UM3037

User manual

Getting started with the X-LINUX-IOT01E/A software for X-STM32MP-IOT01A/X-STM32MP-IOT01E expansion boards plugged onto the STM32MP157F-DK2 discovery kit

Introduction

The X-LINUX-IOT01E/A is an STM32 MPU OpenSTLinux software expansion package that allows developers to evaluate the features of the X-STM32MP-IOT01A/X-STM32MP-IOT01E expansion board.

The package enables the STM32MP157F-DK2 discovery kit to operate with the X-STM32MP-IOT01A/X-STM32MP-IOT01E.

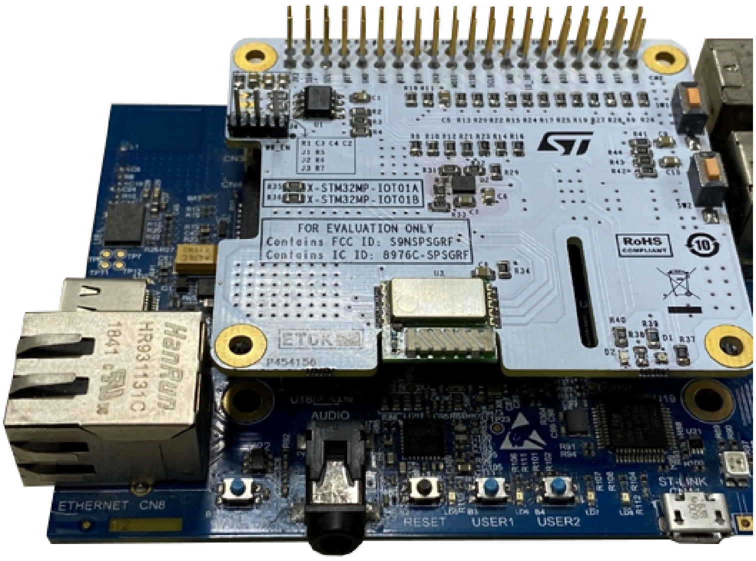
It includes demo applications to evaluate the LSM6DSOX and the SPIRIT1 modules mounted on the X-STM32MP-IOT01A/X-STM32MP-IOT01E expansion board.

The LSM6DSOX is a system-in-package, which features a 3-axis digital accelerometer and a 3-axis digital gyroscope. It boosts performance at 0.55 mA in the high-performance mode and enables always-on low-power features for an optimal motion experience for the consumer.

The SPIRIT1 is a very low-power RF transceiver for RF wireless applications in the Sub-1 GHz band. It is designed to operate both in the license-free ISM and SRD frequency bands at 169, 315, 433, 868, and 915 MHz. It can also be programmed to operate at additional frequencies in the 300-348 MHz, 387-470 MHz, and 779-956 MHz bands.

The X-LINUX-IOT01E/A package integrates the SPIRIT1 library into Linux based software package targeting the STM32MP157F-DK2 board. Thus, it represents a user-mode driver that operates with the SPIRIT1 module.

Figure 1. **X-STM32MP-IOT01A mounted on STM32MP157F-DK2**

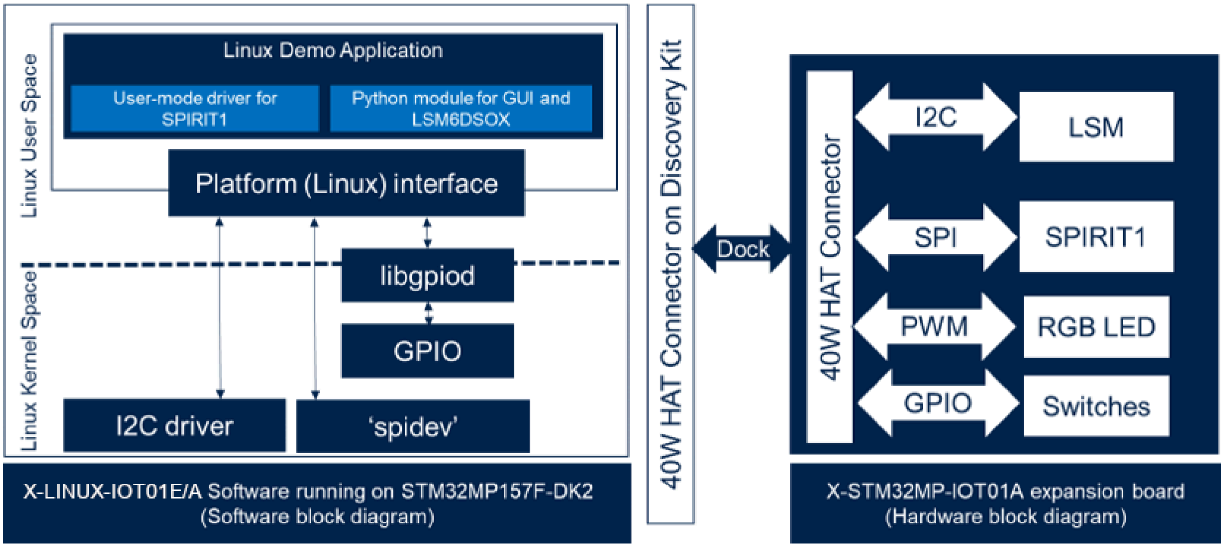


1 Overview

The X-LINUX-IOT01E/A software package runs on the Arm Cortex® A7 core of the STM32MP157 series.

The software interacts with the lower-layer peripherals (I²C and SPI) of the STM32MP1 through standard Linux device drivers.

Figure 2. X-LINUX-IOT01E/A software and hardware architecture



1.1 Functional blocks

The SPIRIT1 user-mode Linux driver communicates with the RF module over a SPI interface. It receives sensor data from a remote sensor node and uploads the data to the DSH-ASSETRACKING dashboard.

A Python-based GUI runs in parallel. It displays the LSM6DSOX sensor data on the screen and provides a button interface to access the on-board RGB LED.

2 Hardware setup

An Ubuntu-based PC/virtual-machine, with at least 18.04 LTS or a higher version, is required as a cross-development platform to build the SPIRIT1 user driver code.

For a quick evaluation using the pre-built binaries provided with the X-LINUX-IOT01E/A package, you can also use a Windows PC.

2.1 Hardware requirements

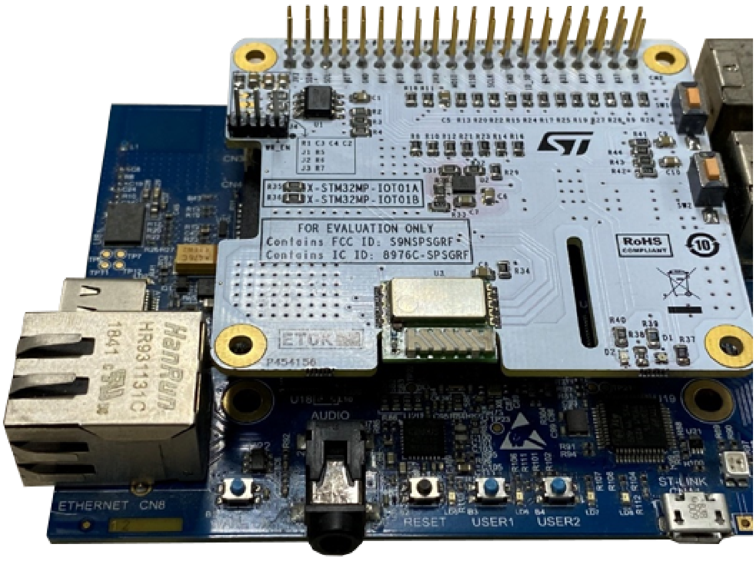
- A PC/virtual machine with Ubuntu® 18.04 or higher
- An STM32MP157F-DK2 discovery kit
- An X-STM32MP-IOT01A/X-STM32MP-IOT01E expansion board
- An 8 GB (or higher) microSD™ card to boot the STM32MP157F-DK2
- An SD card reader or LAN connectivity
- A USB Type-A to micro B USB cable
- A USB Type-A to mini B USB cable
- A USB Type-C to Type-C USB cable
- A USB-PD compliant 5 V 3 A power supply
- A sensor node: NUCLEO-F401RE + X-NUCLEO-IKS01A3 + X-NUCLEO-IDS01A4/X-NUCLEO-IDS01A5

2.2 Hardware connections

2.2.1 Setup for the discovery kit, the expansion board, and the LCD display

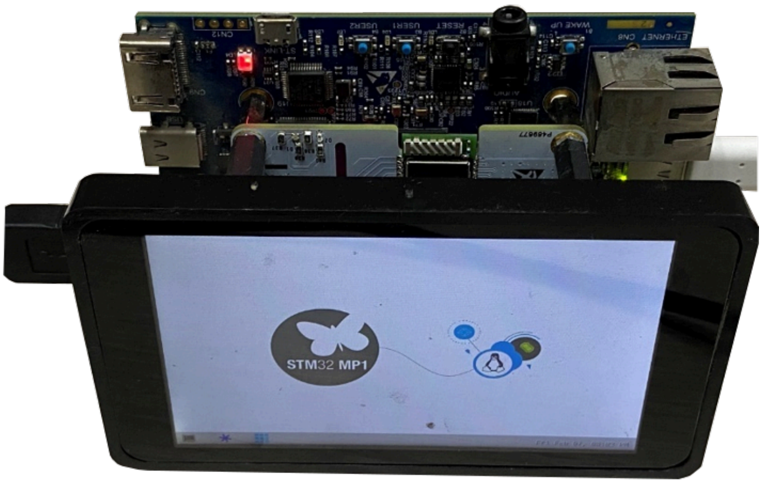
Step 1. Plug the X-STM32MP-IOT01A/X-STM32MP-IOT01E expansion board on the 40 W connector (CN12) of the STM32MP157F-DK2 top side.

Figure 3. **Hardware setup without the LCD display**



Step 2. Connect the LCD display to the STM32MP157F-DK2 on top of the X-STM32MP-IOT01A/X-STM32MP-IOT01E expansion board.

Figure 4. **Hardware setup with the LCD display**



Step 3. Connect the ST-LINK programmer/debugger embedded in the STM32MP157F-DK2 to your host PC via the USB micro-B type port (CN11).

Step 4. Power the STM32MP157F-DK2 through the USB Type-C™ port (CN6). Use a 5 V, 3 A power adapter.

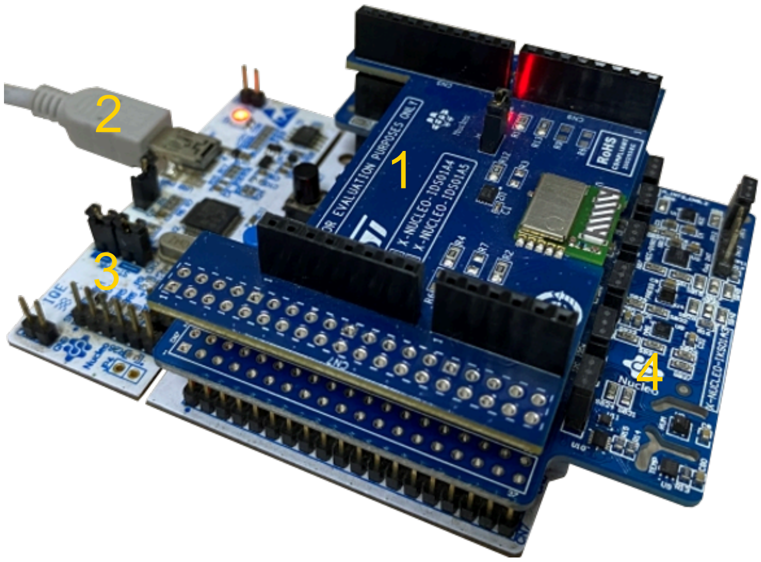
2.2.2 Setup for the sensor node

Step 1. Connect the X-NUCLEO-IKS01A3 and X-NUCLEO-IDS01A4 (for the X-STM32MP-IOT01E)/X-NUCLEO-IDS01A5 (for the X-STM32MP-IOT01A) expansion boards to the NUCLEO-F401RE development board.

Step 2. Use a USB Type-A to mini-B cable to power up and connect this setup to your PC USB port.

Figure 5. **Wireless sensor node**

1. X-NUCLEO-IDS01A4/X-NUCLEO-IDS01A5
2. USB cable for power/programming
3. NUCLEO-F401RE
4. X-NUCLEO-IKS01A3



3 Software setup

Step 1. Unpack the STM32MP157F-DK2 discovery kit, insert a microSD card and power it up using a USB-PD compliant 5 V, 3 A power supply.

Step 2. Install the starter package according to the instructions in the STM32MP157x-DK2 Getting Started wiki.
A minimum 8 GB microSD™ card is required to flash the bootable images.

Step 3. To run the application, the platform should be configured correctly by updating the device tree to enable the relevant peripherals.
- Use the prebuilt images available or
- develop the device tree and build the kernel images.

You can also build this software package by including the Yocto layer (`meta-iot01a`) in the ST distribution package. This operation builds the source code and includes the device tree modifications along with the compiled binaries in the final flashable images. For further details on this process, see Section 3.6 How to include the meta-iot01 layer in the distribution package.

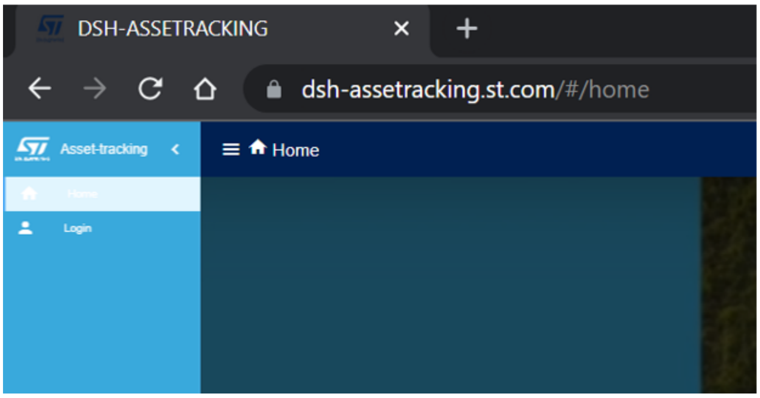
3.1 Using the DSH-ASSETRACKING dashboard

To view the sensor (accelerometer, gyroscope, pressure, and temperature) data, follow the below procedure and create a new cloud device. Then, update the STM32MP157F-DK2 configuration to connect it to the cloud on the DSH-ASSETRACKING dashboard.

Step 1. Open DSH-ASSETRACKING.

Step 2. Go to the DSH-ASSETRACKING login page and create a new login account.

Figure 6. **Opening DSH-ASSETRACKING dashboard**

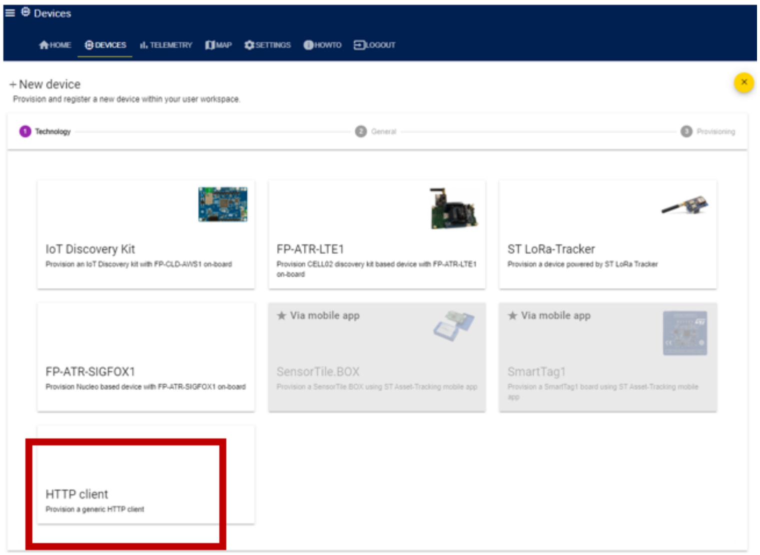


Step 3. After logging in, create a new cloud device by clicking on [**Devices**]>[**NEW**].

Figure 7. **Creating a new cloud device**

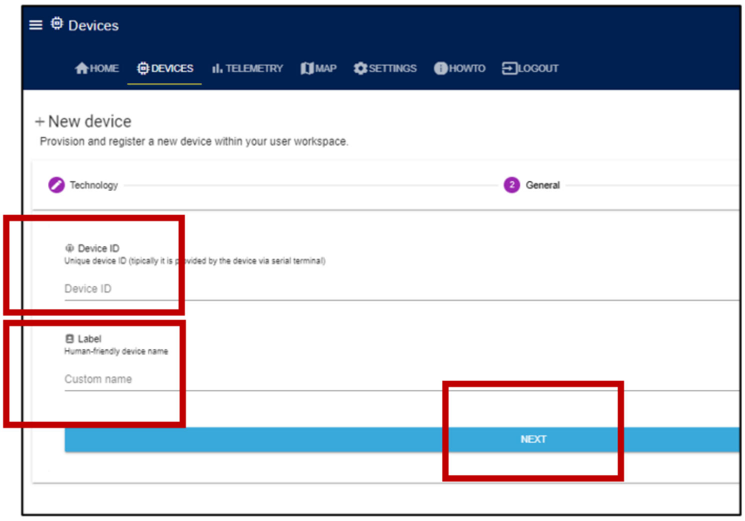
Step 4. Select the device type as “HTTP client”.

Figure 8. **Selecting the device type**



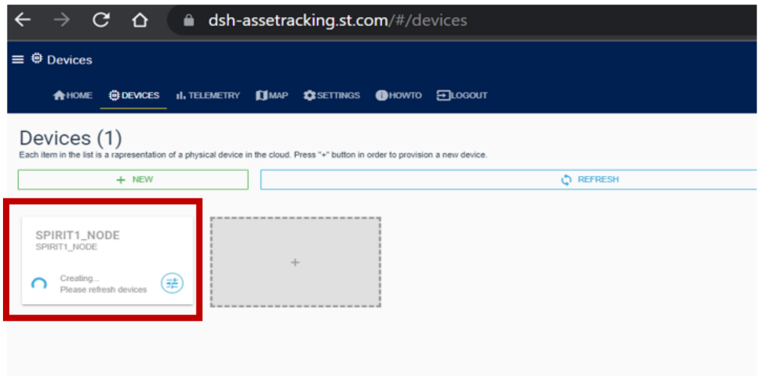
Step 5. Insert an appropriate device ID and label. Then, click on [**NEXT**].

Figure 9. Inserting the device ID and label



The device ID that you have just inserted becomes the device unique ID and a new cloud device is created. As an example, the figure below shows a new device with device ID `SPIRIT1_NODE`.

Figure 10. New device ID

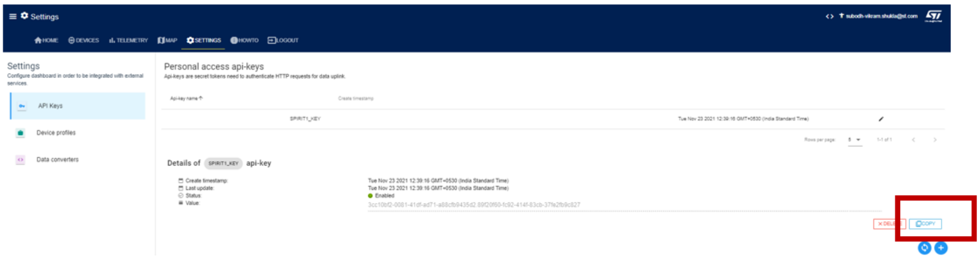


Step 6. Go to [**SETTINGS**] and create a new API key by clicking on the [**+**] button.
The API key is common for all the nodes.

Figure 11. Creating a new API key

Step 7. Copy the API key value by clicking on the [**Copy**] button.

Figure 12. Copying the API key value



This API key and the Device ID should be added to the creds.conf file on the STM32MP157F-DK2 (refer to Section 3.5 , step 5).

3.2 Sensor node setup

To program “STM32F401RE-Nucleo_P2P_SensorNode_Tx.bin” via the STM32CubeProgrammer on the NUCLEO-F401RE, follow the procedure below.

Step 1. Download and install STM32CubeProgrammer.

Step 2. Plug the hardware as shown in Figure 5.

Step 3. Connect the board using a programming cable (USB mini-B to type A).

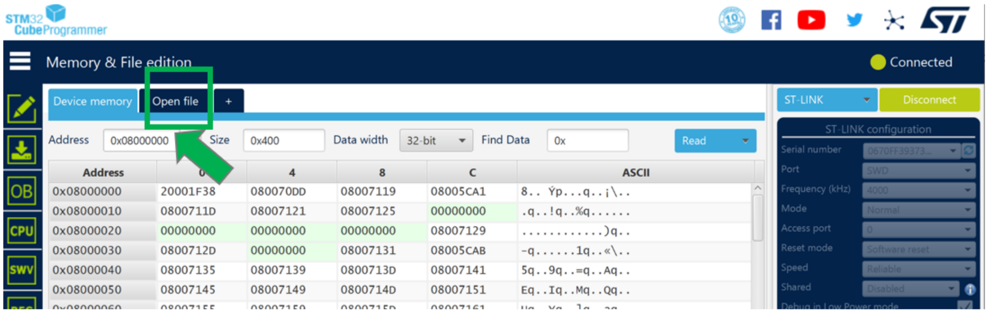
Step 4. Run STM32CubeProgrammer on your Windows/Ubuntu PC.

Step 5. Select [**ST-LINK**] from the drop-down menu and click on the [**Connect**] button.

Figure 13. Connecting target from STM32CubeProgrammer

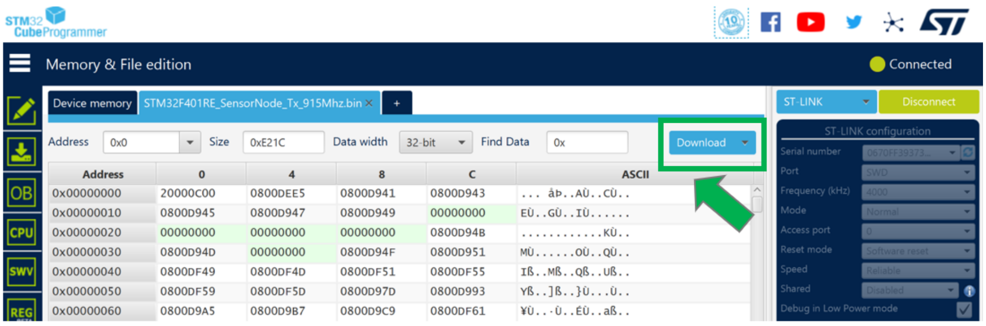
Step 6. Select the “STM32F401RE_SensorNode_Tx_915Mhz.bin” file for X-NUCLEO-IDS01A5 and “STM32F401RE_SensorNode_Tx_868Mhz.bin” file for X-NUCLEO-IDS01A4 from the [**Open file**] menu.

Figure 14. Binary file selection



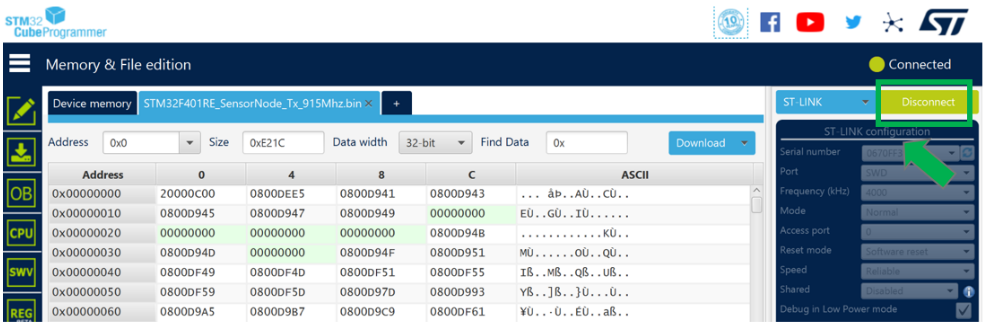
Step 7. Program the device by clicking on the [**Download**] button.

Figure 15. Programming the device



Step 8. Disconnect the device by clicking on the [**Disconnect**] button.

Figure 16. Disconnecting the device



3.3 How to update the platform configuration using the developer package

Follow the procedure below to set up the development environment.

Step 1. Download the developer package and install the SDK in the default directory structure on your Ubuntu machine.

Step 2. Open the device tree file 'stm32mp157f-dk2.dts' in the developer package source code and add the code snippet below to the file (in arch/arm/boot folder).

```
&i2c5 {
    status = "okay";
    lsm6dsox@6b {
        compatible = "st,lsm6dsox";
        reg = <0x6b>;
        st,int-pin = <1>;
        interrupt-parent = <&gpioa>;
        interrupts = <8 IRQ_TYPE_LEVEL_HIGH>;

        disable-shub;
        status = "okay";
    };
};

&spi5 {
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&spi5_pins_a>;
    pinctrl-1 = <&spi5_sleep_pins_a>;
    /*status = "disabled";*/
    cs-gpios = <&gpiof 6 0>;
    status = "okay";

    spidev@0 {
        compatible = "semtech,sx1301";
        spi-max-frequency = <5000000>;
        reg = <0>;
    };
};
```

This updates the device tree to enable and configure the I2C5 and SPI4 driver interface.

Note: Only the 'tab' character is used for indentation in the source files of the device tree. No 'space' character is used.

Step 3. Copy the patch for the LSM6DSOX kernel driver patch in the developer package folder.

```
PC $> cd X-LINUX-IOT01A/ STM32MP157F-DK2_DeviceDriver_LSM6DSOX/
PC $> cp *.patch <ST patches path>
```

Step 4. Compile the developer package to get the updated stm32mp157f-dk2.dtb and st_lsm6dsox.ko files and reflash the board with the new image (created using the developer package).

Refer to Modify, rebuild and reload the Linux® kernel for help.

Step 5. Use the below commands to make the new configuration active on the discovery kit.

```
Board $> /sbin/depmod -a
Board $> sync
Board $> reboot
```

Step 6. After reboot, to verify whether the new configuration has been correctly set:

- check if "/dev/spidev0.0" file has been created;
- check the content of the "/sys/bus/iio/devices/" folder. It must contain the information shown below.

Figure 17. /sys/bus/iio/devices/ folder content

3.4 How to build the SPIRIT1 Linux user driver and Application using ST SDK

Step 1. Download and install the ST SDK from .

Step 2. Download the X-LINUX-IOT01E/A package.

Step 3. Execute the below commands to cross-compile the code.

```
PC $> sudo apt-get install cmake
PC $> source <SDK_install_path>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-
linux-gnueabi
PC $> cd X-LINUX-IOT01\SPIRIT1Application\Source
PC $> tar xvf SPIRIT1Application-source.tar.xz
PC $> cd linux_demo\build
PC $> cmake ..
PC $> make
```

These commands cross-build the c application (‘spirit_application_binary’) in the folder ‘X-LINUX-IOT01A/linux-demo/build/spirit_application/’.

Step 4. Copy the generated binary (spirit_application_binary) to the /usr/local/x-linux-iot01 folder on the STM32MP157F-DK2 discovery kit.

Step 5. Use the complete package as described in Section 3.5 How to evaluate the software package.

3.5 How to evaluate the software package

Step 1. Boot the STM32MP157F-DK2 with the starter package.

Step 2. Enable the internet connectivity on the STM32MP157F-DK2 via Ethernet or Wi-Fi.
Refer to the wiki page for help on the Wi-Fi connection setup.

Step 3. Download the X-LINUX-IOT01E/A software package.

Step 4. Modify the kernel using the steps mentioned in section 3.4.

Step 5. Update the file /usr/local/x-linux-iot01/creds.conf with the board_type (X-STM32MP-IOT01A or X-STM32MP-IOT01E) and the new `API_KEY` and `DEVICE_ID` generated in Section 3.1 , step 7.

Figure 18. Updating the creds.conf file

Step 6. Set up the sensor node hardware (Section 2 Hardware setup) and software (Section 3.2 Sensor node setup).

Step 7. Power off and power on the sensor node.

As soon as the sensor node is powered up, it starts transmitting multiple sensor data packets (accelerometer, gyroscope, pressure, and temperature) every 5 seconds.

Step 8. Configure and open the DSH-ASSETRACKING dashboard as described in Section 3.1 Using the DSH-ASSETRACKING dashboard.

Step 9. Install the dependencies on the STM32MP157F-DK2 using the below commands.

```
Board $> apt-get install python3-pip
Board $> pip3 install requests
```

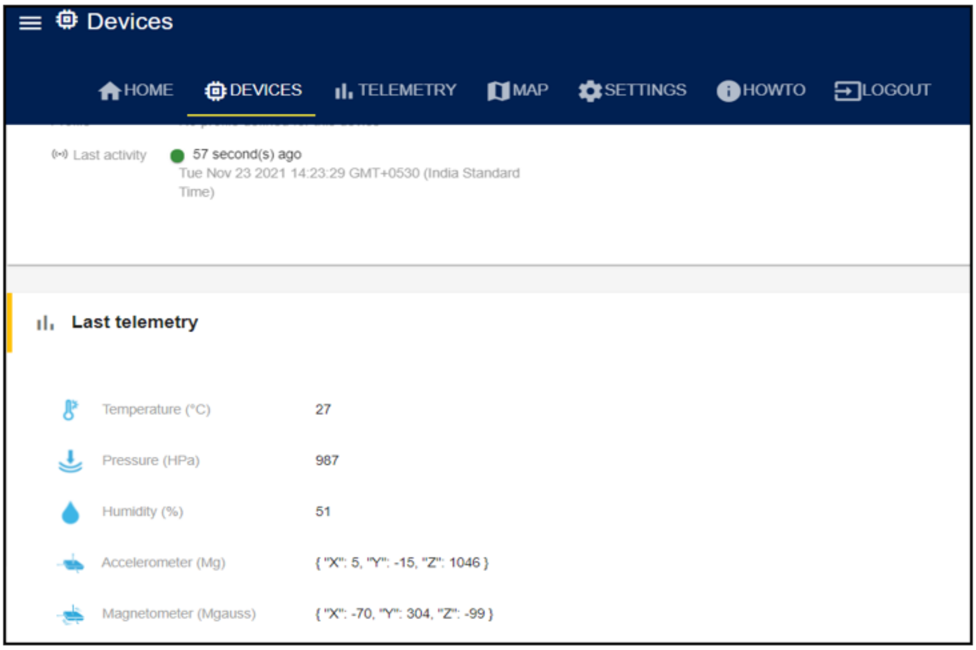
Step 10. Open the terminal on the STM32MP157F-DK2 via ssh or Tera Term and execute the below commands to run the demo.

```
Board $> cd /usr/local/script
Board $> ./StartDemo.sh
```

Note: If the network connectivity is not available, refer to Section 4 to transfer the file locally. You can also use any other tool for this purpose.

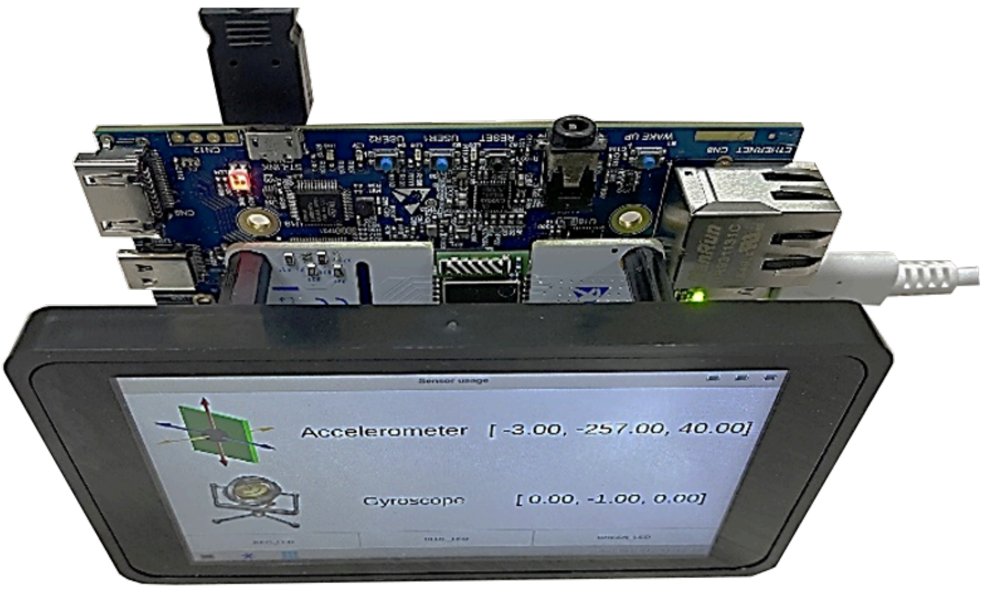
The application starts running after the above commands execution. Then, a GUI runs on the display connected to the STM32MP157F-DK2, showing the accelerometer and gyroscope data from the on-board sensor.

Figure 19. Accelerometer and gyroscope data



A background process handles the data over the wireless (SPIRIT1) interface and the cloud. Hence, the periodic dataset coming from the sensor node is visible on the DSH-ASSETRACKING dashboard.

Figure 20. Sensor data shown on the dashboard



3.6 How to include the meta-iot01 layer in the distribution package

Step 1. Download and compile the distribution package on your Ubuntu PC or virtual machine.

Step 2. Follow the default directory structure to synchronize with the below commands.

```
PC$> tar -xvf X-LINUX-IOT01.tar.xz
PC$> cd X-LINUX-IOT01
PC$> cp -rf meta-iot01/ STM32MP15-Ecosystem-v3.1.0/Distribution-Package/
openstlinux-5.10-dunfell-mp1-21-11-17/layers
PC$> cd STM32MP15-Ecosystem-v3.1.0/Distribution-Package/openstlinux-5.10-dunfell-
mp1-21-11-17/
```

Step 3. Download and extract the X-LINUX-IOT01E/A software application package.

Step 4. Set up the open embedded build environment configuration.

```
PC$> DISTRO=openstlinux-weston MACHINE=stm32mp1 source layers/meta-st/scripts/
envsetup.sh
```

Step 5. Add the meta-iot01a layer to the build configuration of the distribution configuration.

```
PC$> bitbake-layers add-layer ../layers/meta-iot01
```

Step 6. Update the build configuration (‘st-image-weston’), to add new components to the image.

```
PC$> echo 'IMAGE_INSTALL_append += "iot01"' >> meta-st/meta-st-openstlinux/conf/
layer.conf
```

Step 7. Build your layer separately and then build the complete distribution layer.

```
PC$> bitbake st-image-weston
```

Building the distribution page for the first time might take several hours. However, it takes only few minutes to build the meta-iot01ea layer and install the executable files in the final images. Once the build completes, the images are included in the *build-<distro>-<machine>/tmp-glibc/deploy/images/stm32mp1* directory.

Step 8. Follow the instructions to flash the new built images onto the discovery kit.

Step 9. Run the application as mentioned in Section 3.5 How to evaluate the software package, steps 4-10.

4 How to transfer files using Tera Term

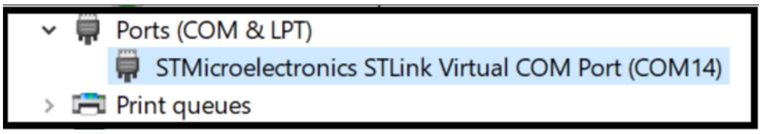
You can use any terminal emulator application. Tera Term is one of them. It can be downloaded from http://tera-term.en.lo4d.com. Follow the below steps to configure Tera Term.

Step 1. Plug the power cable to power the board.

Step 2. Connect the STM32MP157F-DK2 to your PC via a USB micro-B cable through CN11.

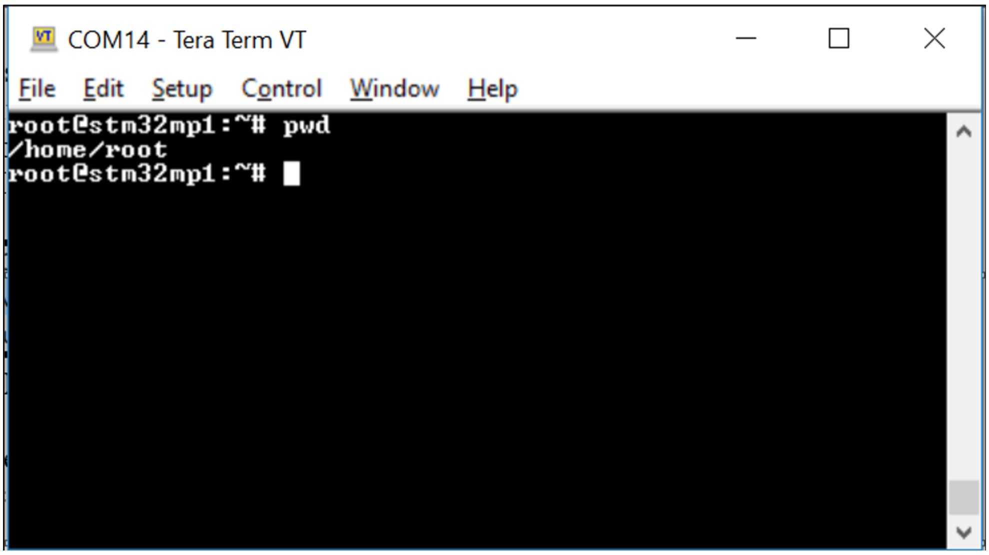
Step 3. Check the virtual COM port number in the device manager.
For example, in the snapshot below, the COM port number is 14.

Figure 21. Device manager: virtual COM port



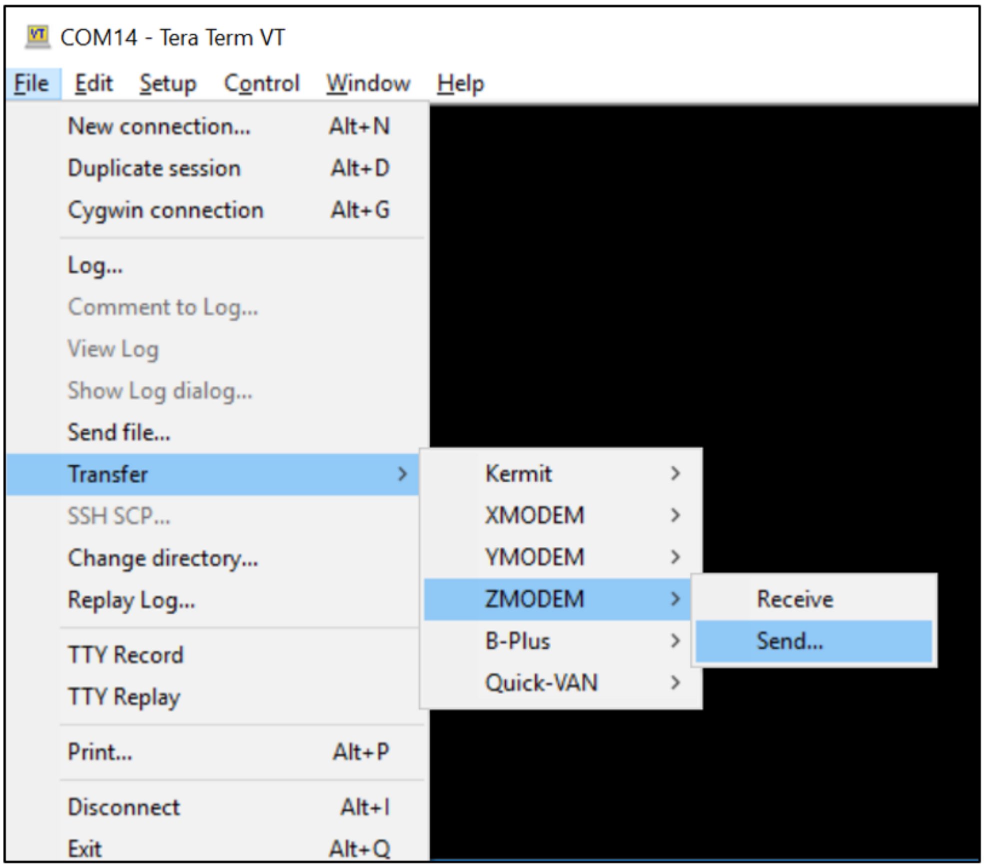
Step 4. Open Tera Term on your PC and select the COM port.
The baud rate should be 115200. The virtual terminal (remote access) appears as shown below.

Figure 22. Remote terminal via Tera Term



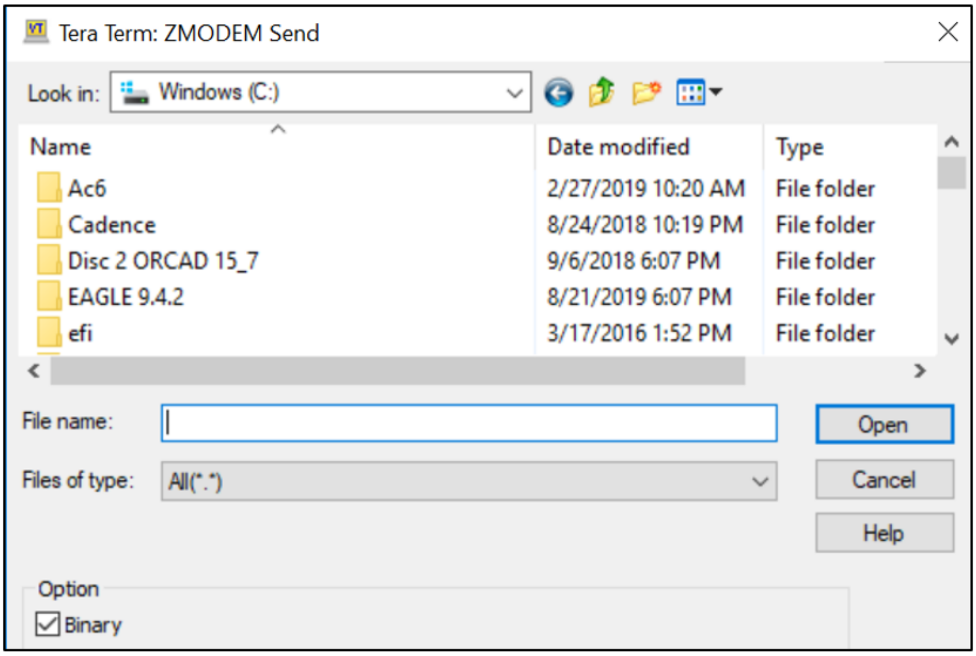
Step 5. To transfer a file from the host PC to the STM32MP157F-DK2, click on the [**File**] menu at the top-left corner of the Tera Term window. Then go to [**File**]>[**Transfer**]>[**ZMODEM**]>[**Send**].

Figure 23. Transferring the file



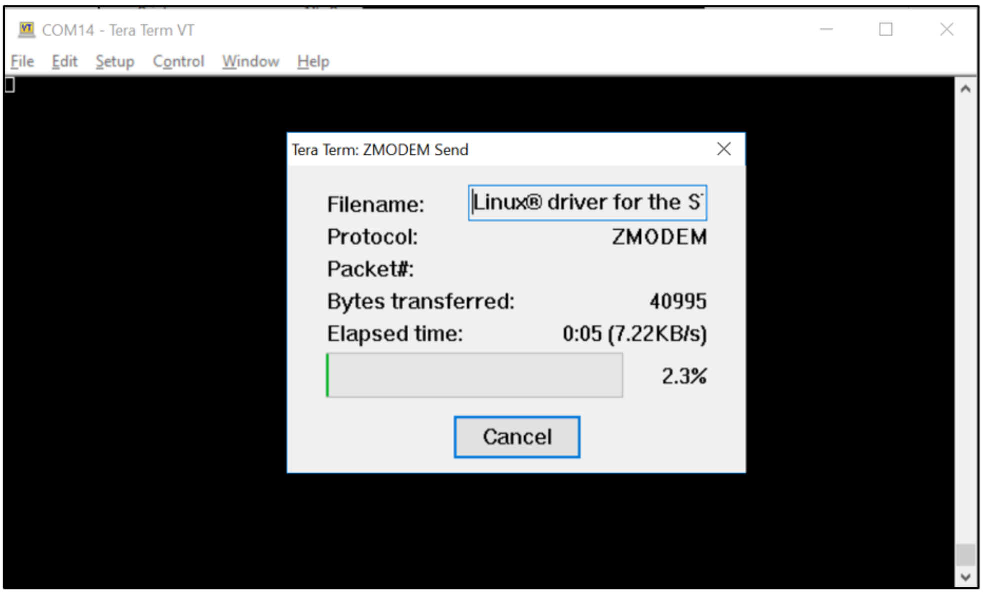
Step 6. Select the file to transfer from the pop-up window. Click on the [**Open**] tab.

Figure 24. Pop-up window to select one or more files to transfer



A progress bar shows the file transfer status.

Figure 25. Progress bar



Revision history

Table 1. **Document revision history**

Date	Revision	Changes
16-Jun-2022	1	Initial release.
01-Dec-2022	2	Updated introduction, Section 1.1 Functional blocks, Section 2.1 Hardware requirements, Section 2.2.1 Setup for the discovery kit, the expansion board, and the LCD display, Section 2.2.2 Setup for the sensor node, Section 3 Software setup, Section 3.1 Using the DSH-ASSETRACKING dashboard, Section 3.2 Sensor node setup, Section 3.3 How to update the platform configuration using the developer package, Section 3.4 How to build the SPIRIT1 Linux user driver and Application using ST SDK, Section 3.5 How to evaluate the software package, and Section 4 How to transfer files using Tera Term.

Contents

List of tables

List of figures