnss_parser.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Middlewares/ST/lib_gnss/LibGNSS5/Src/gnss_data.o
Building /home/saurabh/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux/Build/Middlewares/ST/lib_gnss/LibNMEA/Src/NMEA_parser.o
Linking gnss_app
saurabh@saurabh:~/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux$ ls -l
total 80
-rwxrwxr-x 7 saurabh saurabh 4096 Jul 22 16:03 Build
-rwxrwxr-x 1 saurabh saurabh 67984 Jul 22 16:03 gnss_app
-rwxrwxr-x 1 saurabh saurabh 2744 Jul 22 16:13 Makefile
-rwxrwxr-x 7 saurabh saurabh 4096 Jul 15 22:31 Sources
saurabh@saurabh:~/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux$ ./gnss_app
bash: ./gnss_app: cannot execute binary file: Exec format error
saurabh@saurabh:~/release-package/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_linux$

```

Step 9. Transfer the application to STM32MP157F-DK2.

```
$scp gnss_app root@192.168.72.1:/
```

Step 10. On the STM32MP157F-DK2, sync and reboot.

```
$sync
$reboot
```

Step 11. Run the application.

```
$cd /
$./gnss_app
```

Figure 17. STM32MP157F-DK2 - running the application

```
root@stm32mp1:/#
root@stm32mp1:/# ./gnss_app
Teseo_Consumer_Task_Init...
Console_Parse_Task_Init.....
Select a command:
 1 - getpos
 2 - lastpos
 3 - wakestatus
 4 - help
 5 - debug
 6 - track
 7 - lasttrack
 9 - getgnsmg
10 - getgpgst
11 - getgprmc
12 - getgsamg
13 - getgsvmg
19 - ext-help
20 - Upload to Cloud
21 - Stop Upload to Cloud

Save configuration (y/n)?
```

Step 12. From the menu above, select 11 to get the GPS coordinates.

Figure 18. Selecting option 11 from the gnss_app

```

3 - wakestatus
4 - help
5 - debug
6 - track
7 - lasttrack
9 - getgnsmmsg
10 - getpgst
11 - getprmc
12 - getsamsg
13 - getsvmsg
19 - ext-help
20 - Upload to Cloud
21 - Stop Upload to Cloud

Save configuration (y/n)?
> 11
getprmc=11

UTC: [ 11:56:50 ]
Status: [ V ] -- Warning (reported in NO FIX conditions)
Latitude: [ 28' 32'' N ]
Longitude: [ 77' 21'' E ]
Speed over ground (knots): [ 0.0 ]
Trackgood: [ 0.0 ]
Date (ddmmyy): [ 290721 ]
Magnetic Variation: [ 0.0 ]
Magnetic Var. Direction: [ - ]

Select a command:
1 - getpos
2 - lastpos
3 - wakestatus
4 - help
5 - debug
6 - track
7 - lasttrack
9 - getgnsmmsg
10 - getpgst
11 - getprmc
12 - getsamsg
13 - getsvmsg
19 - ext-help
20 - Upload to Cloud
21 - Stop Upload to Cloud

Save configuration (y/n)?
>
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyACM0

```

4.1.2

How to build C utility gnss_uart and gnss_i2c

Step 1. Enter the <Path to > /X-LINUX-GNSS1_V1.1.0\Application\Source\gnss_uart.
gnss_uart will be created in the same directory of Makefile.

Step 2. Transfer it to STM32MP157F-DK2 using scp.

```
$cd /
$ ./gnss_uart_read
```

Figure 19. Running gnss_uart

```
linux@linux-sr: ~
SPSTMSWCONFIG,1,9,17,00000000000024401d8f19a88c8f4440598b4f0130be2b400000000000002
4407ac2120f289f44402315c616828c2b400000000000002440ace28d8c239f4440*14
SPSTMSWCONFIG,1,10,17,95826e2f698c2b4000000000000024400f0c0c120f0c0c120f0c0c120f0c
0c1244454641554c542056494320434f4e46494755524154494f4e00000000000000*70
SPSTMSWCONFIG,1,11,17,0000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000*78
SPSTMSWCONFIG,1,12,17,0000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000*20
SPSTMSWCONFIG,1,13,17,0000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000*21
SPSTMSWCONFIG,1,14,17,0000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000*20
SPSTMSWCONFIG,1,15,17,0000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000*78
SPSTMSWCONFIG,1,16,17,00000000b5858000d40300000a6a00000049300000000c000000006400
000064000000640000000000000000000000000000000000000000000000*78
SPSTMSWCONFIG,1,17,17,*25
SPSTMSRSENCONFIG,IMU_GYRO,LSM6DSR,125,4375,104,34,0,1*0F
SPSTMSRSENCONFIG,IMU_ACC,LSM6DSR,2,61,104,52,0*56
$GPRMC,180139.200,V,0.000,0.0000,N,0.0000,0.0000,E,1.270422,1.1,N,7.7
$GPGGA,180139.200,0.000,N,0.000,0.000,E,0.00,99.0,0.00,0.0,M,0.0,M,0.0,0.0,0.00
$GPVTG,1.1,M,1.1,N,1.1,K,N*2C
$SPSTMCFG,1,0,50,0,50,0,05,03,60,0,0.00000,0.00000,-2.671600*12
$GPRMC,180139.450,V,0.000,0.0000,N,0.0000,0.0000,E,1.270422,1.1,N,7.4
$GPGGA,180139.450,0.000,N,0.000,0.000,E,0.00,99.0,0.00,0.0,M,0.0,M,0.0,0.0,0.00
$GPVTG,1.1,M,1.1,N,1.1,K,N*2C
$GPRMC,180139.550,V,0.000,0.0000,N,0.0000,0.0000,E,1.270422,1.1,N,7.5
$GPGGA,180139.550,0.000,N,0.000,0.000,E,0.00,99.0,0.00,0.0,M,0.0,M,0.0,0.0,0.00
$GPVTG,1.1,M,1.1,N,1.1,K,N*2C
```

```
$ ./gnss_i2c_read
```

Figure 20. Running gnss_i2c

```
$GPRMC,074246.000,V,2832.48525,N,07720.68458,E,0.0,0.0,250521,1.1,N*74
$GPGGA,074246.000,2832.48525,N,07720.68458,E,0.01,99.0,260.83,M,0.0,M,0.0,0.0,0.00
$GPVTG,0.0,T,1.1,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,76,,,,,,,,,99.0,99.0,99.0*1F
$GPGSV,3,1,09,02,74,280,,06,60,026,,12,39,323,,19,36,051,*7F
$GPGSV,3,2,09,24,33,252,,28,32,127,,17,20,069,,14,13,133,*70
$GPGSV,3,3,09,05,10,187,,,,,,,,,*4A
$GLGSV,2,1,08,76,73,264,31,86,46,000,,77,28,327,,71,12,107,*6A
$GLGSV,2,2,08,85,12,041,,70,10,059,,75,00,000,44,87,00,000,33*61
$GPGLL,2832.48525,N,07720.68458,E,074246.000,V,N*42
$PSTMCPU,30.80,-1,49*44
$GPRMC,074247.000,V,2832.48525,N,07720.68458,E,0.0,0.0,250521,1.1,N*75
$GPGGA,074247.000,2832.48525,N,07720.68458,E,0.01,99.0,260.83,M,0.0,M,0.0,0.0,0.00
$GPVTG,0.0,T,1.1,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,76,,,,,,,,,99.0,99.0,99.0*1F
$GPGSV,3,1,09,02,74,280,,06,60,026,,12,39,323,,19,36,051,*7F
$GPGSV,3,2,09,24,33,252,,28,32,127,,17,20,069,,14,13,133,*70
$GPGSV,3,3,09,05,10,187,,,,,,,,,*4A
$GLGSV,2,1,08,76,73,264,31,86,46,000,,77,28,327,,71,12,107,*6A
$GLGSV,2,2,08,85,12,041,,70,10,059,,75,00,000,44,87,00,000,32*60
$GPGLL,2832.48525,N,07720.68458,E,074247.000,V,N*43
$PSTMCPU,31.81,-1,49*44
$GPRMC,074248.000,V,2832.48525,N,07720.68458,E,0.0,0.0,250521,1.1,N*7A
$GPGGA,074248.000,2832.48525,N,07720.68458,E,0.01,99.0,260.83,M,0.0,M,0.0,0.0,0.00
$GPVTG,0.0,T,1.1,M,0.0,N,0.0,K,N*02
$GNGSA,A,1,,,,,,,,,99.0,99.0,99.0*1E
$GNGSA,A,1,76,,,,,,,,,99.0,99.0,99.0*1F
$GPGSV,3,1,09,02,74,280,,06,60,026,,12,39,323,,19,36,051,*7F
$GPGSV,3,2,09,24,33,252,,28,32,127,,17,20,069,,14,13,133,*70
$GPGSV,3,3,09,05,10,187,,,,,,,,,*4A
$GLGSV,2,1,08,76,73,264,31,86,46,000,,77,28,327,,71,12,107,*6A
```

4.1.3

How to run the Python code

Step 1. Install the pyserial.

Figure 21. Installing Python dependencies - pyserial

```
$wget https://files.pythonhosted.org/packages/1e/7d/
ae3f0a63f41e4d2f6cb66a5b57197850f919f59e558159a4dd3a818f5082/pyserial-3.5.tar.gz
$tar xvf pyserial-3.5.tar.gz
$cd pyserial-3.5
$python setup.py install or python3 setup.py install
```

Step 2. Install pynmea2.

Figure 22. Installing Python dependencies - pynmea2

```
$wget https://files.pythonhosted.org/packages/88/b9/
a0fed4563f5c73eb8f4d7bb115a455863c5327ae824ac1772e2a4b1b95ee/pynmea2-1.18.0.tar.gz
$tar xvf pynmea2-1.18.0.tar.gz
$cd pynmea2-1.18.0
$python setup.py install or python3 setup.py install
```

Step 3. Enter the Python folder and copy the gnss_pynmea2.py file to STM32MP157F-DK2.

```
$scp gnss_pynmea2.py root@192.168.72.1:/
```

Step 4. Run the Python example.

```
$cd /
$python3 gnss_pynmea2.py
```

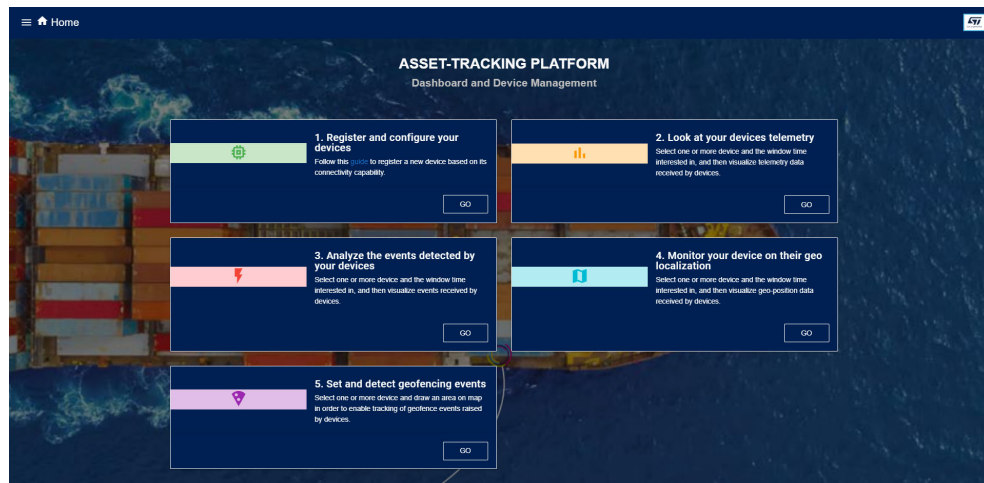
Figure 23. Running the Python example

```
Latitude=28.546478833333333and Longitude=77.353228333333334
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
Latitude=28.546478833333333and Longitude=77.35322833333333
```

4.1.4 Maps and asset tracking

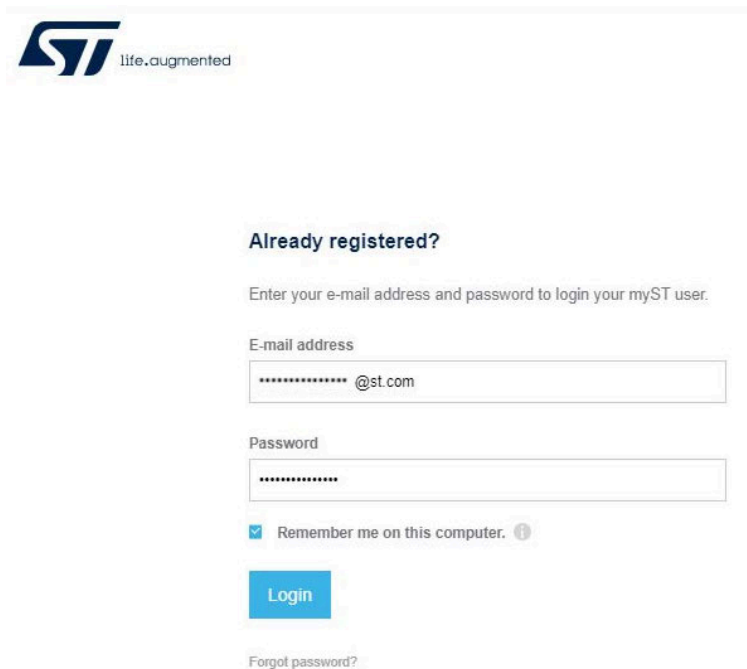
X-LINUX-GNSS1 provides an example to send GNSS data to the cloud over http. DSH-ASSETTRACKING displays GNSS data in real-time.

Figure 24. DSH-ASSETTRACKING asset tracking dashboard homepage



Step 1. Login to or create an account at <https://dsh-assettracking.st.com/#/login>.

Figure 25. DSH-ASSETTRACKING login page



The login page features the ST logo with the tagline "life.augmented" at the top left. Below the logo, the heading "Already registered?" is displayed. A sub-header reads "Enter your e-mail address and password to login your myST user." The form includes two input fields: "E-mail address" with a placeholder "*****@st.com" and "Password" with a placeholder "*****". A checkbox labeled "Remember me on this computer." with an information icon is positioned below the password field. A blue "Login" button is located at the bottom of the form. A link for "Forgot password?" is placed below the login button.

Step 2. Create a device (device name and device ID) from the [Devices] tab.

Figure 26. DSH-ASSETTRACKING - Devices tab

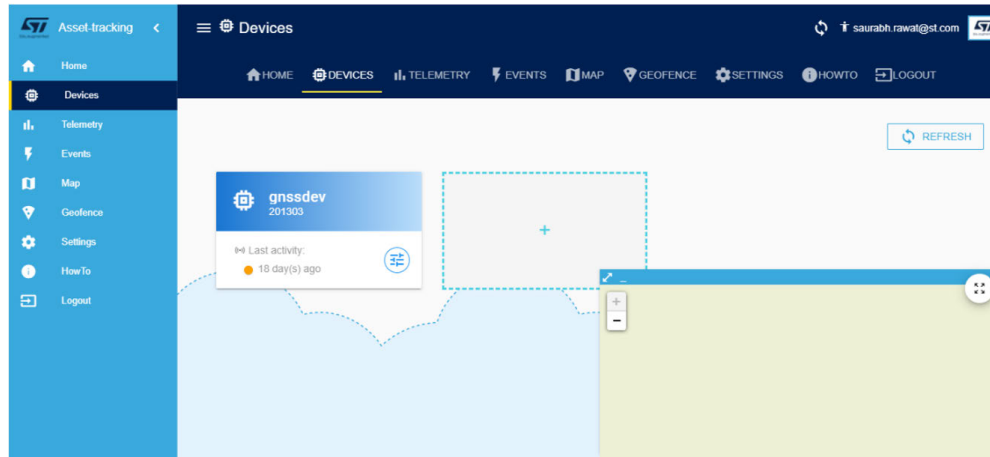
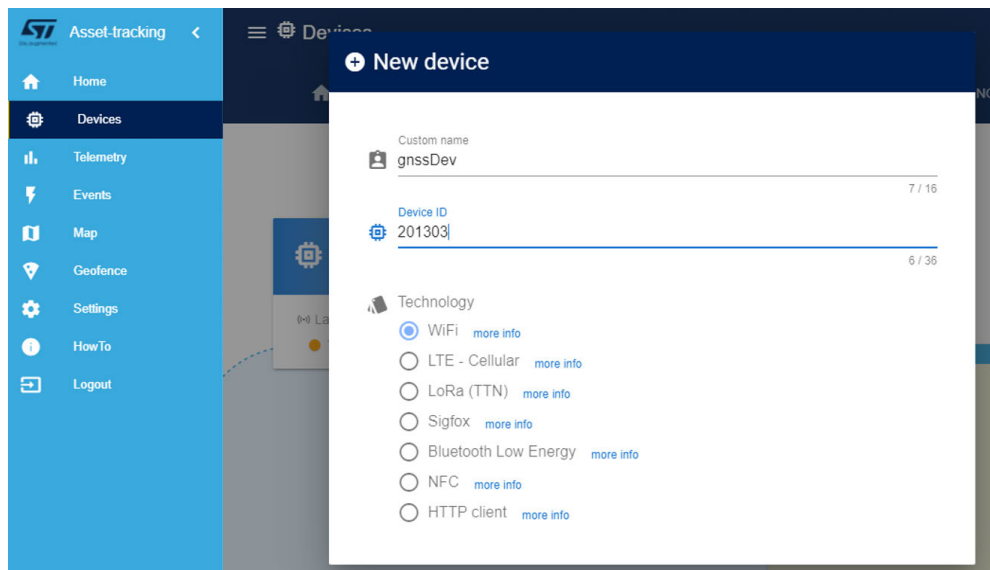
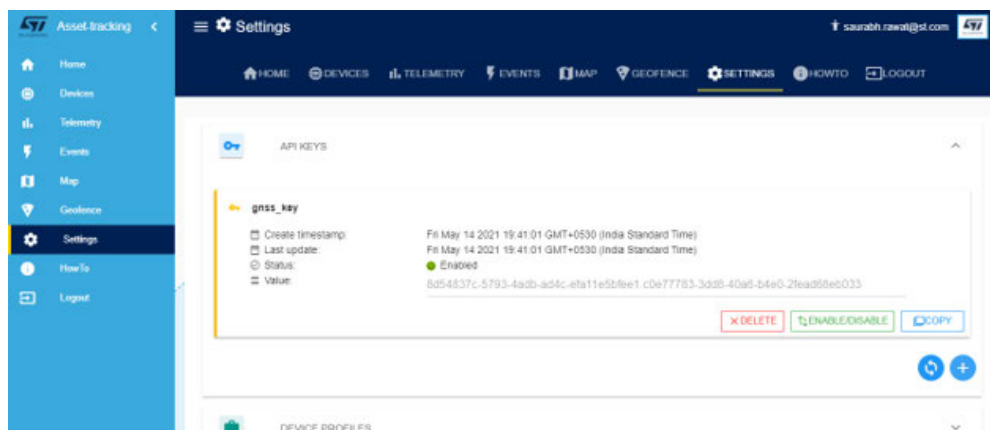


Figure 27. DSH-ASSETTRACKING - device name and ID creation



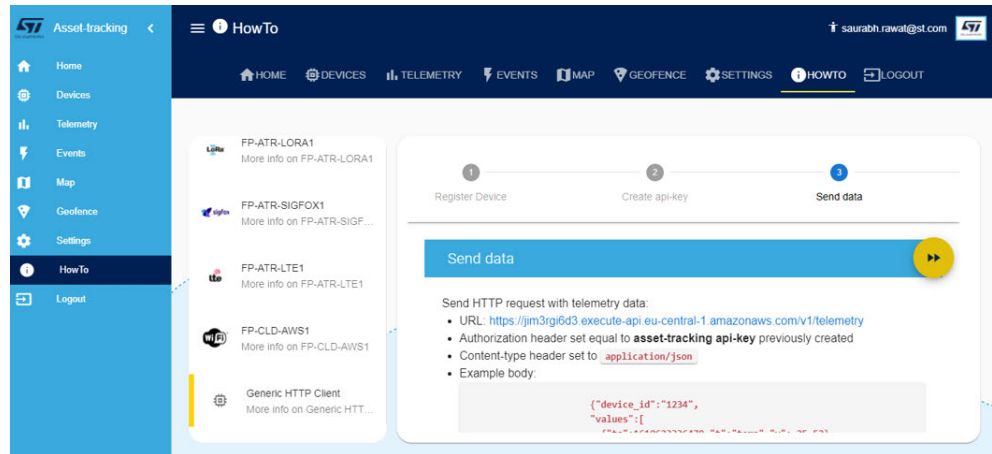
Step 3. Create the API key which will be used to send data to DSH-ASSETTRACKING.

Figure 28. DSH-ASSETTRACKING - new API key generation



Step 4. Note down the end-point.

Figure 29. DSH-ASSETTRACKING - location of the end-point



Step 5. Modify the data in the creds.conf file as per the end-point, device ID and the API key.

```
$cd /path-to/X-LINUX-GNSS1_V1.1.0/Application/Source/gnss_x_linux/Sources/cloud
$vi creds.conf
```

Figure 30. Modifying creds.conf

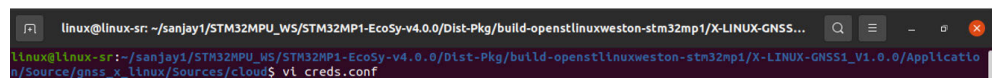
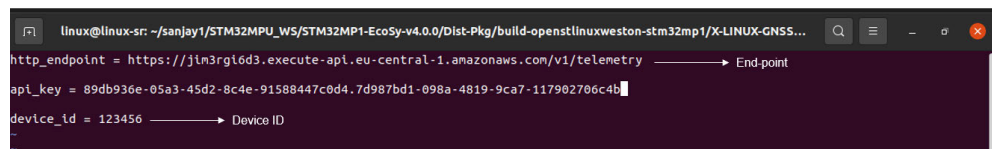


Figure 31. Add credentials in the creds.conf file



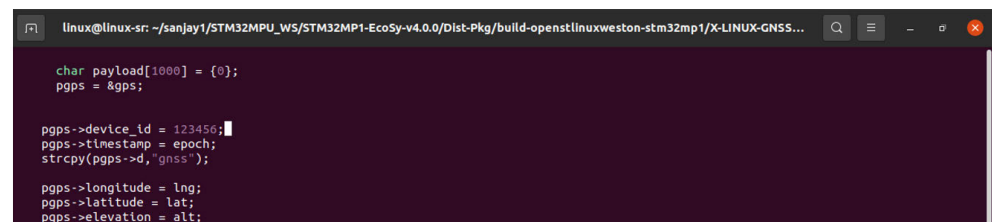
Step 6. Modify the data in the 'cloud_comm_https.c' file as per the device ID created for the device as:

```
pgps->device_id = 123456
$cd /path-to/X-LINUX-GNSS1_V1.1.0/Application/Source/gnss_x_linux/Sources/cloud
$vi cloud_comm_https.c
```

Figure 32. Modifying cloud_comm_https.c



Figure 33. cloud_comm_https.c file modified



Step 7. Build the application (if the creds file only has been modified, this step is not required).

Step 8. Repeat the steps done to build and deploy the gnss_app.

- Step 9.** Enter option 20 to upload the data to the cloud and make sure the [STM32MP157F-DK2](#) is connected to the Internet.

Figure 34. Running gnss_app and enabling cloud upload (option 20)

```
Save configuration (y/n)?
> 11
getgprmc =11

UTC:                [ 08:19:17 ]
Status:             [ V ]          -- Warning (reported in NO FIX conditions)
Latitude:           [ 28' 32'' N ]
Longitude:          [ 77' 21'' E ]
Speed over ground (knots): [ 0.0 ]
Trackgood:          [ 0.0 ]
Date (ddmmyy):      [ 110621 ]
Magnetic Variation: [ 0.0 ]
Magnetic Var. Direction: [ - ]

>Select a command:
1 - getpos
2 - lastpos
3 - wakestatus
4 - help
5 - debug
6 - track
7 - lasttrack
8 - getfwver
9 - getgnsmsg
10 - getpgst
11 - getgprmc
12 - getgsamsg
13 - getgsvmmsg
19 - ext-help
20 - Upload to Cloud
21 - Stop Upload to Cloud
```

You will get the below logs and the live tracking on the [DSH-ASSETTRACKING](#).

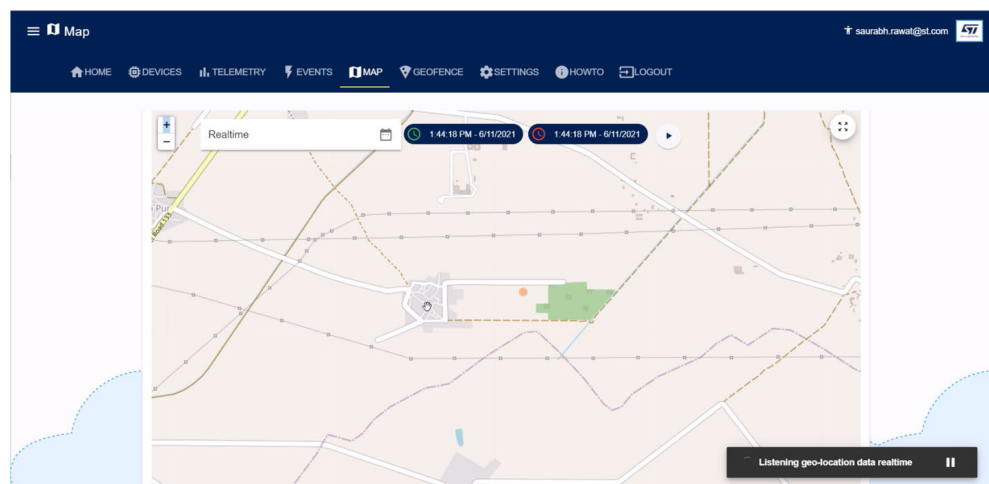
Figure 35. Gnss_app sending data to DSH-ASSETTRACKING over http

```

COM31 - lera term VI
File Edit Setup Control Window Help
* Issuer: C=US,O=Amazon,OU=Server CA 1B,CN=Amazon
* ALPN, server accepted to use http/1.1
* POST /v1/telemetry HTTP/1.1
Host: jin3rgi6d3.execute-api.eu-central-1.amazonaws.com
Accept: application/json
User-Agent: X-LINUX-GNSS101
Authorization: 8d54837c-5793-4adb-ad4c-efalle5bfee1.c8e77783-3dd8-40a6-b4e0-2fead68eb033
Content-Type: application/json
Content-Length: 121
* upload completely sent off: 121 out of 121 bytes
* Mark bundle as not supporting multiuse
* HTTP/1.1 200 OK
* Date: Fri, 11 Jun 2021 08:15:08 GMT
* Content-Type: application/json
* Content-Length: 65
* Connection: keep-alive
* x-amzn-RequestId: 504b591b-0584-4aa8-9ecb-67a6c5249de2
* x-amz-apigw-id: AwE9SPPEF1AFIgg=
* X-Amzn-Trace-Id: Root=1-60c31b3b-5e89732e06f0275097899ad;Sampled=0
* Connection #0 to host jin3rgi6d3.execute-api.eu-central-1.amazonaws.com left intact
("message": "OK", "traceId": "8178c1a2-7bb2-044f-bfdb-b654c119efd9")Uploading to Cloud Every 9800 milliseconds...
Latitude -> 28.327654
Longitude -> 77.212088
Altitude -> 2.562280
* Trying 52.28.83.33:443...
* Connected to jin3rgi6d3.execute-api.eu-central-1.amazonaws.com (52.28.83.33) port 443 (#0)
* found 128 certificates in /etc/ssl/certs/ca-certificates.crt
* ALPN, offering http/1.1
* SSL connection using TLS1.2 / ECDHE_RSA_AES_128_GCM_SHA256
* server certificate verification OK
* server certificate status verification SKIPPED
* common name: *.execute-api.eu-central-1.amazonaws.com (matched)
* server certificate expiration date OK
* server certificate activation date OK
* certificate public key: RSA
* certificate version: #3
* subject: CN=*.execute-api.eu-central-1.amazonaws.com
* start date: Sat, 29 Aug 2020 00:00:00 GMT
* expire date: Wed, 29 Sep 2021 12:00:00 GMT
* issuer: C=US,O=Amazon,OU=Server CA 1B,CN=Amazon
* ALPN, server accepted to use http/1.1
* POST /v1/telemetry HTTP/1.1
Host: jin3rgi6d3.execute-api.eu-central-1.amazonaws.com
Accept: application/json
User-Agent: X-LINUX-GNSS101
Authorization: 8d54837c-5793-4adb-ad4c-efalle5bfee1.c8e77783-3dd8-40a6-b4e0-2fead68eb033
Content-Type: application/json
Content-Length: 121
* upload completely sent off: 121 out of 121 bytes
* Mark bundle as not supporting multiuse
* HTTP/1.1 200 OK
* Date: Fri, 11 Jun 2021 08:15:10 GMT
* Content-Type: application/json
* Content-Length: 65
* Connection: keep-alive
* x-amzn-RequestId: 3df8df6e-1cf5-4f37-af7b-94e5351d472e
* x-amz-apigw-id: AwE_eHqql1AFenA=
* X-Amzn-Trace-Id: Root=1-60c31b3b-399ed67f05c71f3c5dc29896;Sampled=0
* Connection #0 to host jin3rgi6d3.execute-api.eu-central-1.amazonaws.com left intact

```

Figure 36. Gnss_app data in real-time



4.2 Using BitBake (for distribution package)

Step 1. Download the distribution package and do a bitbake.

```

$DISTRO=openstlinux-weston MACHINE=stm32mp1 source layers/meta-st/scripts/
envsetup.sh
$ bitbake st-image-weston

```

Step 2. Accept the EULA by typing "y" and pressing enter.

Figure 37. End user license agreement

```

saurabh7@vm613801: ~/dp/openstlinux-5.10-dunfell-mp1-21-03-31
#####
BROADCOM BCM33XX
SOFTWARE LICENSE AGREEMENT
The accompanying software in binary code form ("Software"), is licensed to you, or, if you are accepting on behalf of an entity, the entity and its affiliates exercising rights hereunder ("Licensee") subject to the terms of this software license agreement ("Agreement"), unless Licensee and Broadcom Corporation ("Broadcom") execute a separate written software license agreement governing use of the Software. ANY USE, REPRODUCTION, OR DISTRIBUTION OF THE SOFTWARE CONSTITUTES LICENSEE'S ACCEPTANCE OF THIS AGREEMENT.
1. License. Subject to the terms and conditions of this Agreement, Broadcom hereby grants to Licensee a limited, non-exclusive, non-transferable, royalty-free license: (i) to use and integrate the Software with any other software; and (ii) to reproduce and distribute the Software complete, unmodified, and as provided by Broadcom, solely for use with Broadcom proprietary integrated circuit product(s) sold by Broadcom with which the Software was designed to be used, or their successors.
2. Restrictions. Licensee shall distribute Software with a copy of this Agreement. Licensee shall not remove, efface or obscure any copyright or trademark notices from the Software. Reproductions of the Broadcom copyright notice shall be included with each copy of the Software, except where such Software is embedded in a manner not readily accessible to the end user. Licensee shall not: (i) use, license, sell or otherwise distribute the Software except as provided in this Agreement; (ii) attempt to modify in any way, reverse engineer, decompile or disassemble any portion of the Software; or (iii) use the Software or other material in violation of any applicable law or regulation, including but not limited to any regulatory agency. This Agreement shall automatically terminate upon Licensee's failure to comply with any of the terms of this Agreement. In such event, Licensee will destroy all copies of the Software and its component parts.
3. Ownership. The Software is licensed and not sold. Title to and ownership of the Software, including all intellectual property rights thereto, and any portion thereof remain with Broadcom or its licensors. Licensee hereby covenants that it will not assert any claim that the Software created by or for Broadcom infringe any intellectual property right owned or controlled by Licensee.
4. Disclaimer. THE SOFTWARE IS OFFERED "AS IS," AND BROADCOM PROVIDES AND GRANTS AND LICENSEE RECEIVES NO SUPPORT AND NO WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, BY STATUTE, COMMUNICATION OR CONDUCT WITH LICENSEE, OR OTHERWISE. BROADCOM SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A SPECIFIC PURPOSE, OR NONINFRINGEMENT CONCERNING THE SOFTWARE OR ANY UPGRADES TO OR DOCUMENTATION FOR THE SOFTWARE. WITHOUT LIMITATION OF THE ABOVE, BROADCOM GRANTS NO WARRANTY THAT THE SOFTWARE IS ERROR-FREE OR WILL OPERATE WITHOUT INTERRUPTION, AND NO GRANTS NO WARRANTY REGARDING ITS USE OR THE RESULTS THEREFROM INCLUDING, WITHOUT LIMITATION, ITS CORRECTNESS, ACCURACY, OR RELIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY LAW, IN NO EVENT SHALL BROADCOM OR ANY OF ITS LICENSORS HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER FOR BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE) OR OTHERWISE, ARISING OUT OF THIS AGREEMENT OR USE, REPRODUCTION, OR DISTRIBUTION OF THE SOFTWARE, INCLUDING BUT NOT LIMITED TO LOSS OF DATA AND LOSS OF PROFITS, EVEN IF SUCH PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.
5. Export Laws. LICENSEE UNDERSTANDS AND AGREES THAT THE SOFTWARE IS SUBJECT TO UNITED STATES AND OTHER APPLICABLE EXPORT-RELATED LAWS AND REGULATIONS AND THAT LICENSEE MAY NOT EXPORT, RE-EXPORT OR TRANSFER THE SOFTWARE OR ANY DIRECT PRODUCT OF THE SOFTWARE EXCEPT AS PERMITTED UNDER THOSE LAWS. WITHOUT LIMITING THE FOREGOING, EXPORT, RE-EXPORT, OR TRANSFER OF THE SOFTWARE TO CUBA, IRAN, NORTH KOREA, SUDAN, AND SYRIA IS PROHIBITED.
Do you accept the EULA you just read? (y/n)

```

After accepting, you will access the build directory.

Step 3. Download the gnss package from www.st.com in the build directory as shown below.

Figure 38. gnss package download

```

saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$ ls
conf X-LINUX-GNSS1_V1.0.0 X-LINUX-GNSS1_V1.0.0.tar.xz
saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$

```

Step 4. Create a layer (meta-gnss1).

```
$bitbake-layers create-layer --priority 7 ../layers/meta-st/meta-gnss1
```

Figure 39. meta-gnss1 layer creation

```

saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$ bitbake-layers create-layer --priority 7 ../layers/meta-st/meta-gnss1
NOTE: Starting bitbake server...
Add your new layer with 'bitbake-layers add-layer ../layers/meta-st/meta-gnss1'
saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$

```

Step 5. Add a layer (recipes-gnss1).

```
$ bitbake-layers add-layer ../layers/meta-st/meta-gnss1
```

Figure 40. Adding meta-gnss1 layer

```
saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$ bitbake-layers add-layer ../layers/meta-st/meta-gnss1
NOTE: Starting bitbake server...
```

You can see the added layer by typing:

```
$bitbake-layers show-layers
```

Figure 41. Showing all meta layers

```
saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer path priority
meta-python /home/saurabh/gnss/layers/meta-openembedded/meta-python 7
meta-oe /home/saurabh/gnss/layers/meta-openembedded/meta-oe 6
meta-gnome /home/saurabh/gnss/layers/meta-openembedded/meta-gnome 7
meta-initramfs /home/saurabh/gnss/layers/meta-openembedded/meta-initramfs 8
meta-multimedia /home/saurabh/gnss/layers/meta-openembedded/meta-multimedia 6
meta-networking /home/saurabh/gnss/layers/meta-openembedded/meta-networking 5
meta-webserver /home/saurabh/gnss/layers/meta-openembedded/meta-webserver 6
meta-filessystems /home/saurabh/gnss/layers/meta-openembedded/meta-filessystems 6
meta-perl /home/saurabh/gnss/layers/meta-openembedded/meta-perl 6
meta-st-stm32mp /home/saurabh/gnss/layers/meta-st-stm32mp 6
meta-qt5 /home/saurabh/gnss/layers/meta-qt5 7
meta-st-openstlinux /home/saurabh/gnss/layers/meta-st-openstlinux 5
meta /home/saurabh/gnss/layers/openembedded-core/meta 5
meta-gnss1 /home/saurabh/gnss/layers/meta-st/meta-gnss1 7
saurabh@saurabh:~/gnss/build-openstlinuxweston-stm32mp1$
```

Step 6. Use IMAGE_INSTALL:append line at the end of the layer.conf.

```
$vi ../layers/meta-st/meta-st-openstlinux/conf/layer.conf
```

Figure 42. IMAGE_INSTALL:append

```
Linux@linux-sr: ~/sanjay/Distribution-Package/build-openstlinuxweston-stm32mp1
# We have a conf and classes directory, add to BBPATH
BBPATH += "${LAYERDIR}"

# We have a recipes-* directories, add to BBFILES
BBFILES += "${LAYERDIR}/recipes-*//*/*.bb \
${LAYERDIR}/recipes-*//*/*.bbappend \
"

# This folder should only contains specific patches to fix issue on oe recipes
# Note that these patches may be pushed on Community
BBFILES += "${LAYERDIR}/oe-core/recipes-*//*/*.bbappend"

# This folder should only contains direct backport from oe recipes
# These recipes may be suppress at next update on oe version
BBFILES += "${LAYERDIR}/oe-backport/recipes-*//*/*.bb \
${LAYERDIR}/oe-backport/recipes-*//*/*.bbappend \
"

BBFILE_COLLECTIONS += "st-openstlinux"
BBFILE_PATTERN_st-openstlinux := "${LAYERDIR}/"
BBFILE_PRIORITY_st-openstlinux = "5"

LAYERDEPENDS_st-openstlinux = "qt5-layer"

# Set a variable to get the openstlinux location
OPENSTLINUX_BASE = "${LAYERDIR}"

# This should only be incremented on significant changes that will
# cause compatibility issues with other layers
LAYERVERSION_st-openstlinux = "1"
LAYERSERIES_COMPAT_st-openstlinux = "kirkstone"

INHERIT += "check-st-openstlinux-compatibility"
# OpenStLinux compatibility version
ST_OSTL_COMPATIBILITY_VERSION_st-openstlinux = "4.0"
IMAGE_INSTALL:append = "gnss1"
```

Step 7. Delete the meta-gnss1 created by the tool.

```
$ rm -rf ../layers/meta-st/meta-gnss1/
```

Figure 43. Deleting default meta-gnss1

```
saaurabh@saaurabh:~/gnss/build-openstlinuxweston-stm32mp1$ rm -rf ../layers/meta-st/meta-gnss1/
saaurabh@saaurabh:~/gnss/build-openstlinuxweston-stm32mp1$
```

Step 8. Copy the layer provided by the X-LINUX-GNSS1 package.

```
$ cp -rf X-LINUX-GNSS1_V1.1.0/Recipe/meta-gnss1/ ../layers/meta-st/
```

Figure 44. Copying the X-LINUX-GNSS1 meta-st layer

```
saaurabh@saaurabh:~/gnss/build-openstlinuxweston-stm32mp1$ cp -rf X-LINUX-GNSS1_V1.0.0/Recipe/meta-gnss1/ ../layers/meta-st/
saaurabh@saaurabh:~/gnss/build-openstlinuxweston-stm32mp1$
```

Step 9. Add the Sources path (location containing CMakeLists.txt) inside gnss1_0.1.bbappend as shown below.

```
path-to/openstlinux-5.15-yocto-kirkstone-mp1-v22.06.15/Distribution-
Package/build-openstlinuxweston-stm32mp1/x-linux-gnss/X-LINUX-GNSS_V1.X.Y/
Application/Source/gnss_x_linux/Sources
```

Figure 45. Modifying bbappend file

```
saaurabh@saaurabh:~/gnss/build-openstlinuxweston-stm32mp1$ vi ../layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bbappend
```

which is inside layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bbappend:

```
$vi ../layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bbappend
```

Figure 46. Adding external source path in bbappend file

```
linux@linux-sr: ~/sanjay/Distribution-Package/build-openstlinuxweston-stm32mp1
inherit externalsrc
EXTERNALSRC="/home/linux/sanjay/Distribution-Package/build-openstlinuxweston-stm32mp1/X-LINUX-GNSS1_V1.0.0/Application/Source/gnss_x_
linux/sources"
```

Step 10. Update “dunfell” with “kirkstone” inside ‘layers/meta-st/meta-gnss1/conf/layer.conf’.

```
$ vi ../layers/meta-st/meta-gnss1/conf/layer.conf
```

Figure 47. Modifying inside ‘..layers/meta-st/meta-gnss1/conf/layer.conf’

```
linux@linux-sr:~/sanjay1/Distribution-Package/build-openstlinuxweston-stm32mp1$ vi ../layers/meta-st/meta-gnss1/conf/layer.conf
linux@linux-sr:~/sanjay1/Distribution-Package/build-openstlinuxweston-stm32mp1$
```

Figure 48. Update “dunfell ” with “ kirkstone”

```
linux@linux-sr: ~/sanjay1/Distribution-Package/build-openstlinuxweston-stm32mp1
# We have a conf and classes directory, add to BBPATH
BBPATH .= "${LAYERDIR}"

# We have recipes-* directories, add to BBFILES
BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \
           ${LAYERDIR}/recipes-*/*/*.bbappend"

BBFILE_COLLECTIONS += "meta-gnss1"
BBFILE_PATTERN_meta-gnss1 = "^${LAYERDIR}/"
BBFILE_PRIORITY_meta-gnss1 = "7"

LAYERDEPENDS_meta-gnss1 = "core"
LAYERSERIES_COMPAT_meta-gnss1 = "kirkstone"
```

Step 11. Change ‘_’ with ‘.’ inside “layers/meta-st/meta-gnss1/recipes-kernel/linux/linux -stm32mp_%.bbappend” file

```
SRC_URI:append = " file://gnss1.patch"
FILESEXTRAPATHS:prepend := "${THISDIR}/${PN}:"
PACKAGE_ARCH = "${MACHINE_ARCH}"
$ vi ../layers/meta-st/meta-gnss1/recipes-kernel/linux/linux-stm32mp_%.bbappend
```

Figure 49. Modifying inside ‘../layers/meta-st/meta-gnss1/recipes-kernel/linux/linux-stm32mp_%.bbappend’

```
linux@linux-sr:~/sanjay1/STM32MPU_WS/STM32MP1-EcoSy-v4.0.0/Dist-Pkg/build-openstlinuxweston-stm32mp1$ vi ../layers/meta-st/meta-gnss1/recipes-kernel/linux/linux-stm32mp_%.bbappend
```

Figure 50. Updating ‘_’ with ‘.’

```
linux@linux-sr: ~/sanjay1/STM32MPU_WS/STM32MP1-EcoSy-v4.0.0/Dist-Pkg/layers/meta-st/meta-gnss1/recipes-kernel/linux
SRC_URI:append = " file://gnss1.patch"
FILESEXTRAPATHS:prepend := "${THISDIR}/${PN}:"
PACKAGE_ARCH = "${MACHINE_ARCH}"
```

- Step 12.** Delete “install -d \${D}\${libdir}” inside “ layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bb”.

```
# install -d ${D}${libdir}
$ vi ../layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bb
```

Figure 51. Modifying inside ‘../layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1/gnss1_0.1.bb’

```
linux@linux-sr:~/sanjay1/STM32MPU_WS/STM32MP1-EcoSy-v4.0.0/Dist-Pkg/build-openstlinuxweston-stm32mp1$ vi ../layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bb
```

Figure 52. Delete “install -d \${D}\${libdir}”

```
linux@linux-sr:~/sanjay1/STM32MPU_WS/STM32MP1-EcoSy-v4.0.0/Dist-Pkg/layers/meta-st/meta-gnss1/recipes-gnss1/gnss1
#SUMMARY = "bitbake-layers recipe"
#DESCRIPTION = "Recipe created by bitbake-layers"
#LICENSE = "MIT"

#python do_display_banner() {
#    bb.plain("*****");
#    bb.plain("*****");
#    bb.plain("*****");
#    bb.plain("*****");
#    bb.plain("*****");
#}

#addtask display_banner before do_build
LICENSE = "CLOSED"
LIC_FILES_CHKSUM = ""

# No information for SRC_URI yet (only an external source tree was specified)
SRC_URI = ""

DEPENDS = "curl"

inherit cmake

# Specify any options you want to pass to cmake using EXTRA_OECMAKE:
EXTRA_OECMAKE = ""
SRC_URI = ""

OECMAKE_GENERATOR = "Unix Makefiles"
inherit cmake

# Specify any options you want to pass to cmake using EXTRA_OECMAKE:
# EXTRA_OECMAKE = ""
do_install() {
    install -d ${D}${bindir}
    # install -d ${D}${libdir}
    install -m 755 ${D}/gnss_app ${D}${bindir}/
}

--
--
--
```

- Step 13.** Build the ST image.

```
$ bitbake st-image-weston
```

Figure 53. Building the ST image

```
saarabhisaurabh:~/gnss/build-openstlinuxweston-stm32mp1$ bitbake st-image-weston
NOTE: Storing Proberver with doFile: /home/saarabhisaurabh/gnss/build-openstlinuxweston-stm32mp1/cache/proserv.sqlite3, IP: 127.0.0.1, PORT: 40605, PID: 1090706
Bringing recipes: 4h [#####] ETA: 0:04:40
```

New images (including FlashLayout_sdcard_stm32mp157f-dk2-trusted.tsv and FlashLayout_sdcard_stm32mp157f-dk2-trusted) will be created in the tmp-glibc/deploy/images/stm32mp1/ directory.

```
$ cd tmp-glibc/deploy/images/stm32mp1/
```

- Step 14.** Flash the binary following the instructions at https://wiki.st.com/stm32mpu/wiki/STM32MP15_Discovery_kits_Starter_Package#Image_flashing.

- Step 15.** Check if the file below is present on the discovery kit.

```
$ ls -l /dev/ttySTM2
```

- Step 16.** Run the application.

```
$ /usr/bin/gnss_app
```

or

```
gnss_app
```

Revision history

Table 3. Document revision history

Date	Revision	Changes
02-Aug-2021	1	Initial release.
01-Dec-2022	2	Updated Section 2 Hardware setup, Section 4.1.4 Maps and asset tracking, and Section 4.2 Using BitBake (for distribution package).

Contents

1	X-LINUX-GNSS1 overview.....	2
1.1	Features	3
1.2	Architecture	3
1.3	Software package structure	4
1.3.1	gnss_app	5
1.3.2	C utility	6
1.3.3	Python code application	6
2	Hardware setup.....	7
3	Software setup	11
3.1	PC/laptop requirements	11
3.2	Installing the SDK	11
3.3	Downloading the kernel sources (developer package).....	11
3.4	Downloading the distribution package	11
3.5	Connecting to the discovery kit	11
4	Building and running the example	12
4.1	Using Makefile (for starter package).....	12
4.1.1	How to build the gnss_app	12
4.1.2	How to build C utility gnss_uart and gnss_i2c	15
4.1.3	How to run the Python code	17
4.1.4	Maps and asset tracking	18
4.2	Using BitBake (for distribution package)	23
	Revision history	29
	List of tables	31
	List of figures.....	32

List of tables

Table 1.	X-NUCLEO-GNSS1A1 jumper configuration	9
Table 2.	X-NUCLEO-GNSS2A1 jumper configuration	9
Table 3.	Document revision history	29

List of figures

Figure 1.	X-LINUX-GNSS1 user space application architecture	2
Figure 2.	X-NUCLEO-GNSS1A1 hardware connections with required tree modification	3
Figure 3.	X-LINUX-GNSS1 application architecture in Linux environment	4
Figure 4.	X-LINUX-GNSS1_V1.1.0 release package structure - top level	4
Figure 5.	X-LINUX-GNSS_V1.1.0 release package structure - application folders	5
Figure 6.	X-LINUX-GNSS_V1.1.0 release package structure - GNSS other folders	5
Figure 7.	gnss_app folder structure	6
Figure 8.	C utility - gnss-uart and gnss-i2c	6
Figure 9.	Python code application structure	6
Figure 10.	X-NUCLEO-GNSS1A1 expansion board jumper settings	8
Figure 11.	X-NUCLEO-GNSS2A1 expansion board jumper settings	8
Figure 12.	X-NUCLEO-GNSS1A1 and STM32MP157F-DK2 connection	10
Figure 13.	Cloning the package	12
Figure 14.	Sourcing the SDK path	12
Figure 15.	STM32MP157F-DK2 - building the application	13
Figure 16.	gnss_app created	13
Figure 17.	STM32MP157F-DK2 - running the application	14
Figure 18.	Selecting option 11 from the gnss_app	15
Figure 19.	Running gnss_uart	16
Figure 20.	Running gnss_i2c	16
Figure 21.	Installing Python dependencies - pyserial	17
Figure 22.	Installing Python dependencies - pynmea2	17
Figure 23.	Running the Python example	18
Figure 24.	DSH-ASSETTRACKING asset tracking dashboard homepage	18
Figure 25.	DSH-ASSETTRACKING login page	19
Figure 26.	DSH-ASSETTRACKING - Devices tab	20
Figure 27.	DSH-ASSETTRACKING - device name and ID creation	20
Figure 28.	DSH-ASSETTRACKING - new API key generation	20
Figure 29.	DSH-ASSETTRACKING - location of the end-point	21
Figure 30.	Modifying creds.conf	21
Figure 31.	Add credentials in the creds.conf file	21
Figure 32.	Modifying cloud_comm_https.c	21
Figure 33.	cloud_comm_https.c file modified	21
Figure 34.	Running gnss_app and enabling cloud upload (option 20)	22
Figure 35.	Gnss_app sending data to DSH-ASSETTRACKING over http	23
Figure 36.	Gnss_app data in real-time	23
Figure 37.	End user license agreement	24
Figure 38.	gnss package download	24
Figure 39.	meta-gnss1 layer creation	24
Figure 40.	Adding meta-gnss1 layer	25
Figure 41.	Showing all meta layers	25
Figure 42.	IMAGE_INSTALL:append	25
Figure 43.	Deleting default meta-gnss1	26
Figure 44.	Copying the X-LINUX-GNSS1 meta-st layer	26
Figure 45.	Modifying bbappend file	26
Figure 46.	Adding external source path in bbappend file	26
Figure 47.	Modifying inside './layers/meta-st/meta-gnss1/conf/layer.conf'	27
Figure 48.	Update "dunfell" with "kirkstone"	27
Figure 49.	Modifying inside './layers/meta-st/meta-gnss1/recipes-kernel/linux/linux-stm32mp_%.bbappend'	27
Figure 50.	Updating '_' with ':'	27
Figure 51.	Modifying inside './layers/meta-st/meta-gnss1/recipes-gnss1/gnss1/gnss1_0.1.bb'	28
Figure 52.	Delete "install -d \${D}\${libdir}"	28
Figure 53.	Building the ST image	28

IMPORTANT NOTICE – 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 acknowledgment.

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, 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.

© 2022 STMicroelectronics – All rights reserved