

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

X-STM32MP-IOT01E expansion board.

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/

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

57

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.





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

51

An Ubuntu-based PC/virtual-machine, with at least 18.04 LTS or a higher version, is required as a crossdevelopment 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

57/

- 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-iot0la) 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



Figure 7. Creating a new cloud device



Step 4. Select the device type as "HTTP client".

chinology	Central	Another and the second seco
IOT Discovery Kit Provision an IoT Discovery Mt with FP-CLD-AWS1 on-board	PP-ATR-LTE1 Promition CELLI2 discovery Mit based device with FP-ATR-LTE1 on-based	ST LoRa-Tracker Provision a device powered by ST LoRa Tracker
	★ Via mobile app	* Via mobile app
FP-ATR-SIGFOX1 Provision Nucleo based device with FP-ATR-SIGFOX1 on 8	SensorTile.BOX Provision a SensorTile BOX using ST Asset-Tracking mobile app	SmartTag1 Provision a SmartTag1 board using ST Asset-Tracking mobile app

Figure 8. Selecting the device type



-	DEVICES	II. TELEMETRY	Юмар	SETTINGS	вножто	➡LOCOUT	
lew device	ter a new devi	ce within your use	r workspace				
Technology		, your use		-		2 General	
(i) Davica ID							
w Device ID							
Unique device ID ((tpically it is provide	d by the device via seria	l terminal)				
Unique device ID (Device ID	tipically it is provide	d by the device via seria	l terminal)				
Device ID Device ID Label Human-friendly de	ivice name	d by the device via seria	l terminal)				
Device ID Device ID E Label Human-friendly de Custom name	itipically it is p bloe	d by the device via sena	(terminal)				
Unique device ID I Device ID E Label Human-friendly de Custom name	vice name	d by the device via sena	(terminal)			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 <code>SPIRIT1_NODE</code>.

Figure 10. New device ID

←	\rightarrow (C D	🗎 dsh-	assetra	cking.st.c	om/ #/d	devices
≡⊜	Devices						
	A HOME	C DEVICE	ES II. TELEMETRY	Юмар	SETTINGS	Пноwто	0 Erocout
De Each if	VICES term in the list	(1) is a rapresenta	tion of a physical device	in the cloud.	Press "+" button in	order to provisi	ision a new device.
		+ NEW	/				REFRESH
SI SF	PIRIT1_N PIRIT1_NODE Creating Please re	ODE	≆		+		

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

		<> T suboth-vitram.shuitia	gat com
() шир Саяттика (Сноито Е).0000/Г			
Personal access api-keys apikeys are secret taken need to authenticate HTTP requests for data uplice.			
ApJay rang 0	Create timestamp		
	No data available	Rossperpage: <u>6 •</u> -	$\langle \rangle$
		[00
	Плик Флатика Реголо Воссеза вр:Керз съста, на инализия ната въемски ИТИ прави ва кан циск. дили има?	Operatives Operatives Operatives Perconditional scales significants States are surface and states in State spaces. States are surface and states in State spaces. Applications States are surface. States in states in State spaces.	O Table Annu Annu

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

Figure 12. Copying the API key value

≡ ♥ Settings			() Takah	ekram shuklağırl.com
AHOME ODMICES IL TELEMETRY	(Димар 😄 зетлика 🕢 номто 🖃 Locout			
Settings Configure dishiboard in order to be integrated with external services.	Personal access api-keys Api-keys are secret tokens need to authenticate HTTP requests for data u	den.		
👟 API Keys	Apikey name 1	Create Snesterp		
	SPIRITI_KEY		Tue Nov 23 2021 12:39:16 ON/T+0530 (India Standard Time)	/
Device profiles			Rom per paga 🛛 5 👻	statt < >
O Data converters	Details of seamulary api-key			
	⊟ Create Envestamp: □ Least update: Status:	Twe Nov 23 2021 12:39:16 GMT=4530 (india Standard Time) Twe Nov 23 2021 12:39:16 GMT=6530 (india Standard Time)		
	Value:	3cc1082-0081-41dF-ad71-a88ctb9435d2.89(20161-fc92-4145-83cb-3714289c827		
				X DELS E
				0 🕀

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

<u>10</u>	1 🕒 🔰	× 🖅
		Not connected
	ST-LINK	Connect
	ST-LIN	K configuration
•	Serial number	0670FF39373 👻 🧭
	Port	SWD 👻
	Frequency (kHz)	4000 👻
	Mode	Normal 👻
	Access port	0 👻
	Reset mode	Software reset 🔹
	Speed	Reliable 👻
	Shared	Disabled 🔹 👔
	Debug in Low Pov	ver mode 🗸 🗸
	External loader	_

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.

STM32 Cube	Programmer						1	1 🖸 🔰	* 🖅
≡	Memory & File e	dition							Connected
	Device memory	Open file +						ST-LINK	 Disconnect
	Address 0x08000	000 Size	0x400	Data width 32-	bit 👻 Find D	Data Ox Read	•	ST-LIN Serial number	K configuration 0670FF39373 🖸
=	Address		4	8	с	ASCII		Port	SWD 👻
OB	0x08000000	20001F38	080070DD	08007119	08005CA1	8 Ýpq;\	_	Frequency (kHz)	4000 👻
	0x08000010	0800711D	08007121	08007125	00000000	.q!q%q		Mode	Normal
CPU	0x08000020	00000000	00000000	00000000	08007129)q		Access port	0
╘	0x08000030	0800712D	00000000	08007131	08005CAB	-q1q«\		Reset mode	Software reset 🔹
swv	0x08000040	08007135	08007139	0800713D	08007141	5q9q=qAq		Speed	Reliable 👻
	0x08000050	08007145	08007149	0800714D	08007151	EqIqMqQq		Shared	Disabled 👻 👔
								Data in Low Data	

Figure 14. Binary file selection

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

Figure 15. Programming the device

STM32 CubePr	rogramme	er											(19)	f 🖸	3	\star	57
	Memory	v & Fil	e editio	n												Conne	ected
	Device m	nemory	STM32F	401F	RE_Sen	sorNode_Tx_91	5Mhz.bin × +							ST-LIN	чK	- Disc	onnect
	Address	0x0		•	Size	0xE21C	Data width	32-b	pit 👻	Find Data	0x		Download 🔻	Serial n	ST-LINK umber	configuratio	on 5 👻 🛃
	Ad	dress		0		4	8		(:	AS	scii	-	Port		SWD	
OB	0x00000	0000	200	00C0	00	0800DEE5	0800D941		0800D9	43	. åÞAÙCÙ			Freque		4000	*
	0x00000	0010	080	0D94	45	0800D947	0800D949		000000	00 EÙ	GÙIÙ			Mode		Normal	*
CPU	0x00000	020	000	0000	00	00000000	0000000		0800D9	4B	КÙ			Access		0	*
	0x00000	0030	080	0D94	4D	00000000	0800D94F		0800D9	51 MÙ	QÙQÙ			Reset n		Software rese	et 👻
swv	0x00000	0040	080	0DF4	19	0800DF4D	0800DF51		0800DF	55 Iß	MBQBUB			Speed		Reliable	*
	0x00000	050	080	ODF5	59	0800DF5D	0800D97D		0800D9	93 Yß]ß}ÙÙ			Shared		Disabled	• •
REG	0x00000	0060	080	0D9A	45	0800D9B7	0800D9C9		0800DF	61 ¥Ù	ÙÉÙaß			Debug			

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

Figure 16. Disconnecting the device

STM32 Cube	2 Programmer							1	f 🕒	🖌 🖈	57
≡	Memory & F	ile editio	n							Conn	ected
	Device memor	y STM32F	401RE_Se	nsorNode_Tx_91	5Mhz.bin × +				ST-LINK	👻 Disc	connect
	Address 0x0)	▼ Size	0xE21C	Data width 32	e-bit 👻 Find D	Data Ox Dow	nload 👻	ST-LI Serial number	NK configuratio	on Con
	Address		0	4	8	с	ASCII		Port	SWD	
OB	0x00000000	2000	00000	0800DEE5	0800D941	0800D943	åÞAÙCÙ	^	Frequency (kHz)	4000	
	0x0000010	080	DD945	0800D947	0800D949	00000000	EÙGÙIÙ		Mode	Normal	~
CPU	0x00000020	0000	00000	00000000	00000000	0800D94B	KÙ		Access port	0	~
	0x0000030	080	0D94D	00000000	0800D94F	0800D951	MÙOÙQÙ		Reset mode	Software res	et 🔻
swv	0x00000040	080	DDF49	0800DF4D	0800DF51	0800DF55	IßMBQBUB		Speed	Reliable	~
	0x00000050	080	DDF59	0800DF5D	0800D97D	0800D993	Yß]ß}ÙÙ		Shared	Disabled	- 0
REG	0x00000060	080	DD9A5	0800D9B7	0800D9C9	0800DF61	¥ÙÙÉÙaß		Debug in Low P		

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 = \langle 0x6b \rangle;
        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\SPIRITIApplication\Source
PC $> tar xvf SPIRITIApplication-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

Creds.conf - Notepad						×
File Edit Format View Help						
<pre>http_endpoint = https://jim3rgi6d3</pre>	.execute-api.eu-central-1.	amazonaw	s.com/v1,	/tele	metr	у ^
api_key = 00b23ea3-bc57-5bca-d8e9-	dff8d7f3e8e8.ae654c67-8725	-8ac1-0f	3a-d5bd7	35103	4d	
device_id = My_Node_1						
 board_type = X-STM32MP-IOT01A						
						~
	Ln 6 Col 1	100%	Unix (LF)	UTF	-8	

- 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.

VT	COM1	4 - Tera	Term VT					×
<u>F</u> ile	<u>E</u> dit	<u>S</u> etup	C <u>o</u> ntrol	<u>W</u> indow	<u>H</u> elp			
roo Zho	t@stm me∕ro	32mp1: ot	* *# pwd					^
r00	testm	32mp1	:~#					
¢								
								\sim

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].

VT	COM14 - Tera Term VT				
<u>F</u> ile	<u>E</u> dit <u>S</u> etup C <u>o</u> ntrol	<u>W</u> indow	<u>H</u> elp		
	New connection	Alt+N			
	Duplicate session	Alt+D			
	Cygwin connection	Alt+G			
	Log				
	Comment to Log				
	View Log				
	Show Log dialog				
	Send file				
	Transfer	>	Kermit	>	
	SSH SCP		XMODEM	>	
	Change directory		YMODEM	>	
	Replay Log		ZMODEM	>	Receive
	TTY Record		B-Plus	>	Send
	TTY Replay		Quick-VAN	>	
	Print	Alt+P			
	Disconnect	Alt+I			
	Exit	Alt+Q			

Figure 23. Transferring the file



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

Itera Term: ZMODEM Send		×
Look in: 🔛 Windows (C:) 🗸 🗸	G 🤌 📂 🛄 🗸	
Name Ac6 Cadence Disc 2 ORCAD 15_7 EAGLE 9.4.2 efi	Date modified 2/27/2019 10:20 AM 8/24/2018 10:19 PM 9/6/2018 6:07 PM 8/21/2019 6:07 PM 3/17/2016 1:52 PM	Type File folder File folder File folder File folder File folder
< File name: Files of type: All(*.*) Option Binary		> Open Cancel Help

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

A progress bar shows the file transfer status.

M	COM1	4 - Tera	Term VT			_	_		\times
<u>F</u> ile	<u>E</u> dit	<u>S</u> etup	C <u>o</u> ntrol	Window	<u>H</u> elp				
									^
					Tera Term: ZMODEM Send	×			
					The second se	. .			
					Filename: Linux® driver for the s)			
					Protocol: ZMODEI	M			
					Packet#:				
					Bytes transferred: 4099	5			
					Elapsed time: 0:05 (7.22KB/s	5)			
					2.29	~			
					2.3/	6			
					Canaal				
					Cancel				
									Ť

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.

57

Contents

1	Overview						
	1.1	Functi	onal blocks	2			
2	Hard	ware s	etup	3			
	2.1	Hardw	are requirements	3			
	2.2	Hardware connections					
		2.2.1	Setup for the discovery kit, the expansion board, and the LCD display	3			
		2.2.2	Setup for the sensor node	4			
3	Software setup						
	3.1	Using	the DSH-ASSETRACKING dashboard	5			
	3.2	Senso	r node setup	8			
	3.3	How to	o update the platform configuration using the developer package	9			
	3.4	How to	b build the SPIRIT1 Linux user driver and Application using ST SDK	11			
	3.5	How to	o evaluate the software package	11			
	3.6	How to	o include the meta-iot01 layer in the distribution package	14			
4	How	to tran	sfer files using Tera Term				
Rev	ision	history	·				
List	of tab	oles					
List	of fig	ures					

List of tables

Table 1. Document revision history	18
--	----

List of figures

Figure 1.	X-STM32MP-IOT01A mounted on STM32MP157F-DK2	1
Figure 2.	X-LINUX-IOT01E/A software and hardware architecture	2
Figure 3.	Hardware setup without the LCD display	3
Figure 4.	Hardware setup with the LCD display	4
Figure 5.	Wireless sensor node	4
Figure 6.	Opening DSH-ASSETRACKING dashboard	5
Figure 7.	Creating a new cloud device.	6
Figure 8.	Selecting the device type	6
Figure 9.	Inserting the device ID and label	7
Figure 10.	New device ID	7
Figure 11.	Creating a new API key	7
Figure 12.	Copying the API key value	8
Figure 13.	Connecting target from STM32CubeProgrammer	8
Figure 14.	Binary file selection	9
Figure 15.	Programming the device	9
Figure 16.	Disconnecting the device	9
Figure 17.	/sys/bus/iio/devices/ folder content	10
Figure 18.	Updating the creds.conf file	12
Figure 19.	Accelerometer and gyroscope data	13
Figure 20.	Sensor data shown on the dashboard	13
Figure 21.	Device manager: virtual COM port	15
Figure 22.	Remote terminal via Tera Term	15
Figure 23.	Transferring the file	16
Figure 24.	Pop-up window to select one or more files to transfer	17
Figure 25.	Progress bar	17

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