life.augmented

Quick Start Guide for X-LINUX-IGWWSN1

*Industrial edge gateway software package for wireless sensor networks (WSN).

x-linux-igwwsn1_qsg v1 2024-12-20

Contents

0. Prerequisites
1. Edge GW Set up
2. OS Installation
3. X-LINUX-AZURE Installation
4. Zigbee WSN Set up
5. X-LINUX-IGWWSN1
6. Application

Prerequisites

Requirements

Hardware

- STM32MP257F-EV1 cut 2.0 board
- Micro SD 32GB
- 5 STEVAL-PROTEUS1 boards (at least 2 boards)
- STLINK-V3 or STLINK-V3MINIE
- Power supply classic or USB type-C 5V@3A

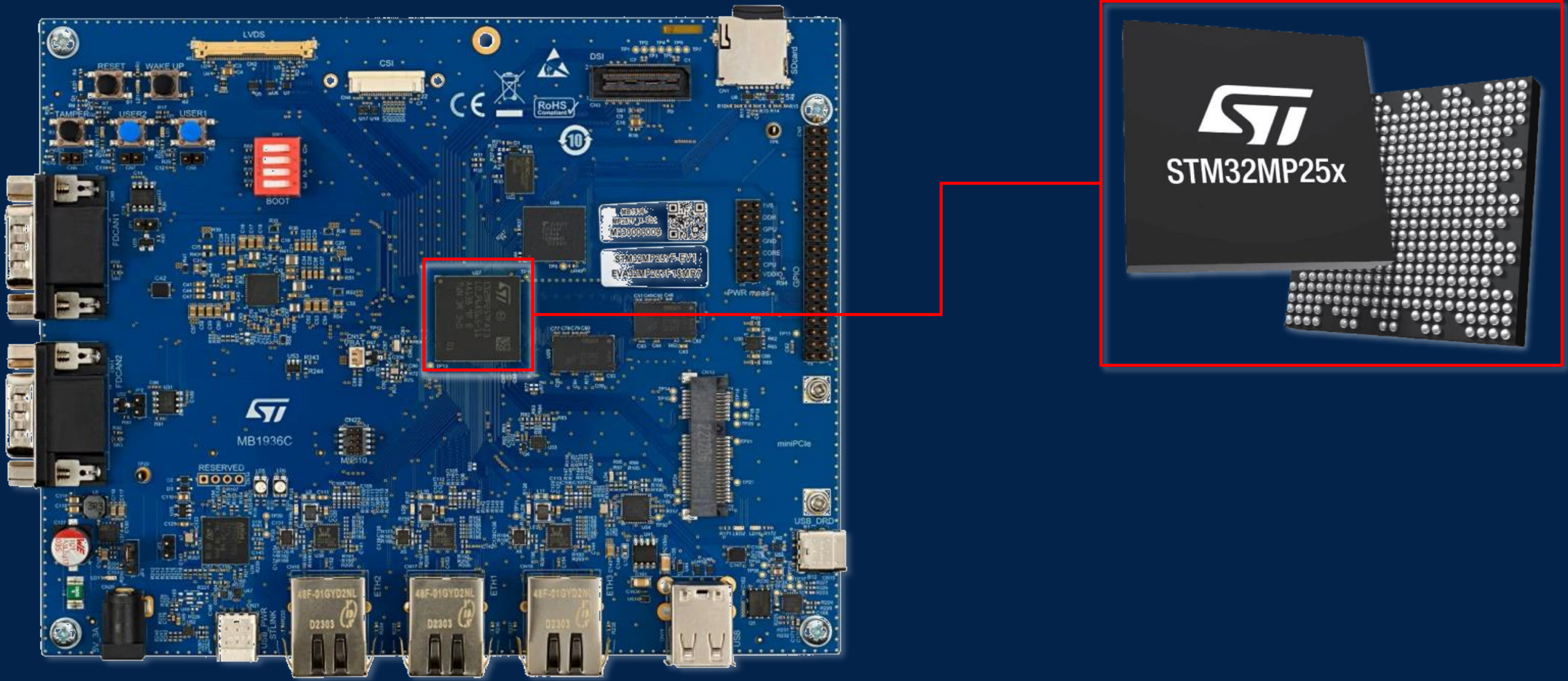
Software

- STM32CubeProgrammer v.2.18 or higher
- OpenSTLinux Distribution v6.0
- X-LINUX-AZURE v5.1
- STSW-PROTEUS v1.1.1

Misc

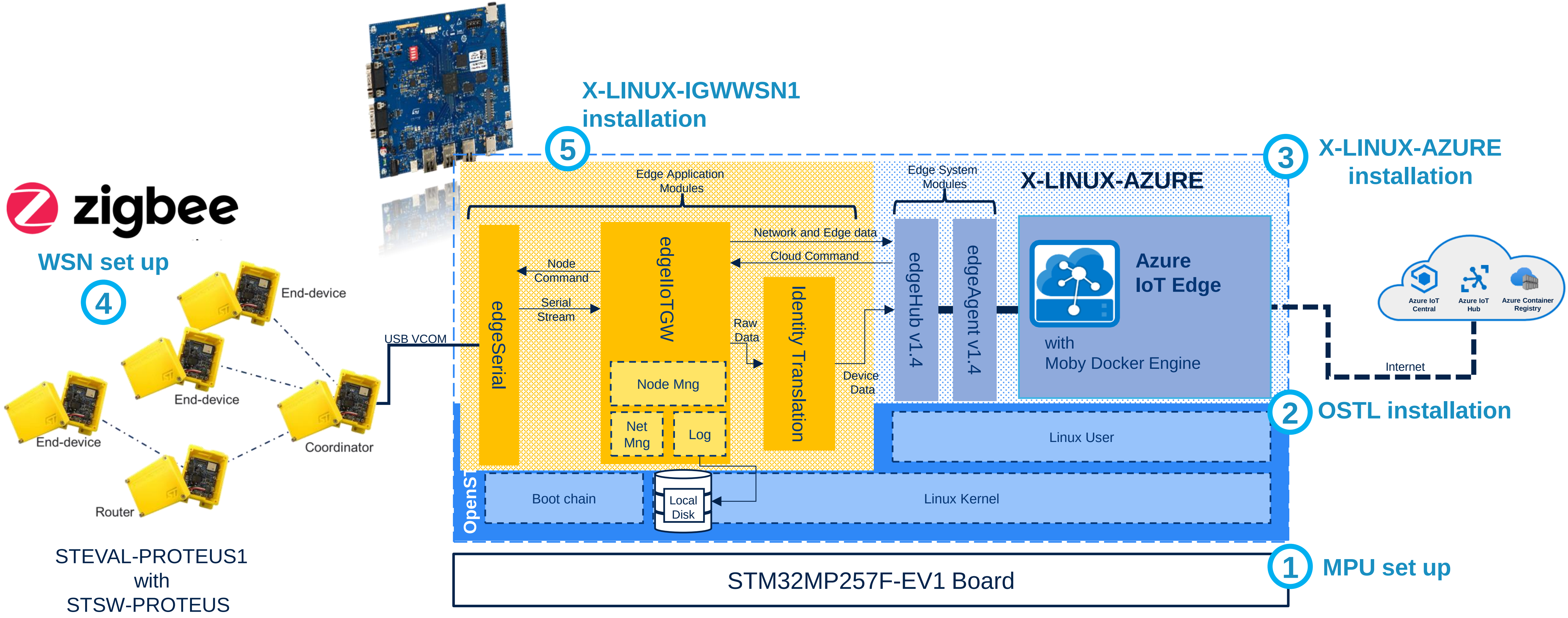
- A Linux® PC running Ubuntu® 20.04 or 22.04 is recommended
- Internet connection by ethernet cable
- **Microsoft Azure IoT Central account**

1 Edge GW Set up



Edge Software architecture

X-LINUX-IGWWSN1



X-LINUX-IGWWSN1 installation 5

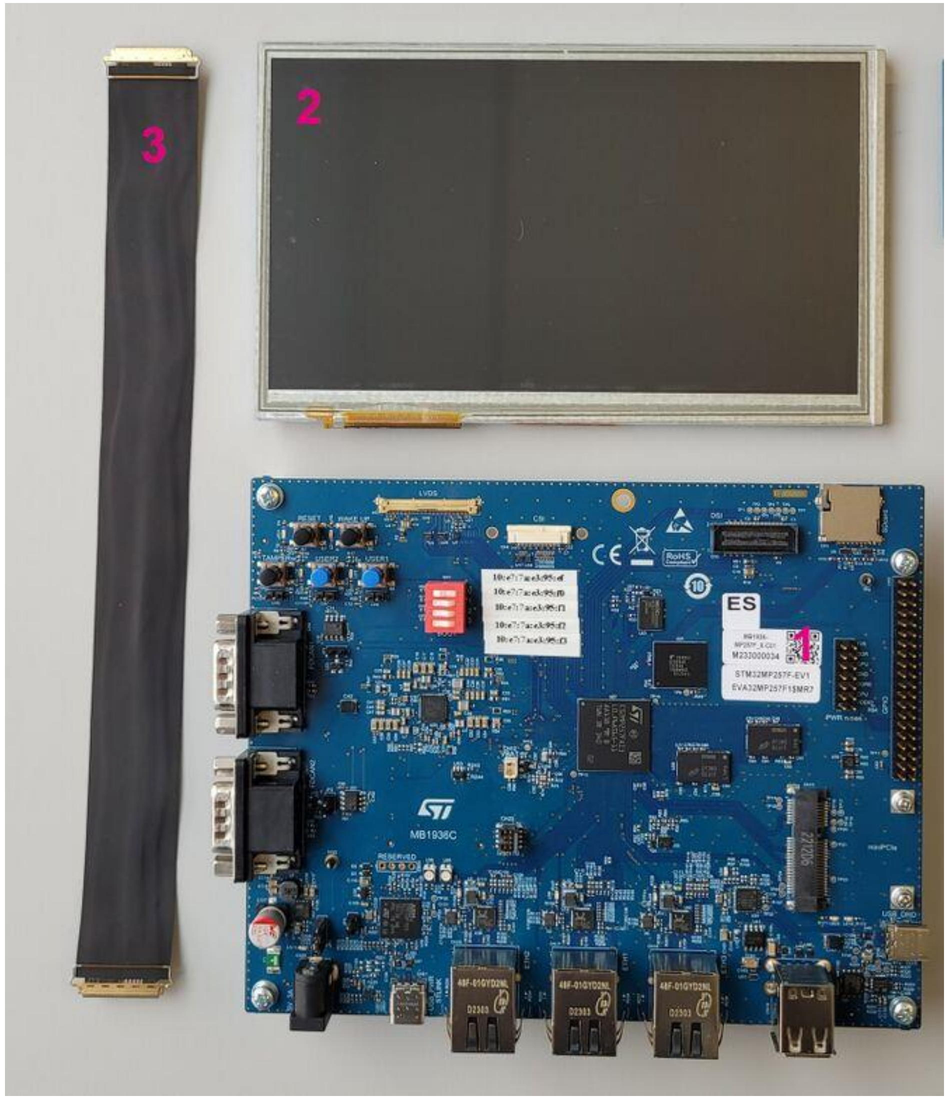
X-LINUX-AZURE installation 3

OSTL installation 2

MPU set up 1

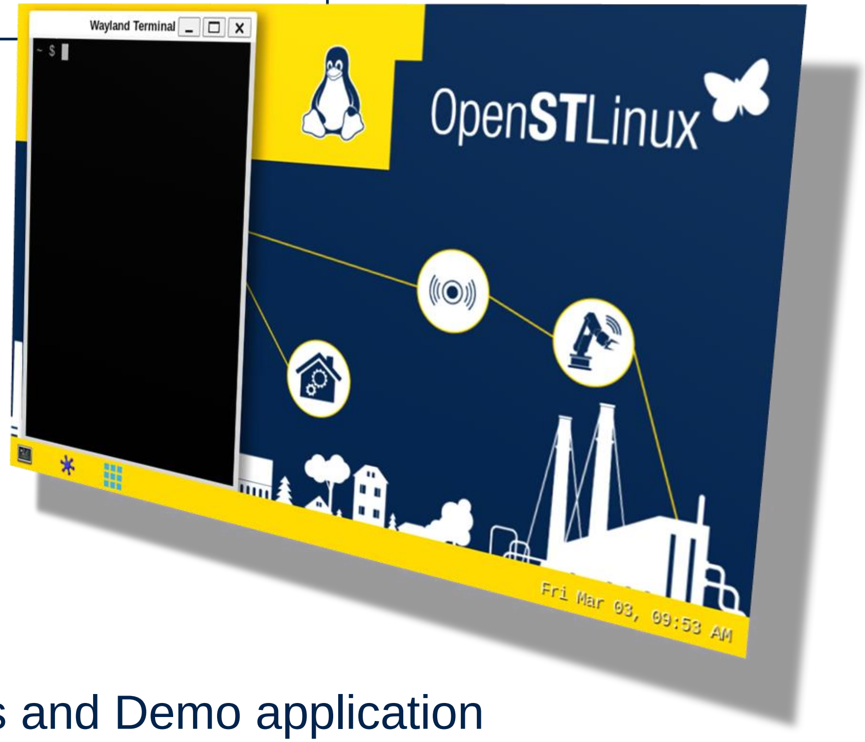
WSN set up 4

STEVAL-PROTEUS1
with
STSW-PROTEUS



STM32MP257F-EV1 kit overview

N	Description
1	MB1936 main board
2	7” LVDS WSVGA display with touch panel (EDT ETML0700Z9NDHA panel)
3	LVDS display cable

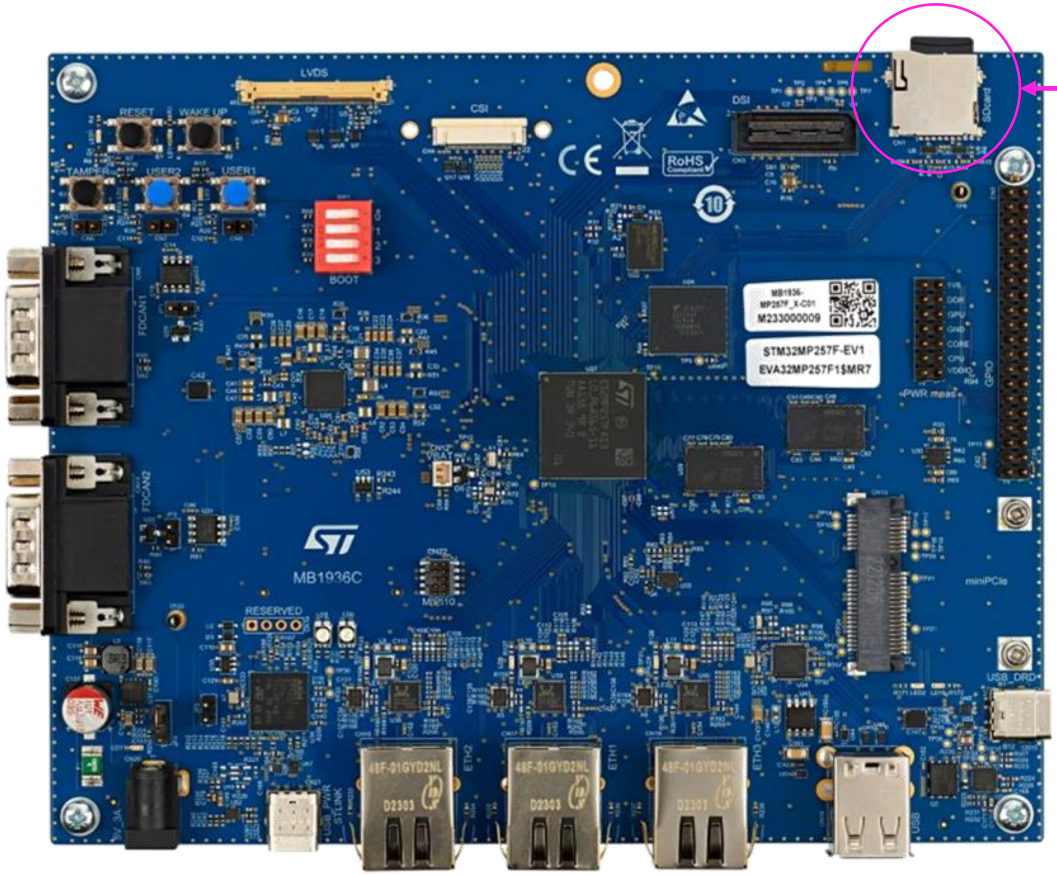


The display is useful to show debug messages and Demo application

For more information refer to: https://wiki.st.com/stm32mpu/wiki/STM32MP25_Evaluation_boards_-_Starter_Package

Preliminary

A micro SD, at least 16GB, is necessary to install the Open ST Linux



Insert the micro-SD in the slot

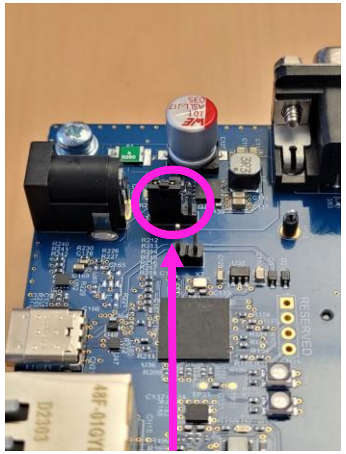
micro SD

⚠ Put the micro-SD in the slot before supplying the board

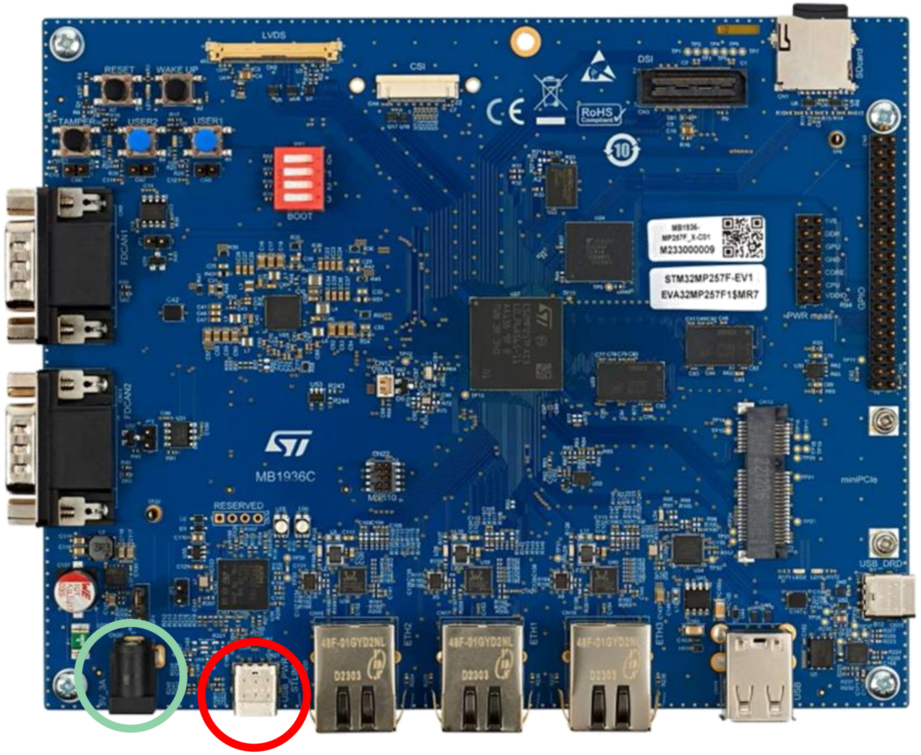
Power supply

The board can be powered in two different modes (selected with the dedicated jumper):

1) 5V@3A Power Supply Jack
2) 5V@3A standard USB Type-C source with profile



Power mode jumper

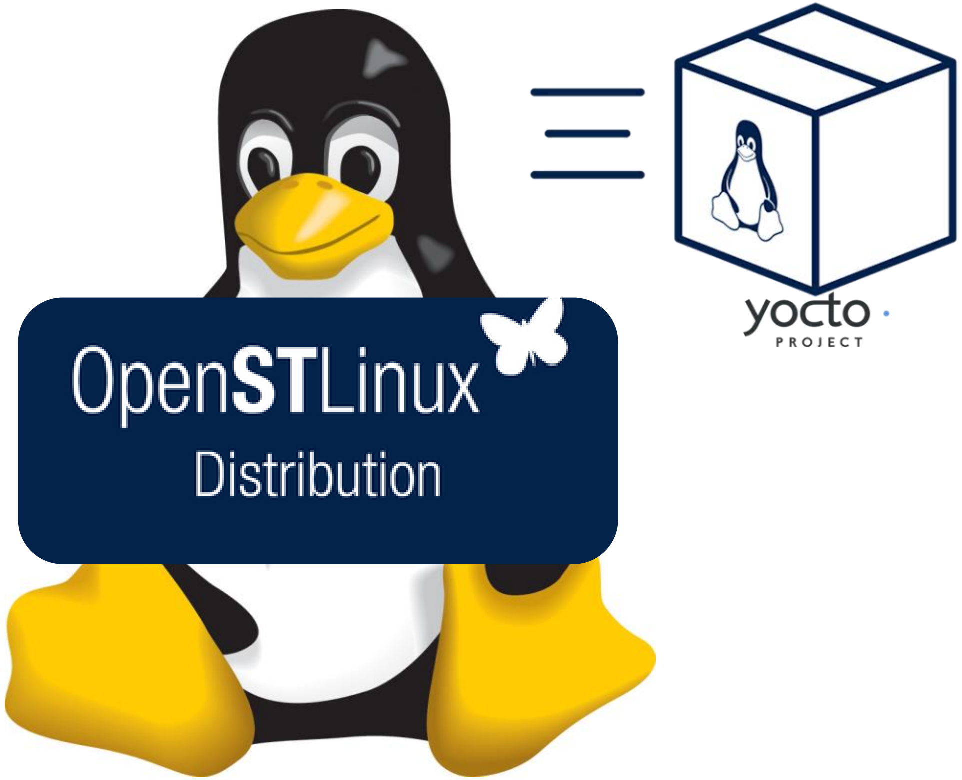


5V/3A Power Supply Jack

USB Type-C Adapter

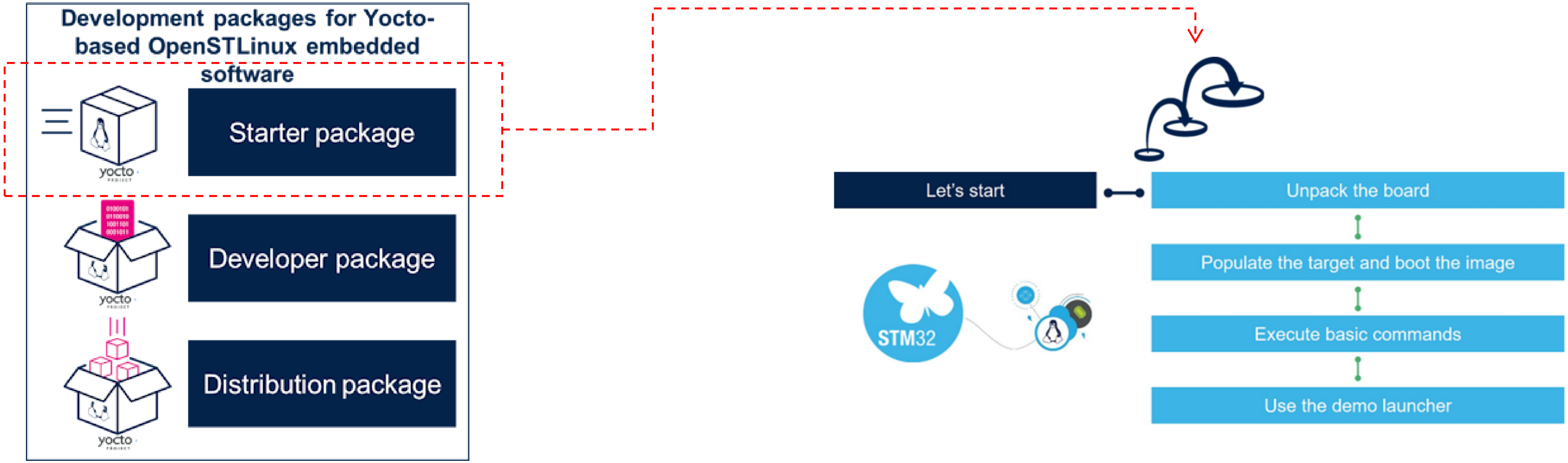
Important: in case to connect the USB Type-C port directly to the PC, ensure to have enough power.
For more information refer to the official documentation of the STM32MP257F-EV1 board.

2 OS installation



OSTL Installation

- Refer to the official ST guide Starter Package to install Open ST Linux (OSTL)



Development packages for Yocto-based OpenSTLinux embedded software

Starter package

Developer package

Distribution package

Let's start

Unpack the board

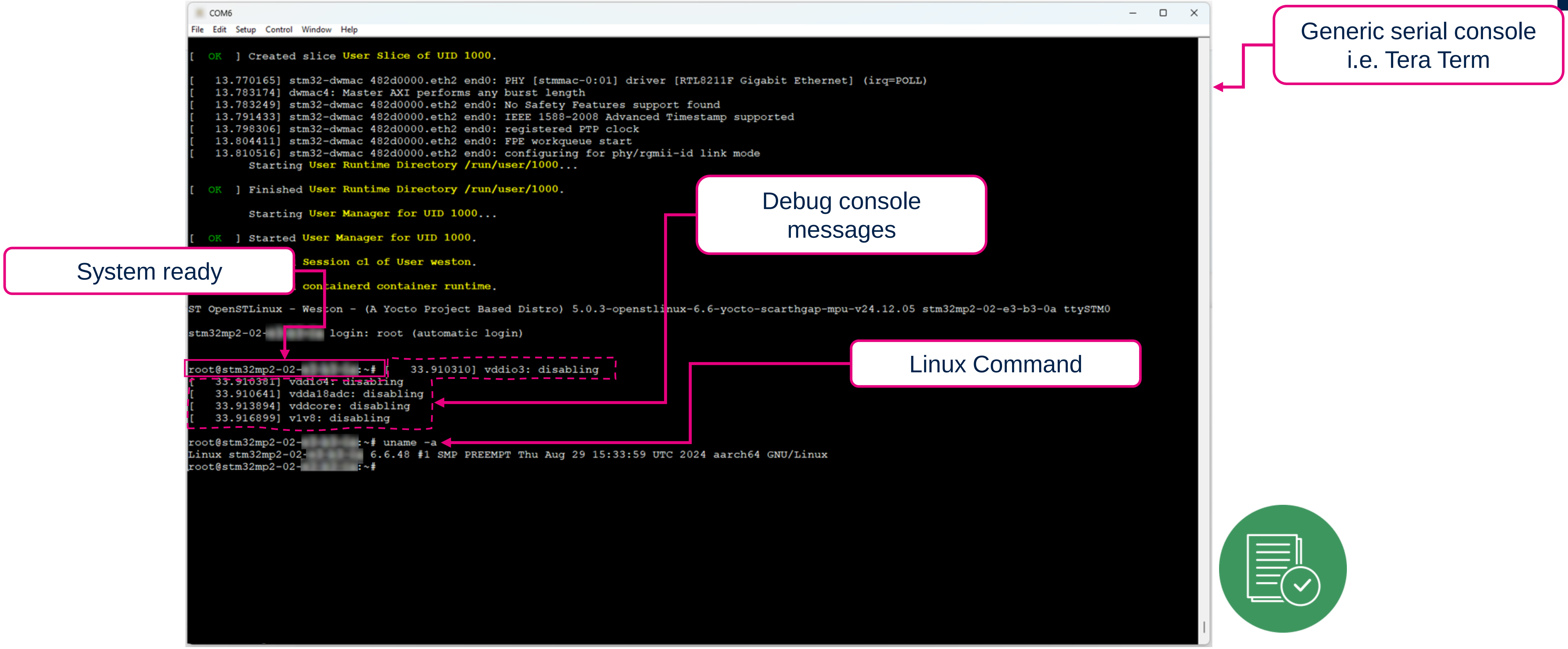
Populate the target and boot the image

Execute basic commands

Use the demo launcher

wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP2_boards/STM32MP257x-EV1

OSTL boot



```
[  OK  ] Created slice User Slice of UID 1000.

[   13.770165] stm32-dwmac 482d0000.eth2 end0: PHY [stmmac-0:01] driver [RTL8211F Gigabit Ethernet] (irq=POLL)
[   13.783174] dwmac4: Master AXI performs any burst length
[   13.783249] stm32-dwmac 482d0000.eth2 end0: No Safety Features support found
[   13.791433] stm32-dwmac 482d0000.eth2 end0: IEEE 1588-2008 Advanced Timestamp supported
[   13.798306] stm32-dwmac 482d0000.eth2 end0: registered PTP clock
[   13.804411] stm32-dwmac 482d0000.eth2 end0: FPE workqueue start
[   13.810516] stm32-dwmac 482d0000.eth2 end0: configuring for phy/rgmii-id link mode
         Starting User Runtime Directory /run/user/1000...

[  OK  ] Finished User Runtime Directory /run/user/1000.

         Starting User Manager for UID 1000...

[  OK  ] Started User Manager for UID 1000.

                 Session c1 of User weston.

                 containerd container runtime.

ST OpenSTLinux - Weston - (A Yocto Project Based Distro) 5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.12.05 stm32mp2-02-e3-b3-0a ttySTM0

stm32mp2-02-[redacted] login: root (automatic login)

root@stm32mp2-02-[redacted]:~# [   33.910310] vddio3: disabling
[   33.910381] vddio4: disabling
[   33.910641] vdda18adc: disabling
[   33.913894] vddcore: disabling
[   33.916899] v1v8: disabling

root@stm32mp2-02-[redacted]:~# uname -a
Linux stm32mp2-02-[redacted] 6.6.48 #1 SMP PREEMPT Thu Aug 29 15:33:59 UTC 2024 aarch64 GNU/Linux
root@stm32mp2-02-[redacted]:~#
```

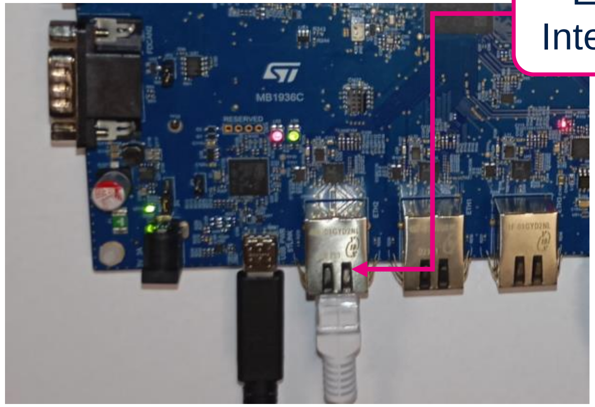
Generic serial console
i.e. Tera Term

Debug console
messages

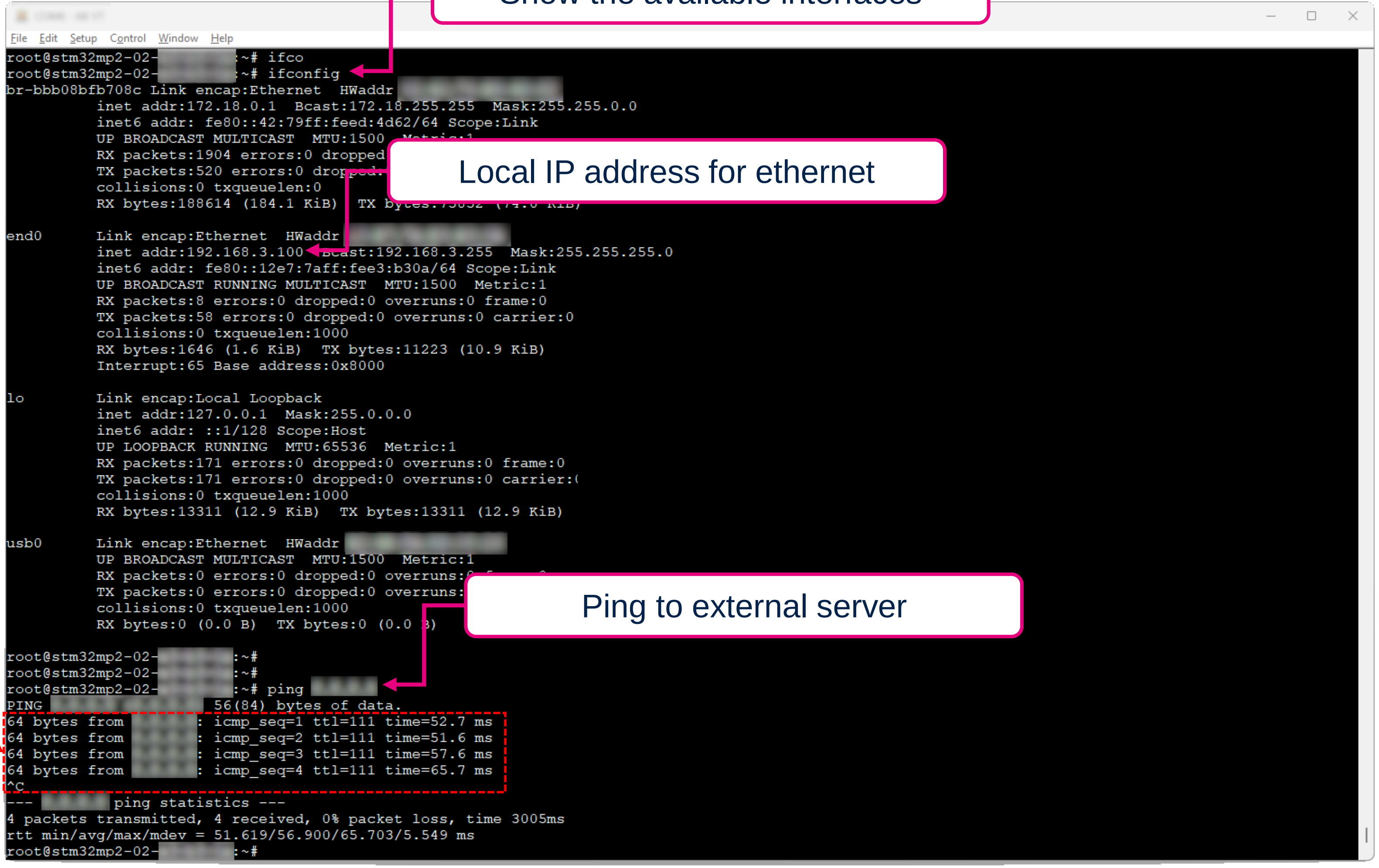
System ready

Linux Command

Internet connectivity



Ethernet port to Internet connection



Show the available interfaces

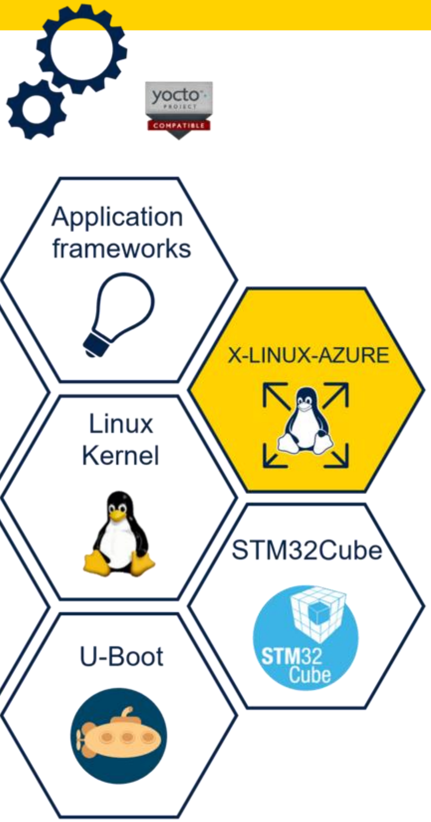
Local IP address for ethernet

Ping to external server

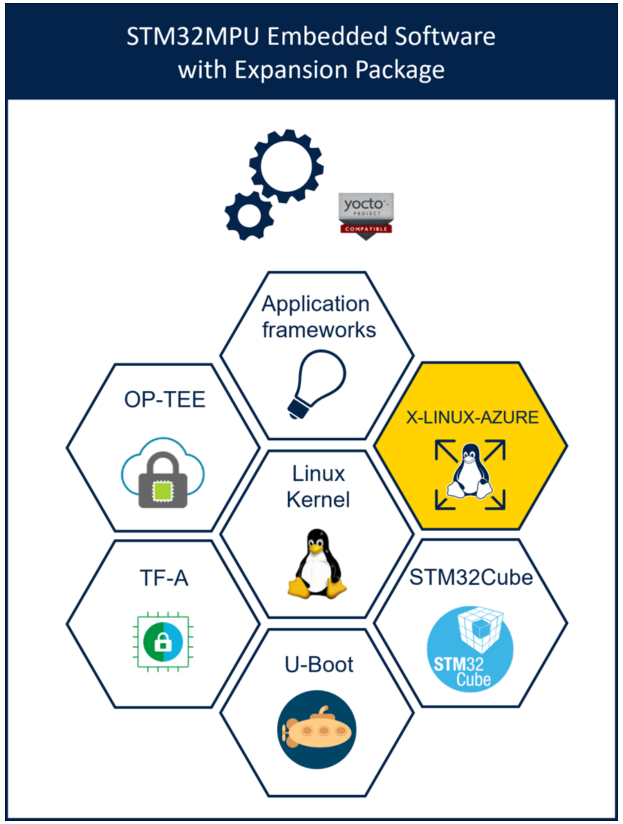
Positive feedback

```
root@stm32mp2-02-[redacted]:~# ifco
root@stm32mp2-02-[redacted]:~# ifconfig
br-bbb08bfb708c Link encap:Ethernet  HWaddr [redacted]
          inet addr:172.18.0.1  Bcast:172.18.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:79ff:feed:4d62/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500
          RX packets:1904 errors:0 dropped
          TX packets:520 errors:0 dropped
          collisions:0 txqueuelen:0
          RX bytes:188614 (184.1 KiB)  TX

end0      Link encap:Ethernet  HWaddr [redacted]
          inet addr:192.168.3.100  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::12e7:7aff:fee3:b30a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1646 (1.6 KiB)  TX bytes:11223 (10.9 KiB)
          Interrupt:65 Base address:0x8000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:171 errors:0 dropped:0 overruns:0 frame:0
          TX packets:171 errors:0 dropped:0 overruns:0 carrier:(
          collisions:0 txqueuelen:1000
          RX bytes:13311 (12.9 KiB)  TX bytes:13311 (12.9 KiB)

usb0      Link encap:Ethernet  HWaddr [redacted]
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:
          TX packets:0 errors:0 dropped:0 overruns:
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@stm32mp2-02-[redacted]:~#
root@stm32mp2-02-[redacted]:~#
root@stm32mp2-02-[redacted]:~# ping [redacted]
PING [redacted] 56(84) bytes of data.
64 bytes from [redacted]: icmp_seq=1 ttl=111 time=52.7 ms
64 bytes from [redacted]: icmp_seq=2 ttl=111 time=51.6 ms
64 bytes from [redacted]: icmp_seq=3 ttl=111 time=57.6 ms
64 bytes from [redacted]: icmp_seq=4 ttl=111 time=65.7 ms
^C
--- [redacted] ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 51.619/56.900/65.703/5.549 ms
root@stm32mp2-02-[redacted]:~#
```

3 X-LINUX-AZURE installation



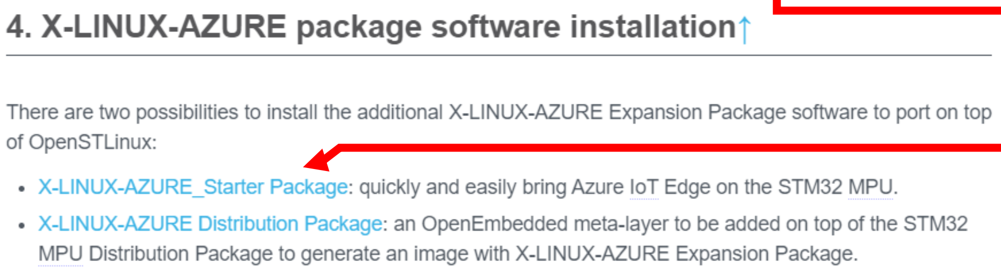
X-LINUX-AZURE installation



- Open the related page on st.com

  X-LINUX-AZURE expansion package - stm32mpu

- *Section n4 select the Starter Package link:*



4. X-LINUX-AZURE package software installation↑

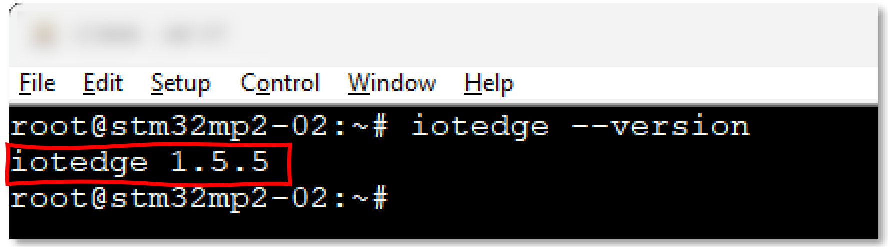
There are two possibilities to install the additional X-LINUX-AZURE Expansion Package software to port on top of OpenSTLinux:

- X-LINUX-AZURE_Starter Package: quickly and easily bring Azure IoT Edge on the STM32 MPU.
- X-LINUX-AZURE Distribution Package: an OpenEmbedded meta-layer to be added on top of the STM32 MPU Distribution Package to generate an image with X-LINUX-AZURE Expansion Package.

- *Section n5 package software setup*
  - Select the section 5.1 - Authentication with symmetric key stored in file system
  - Go to the section 5.4 – Troubleshooting
- *Section n6. How to run the Simulated Sensor Module*
  - 6.1. On Azure IoT Central

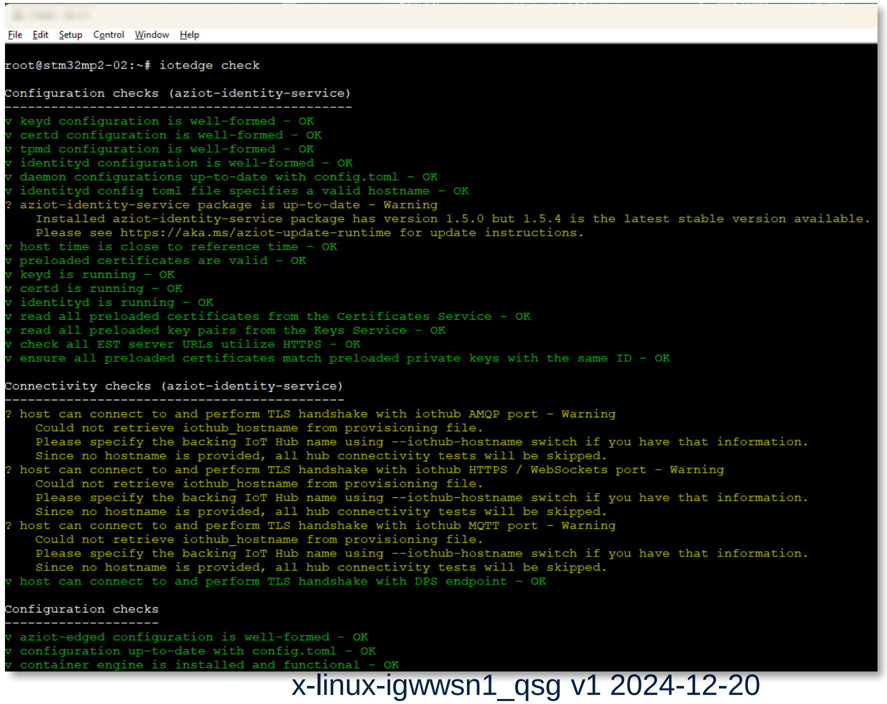
Microsoft *iotedge* check

- Check the installation: *iotedge --version*



```
root@stm32mp2-02:~# iotedge --version
iotedge 1.5.5
root@stm32mp2-02:~#
```

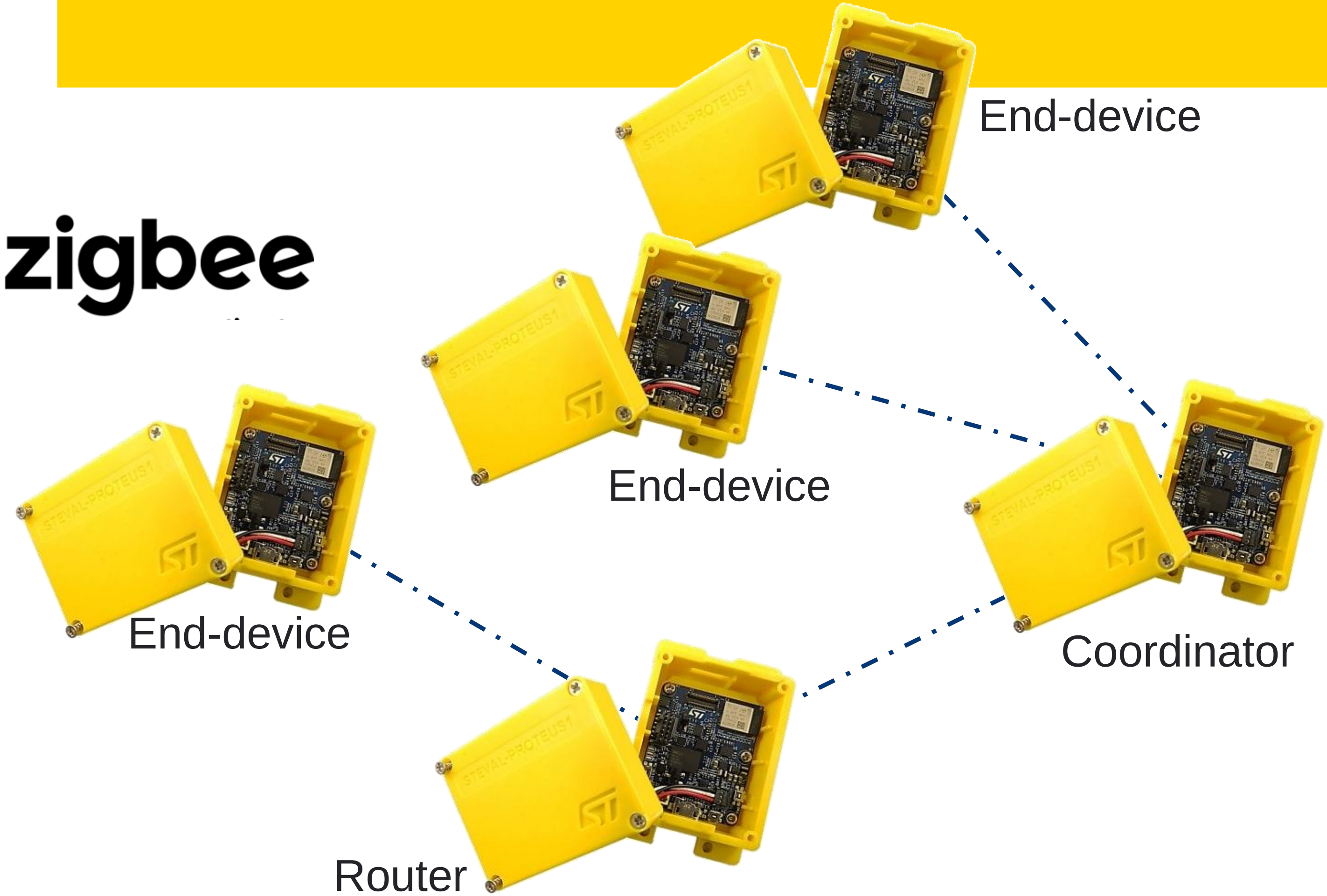
- Check the configuration: *iotedge check*



```
root@stm32mp2-02:~# iotedge check

Configuration checks (aziot-identity-service)
---------------------------------------------
v keyd configuration is well-formed - OK
v certd configuration is well-formed - OK
v tpmd configuration is well-formed - OK
v identityd configuration is well-formed - OK
v daemon configurations up-to-date with config.toml - OK
v identityd config toml file specifies a valid hostname - OK
? aziot-identity-service package is up-to-date - Warning
    Installed aziot-identity-service package has version 1.5.0 but 1.5.4 is the latest stable version available.
    Please see https://aka.ms/aziot-update-runtime for update instructions.
v host time is close to reference time - OK
v preloaded certificates are valid - OK
v keyd is running - OK
v certd is running - OK
v identityd is running - OK
v read all preloaded certificates from the Certificates Service - OK
v read all preloaded key pairs from the Keys Service - OK
v check all EST server URLs utilize HTTPS - OK
v ensure all preloaded certificates match preloaded private keys with the same ID - OK

Connectivity checks (aziot-identity-service)
--------------------------------------------
? host can connect to and perform TLS handshake with iothub AMQP port - Warning
    Could not retrieve iothub_hostname from provisioning file.
    Please specify the backing IoT Hub name using --iothub-hostname switch if you have that information.
    Since no hostname is provided, all hub connectivity tests will be skipped.
? host can connect to and perform TLS handshake with iothub HTTPS / WebSockets port - Warning
    Could not retrieve iothub_hostname from provisioning file.
    Please specify the backing IoT Hub name using --iothub-hostname switch if you have that information.
    Since no hostname is provided, all hub connectivity tests will be skipped.
? host can connect to and perform TLS handshake with iothub MQTT port - Warning
    Could not retrieve iothub_hostname from provisioning file.
    Please specify the backing IoT Hub name using --iothub-hostname switch if you have that information.
    Since no hostname is provided, all hub connectivity tests will be skipped.
v host can connect to and perform TLS handshake with DPS endpoint - OK

Configuration checks
--------------------
v aziot-edged configuration is well-formed - OK
v configuration up-to-date with config.toml - OK
v container engine is installed and functional - OK
```

life.augmented

4 ZigBee WSN setup



STEVAL-PROTEUS1 setup

The STEVAL-PROTEUS1 evaluation kit package includes:
1. a main board (dimensions: 29 mm x 35 mm)
2. a LiPo battery 3.7 V 480 mAh (HiMax)
3. a plastic case and some screws

STSW-PROTEUS v1.1.1 Software package

- ZigBee 3.0 stack (preliminary for all boards)
- Coordinator firmware
- Router firmware
- End node firmware

Links:

- STEVAL-PROTEUS1 web page
- UM3000 Getting started with STEVAL-PROTEUS1
- STSW-PROTEUS Software package
- UM3045 Getting started with STSW-PROTEUS1 (3.2 ZigBee-based application section)

Tools to update STEVAL-PROTEUS1

It is necessary to use:

- STM32CubeProgrammer software
- STLINK-V3MINIE hardware programmer

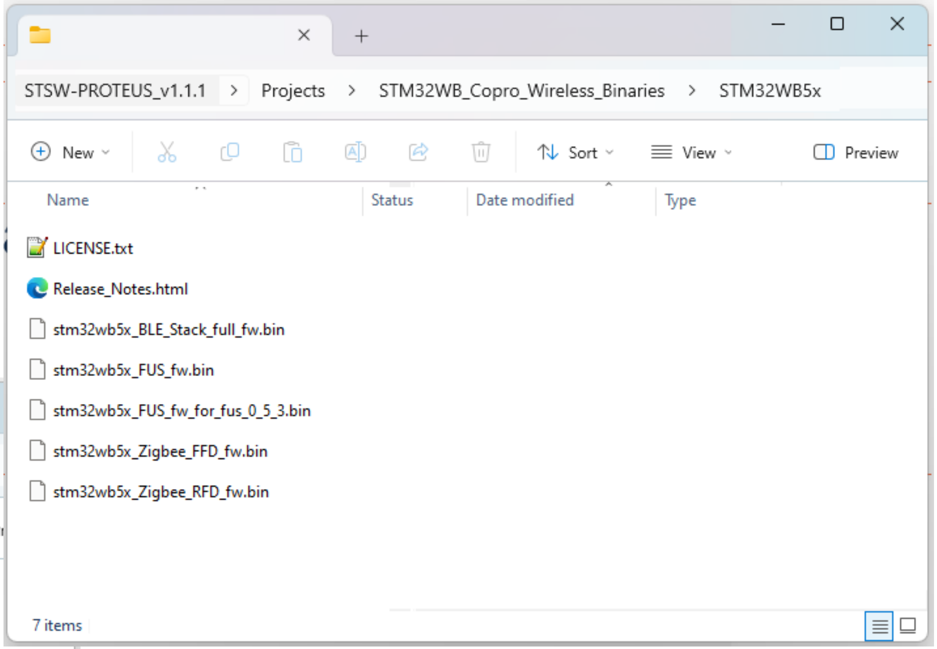
STM32CubeProgrammer

STLINK-V3MINIE

STSW-PROTEUS package

https://www.st.com/en/embedded-software/stsw-proteus.html

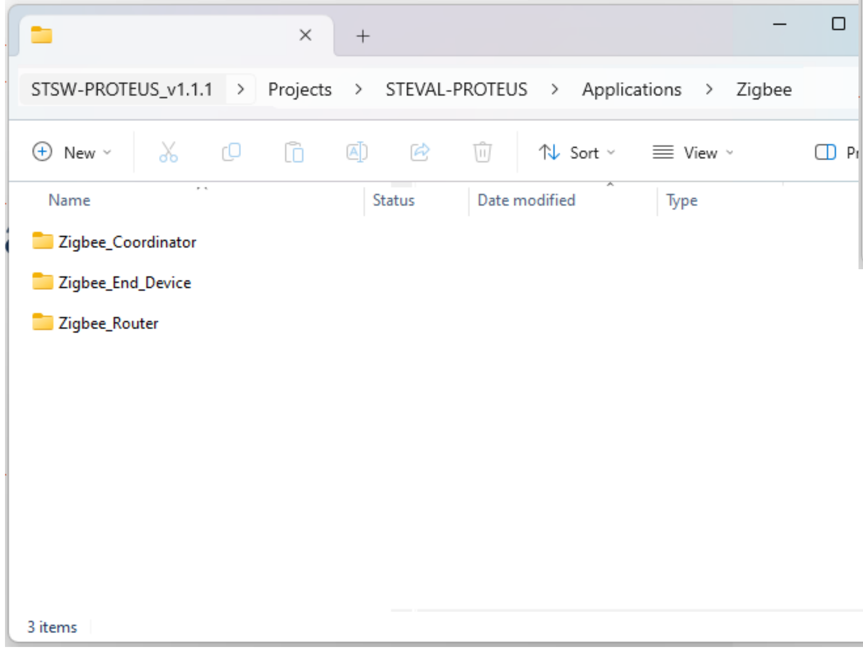
- Download and unzip → en.stsw-proteus.zip
- Stack binary firmware
- Application binary firmware



STSW-PROTEUS_v1.1.1 > Projects > STM32WB_Copro_Wireless_Binaries > STM32WB5x

- LICENSE.txt
- Release_Notes.html
- stm32wb5x_BLE_Stack_full_fw.bin
- stm32wb5x_FUS_fw.bin
- stm32wb5x_FUS_fw_for_fus_0_5_3.bin
- stm32wb5x_Zigbee_FFD_fw.bin
- stm32wb5x_Zigbee_RFD_fw.bin

7 items



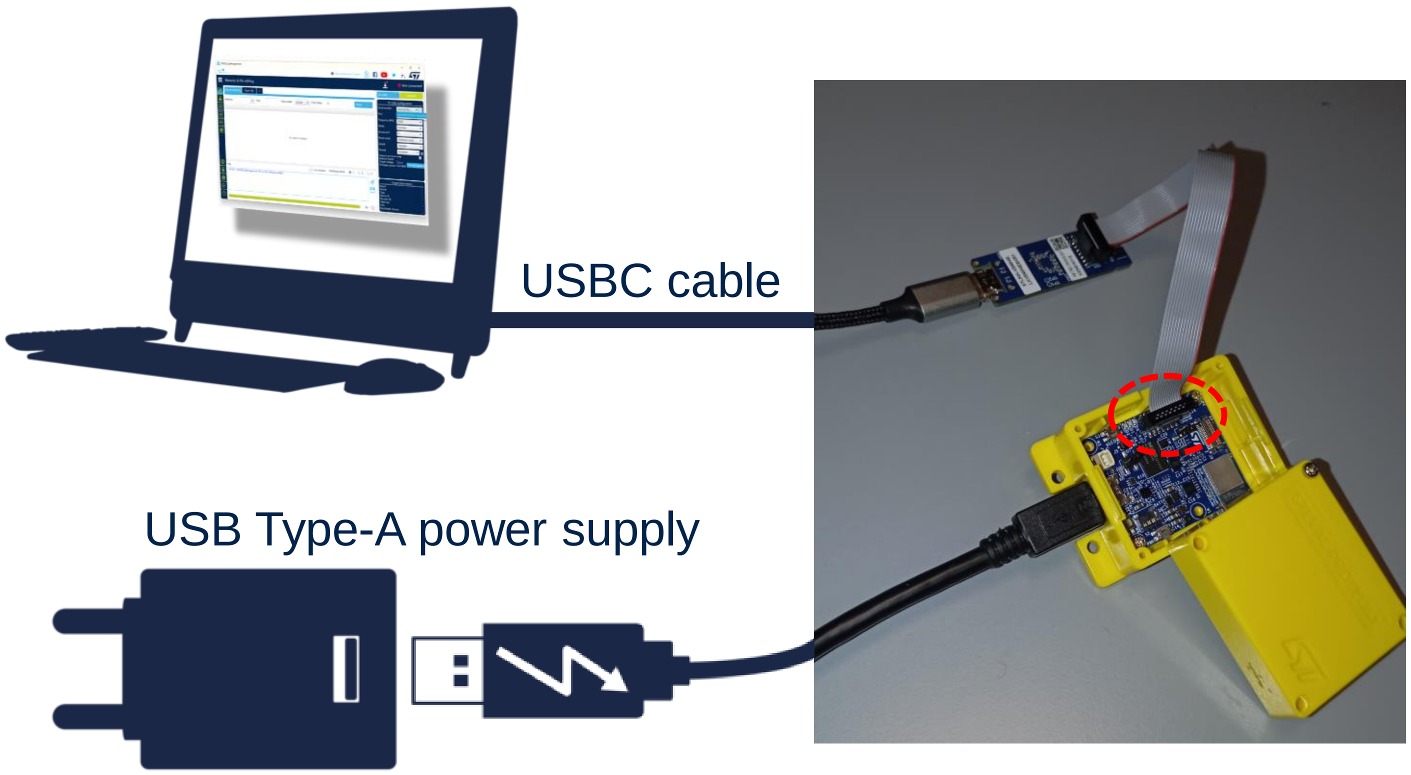
STSW-PROTEUS_v1.1.1 > Projects > STEVAL-PROTEUS > Applications > Zigbee

- Zigbee_Coordinator
- Zigbee_End_Device
- Zigbee_Router

3 items

life.augmented

Connect the STLINK-V3MINIE

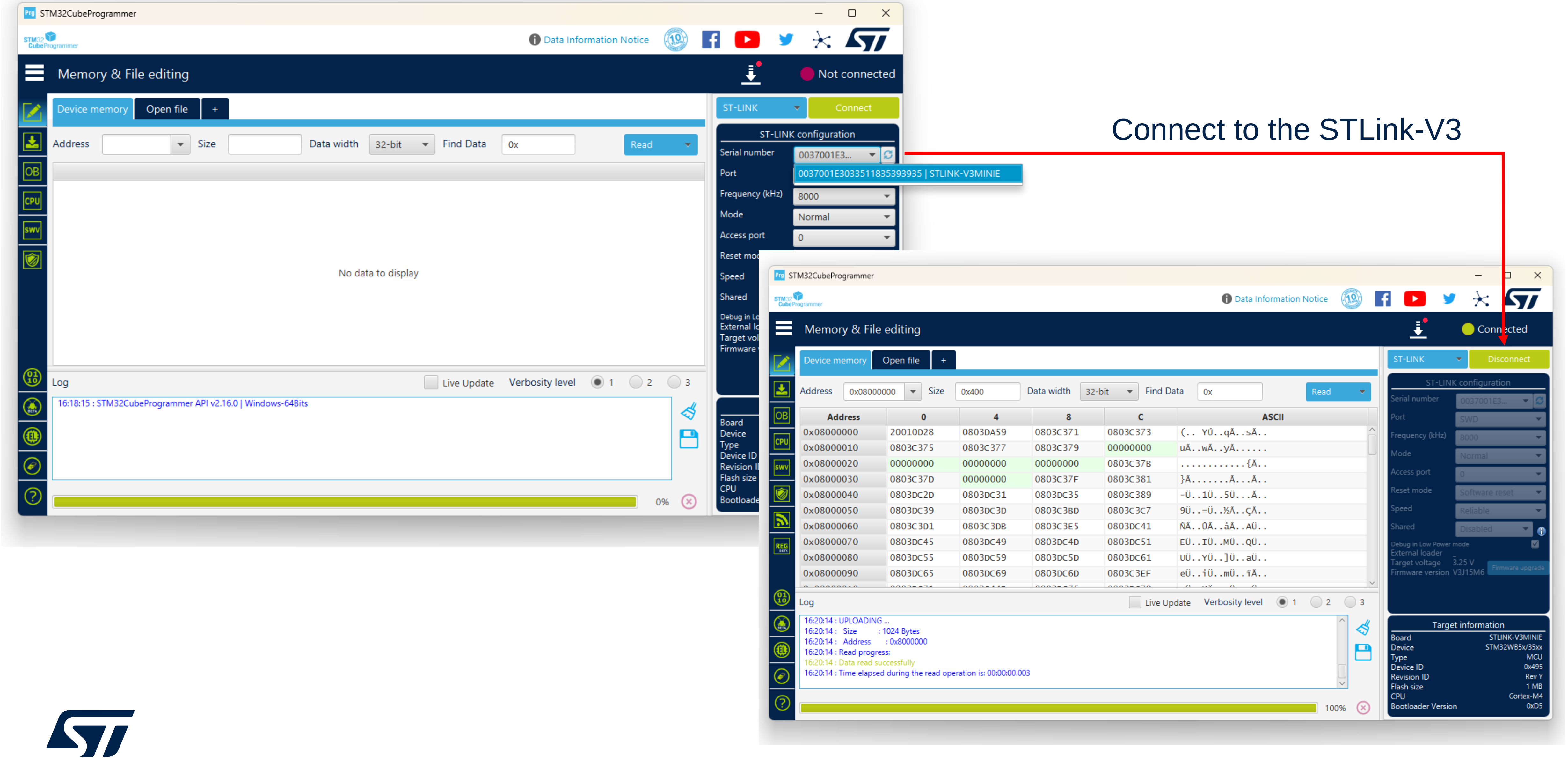


PC with STM32CubeProgrammer

USBC cable

USB Type-A power supply

Open STM32CubeProgrammer and select STLink-V3



Connect to the STLink-V3

STM32CubeProgrammer — Memory & File editing — Not connected

Address: | Size: | Data width: 32-bit | Find Data: 0x

ST-LINK configuration

- Serial number: 0037001E3...
- 0037001E3033511835393935 | STLINK-V3MINIE
- Port
- Frequency (kHz): 8000
- Mode: Normal
- Access port: 0
- Reset mode
- Speed
- Shared

No data to display

Log — □ Live Update — Verbosity level ● 1 ○ 2 ○ 3

```
16:18:15 : STM32CubeProgrammer API v2.16.0 | Windows-64Bits
```

0%

STM32CubeProgrammer — Memory & File editing — Connected

Address: 0x08000000 | Size: 0x400 | Data width: 32-bit | Find Data: 0x

```
Address      0          4          8          C          ASCII
0x08000000   20010D28   0803DA59   0803C371   0803C373   (.. YÚ..qÃ..sÃ..
0x08000010   0803C375   0803C377   0803C379   00000000   uÃ..wÃ..yÃ......
0x08000020   00000000   00000000   00000000   0803C37B   ............{Ã..
0x08000030   0803C37D   00000000   0803C37F   0803C381   }Ã.......Ã...Ã..
0x08000040   0803DC2D   0803DC31   0803DC35   0803C389   -Ü..1Ü..5Ü...Ã..
0x08000050   0803DC39   0803DC3D   0803C3BD   0803C3C7   9Ü..=Ü..½Ã..ÇÃ..
0x08000060   0803C3D1   0803C3DB   0803C3E5   0803DC41   ÑÃ..ÛÃ..åÃ..AÜ..
0x08000070   0803DC45   0803DC49   0803DC4D   0803DC51   EÜ..IÜ..MÜ..QÜ..
0x08000080   0803DC55   0803DC59   0803DC5D   0803DC61   UÜ..YÜ..]Ü..aÜ..
0x08000090   0803DC65   0803DC69   0803DC6D   0803C3EF   eÜ..iÜ..mÜ..ïÃ..
```

ST-LINK configuration

- Serial number: 0037001E3...
- Port: SWD
- Frequency (kHz): 8000
- Mode: Normal
- Access port: 0
- Reset mode: Software reset
- Speed: Reliable
- Shared: Disabled
- Debug in Low Power mode ☑
- External loader: _
- Target voltage: 3.25 V
- Firmware version: V3J15M6

Log — □ Live Update — Verbosity level ● 1 ○ 2 ○ 3

```
16:20:14 : UPLOADING ...
16:20:14 :   Size        : 1024 Bytes
16:20:14 :   Address     : 0x8000000
16:20:14 : Read progress:
16:20:14 : Data read successfully
16:20:14 : Time elapsed during the read operation is: 00:00:00.003
```

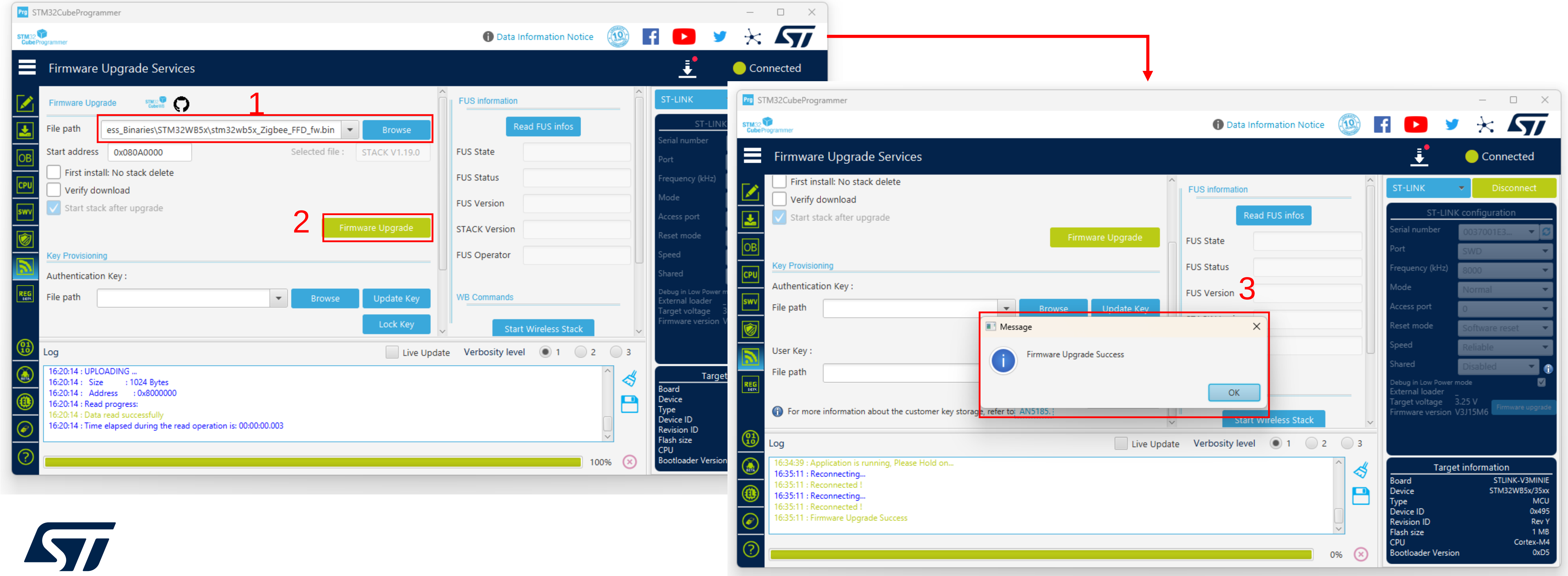
100%

Target information

Board	STLINK-V3MINIE
Device	STM32WB5x/35xx
Type	MCU
Device ID	0x495
Revision ID	Rev Y
Flash size	1 MB
CPU	Cortex-M4
Bootloader Version	0xD5

Update the ZigBee stack

- Folder → `STSW-PROTEUS_v1.1.1\Projects\STM32WB_Copro_Wireless_Binaries\STM32WB5x\`
- Coordinator and router → `stm32wb5x_Zigbee_FFD_fw.bin`
- End Device → `stm32wb5x_Zigbee_RFD_fw.bin`



File path: ess_Binaries\STM32WB5x\stm32wb5x_Zigbee_FFD_fw.bin

Start address: 0x080A0000

Selected file : STACK V1.19.0

☐ First install: No stack delete

☐ Verify download

☑ Start stack after upgrade

FUS information: FUS State, FUS Status, FUS Version, STACK Version, FUS Operator

Authentication Key :

File path

User Key :

Verbosity level ● 1 ○ 2 ○ 3

```
16:20:14 : UPLOADING ...
16:20:14 :  Size          : 1024 Bytes
16:20:14 :  Address       : 0x8000000
16:20:14 : Read progress:
16:20:14 : Data read successfully
16:20:14 : Time elapsed during the read operation is: 00:00:00.003
```

100%

Message

Firmware Upgrade Success

OK

For more information about the customer key storage, refer to AN5185.

```
16:34:39 : Application is running, Please Hold on...
16:35:11 : Reconnecting...
16:35:11 : Reconnected !
16:35:11 : Reconnecting...
16:35:11 : Reconnected !
16:35:11 : Firmware Upgrade Success
```

0%

ST-LINK configuration: Serial number 0037001E3..., Port SWD, Frequency (kHz) 8000, Mode Normal, Access port 0, Reset mode Software reset, Speed Reliable, Shared Disabled

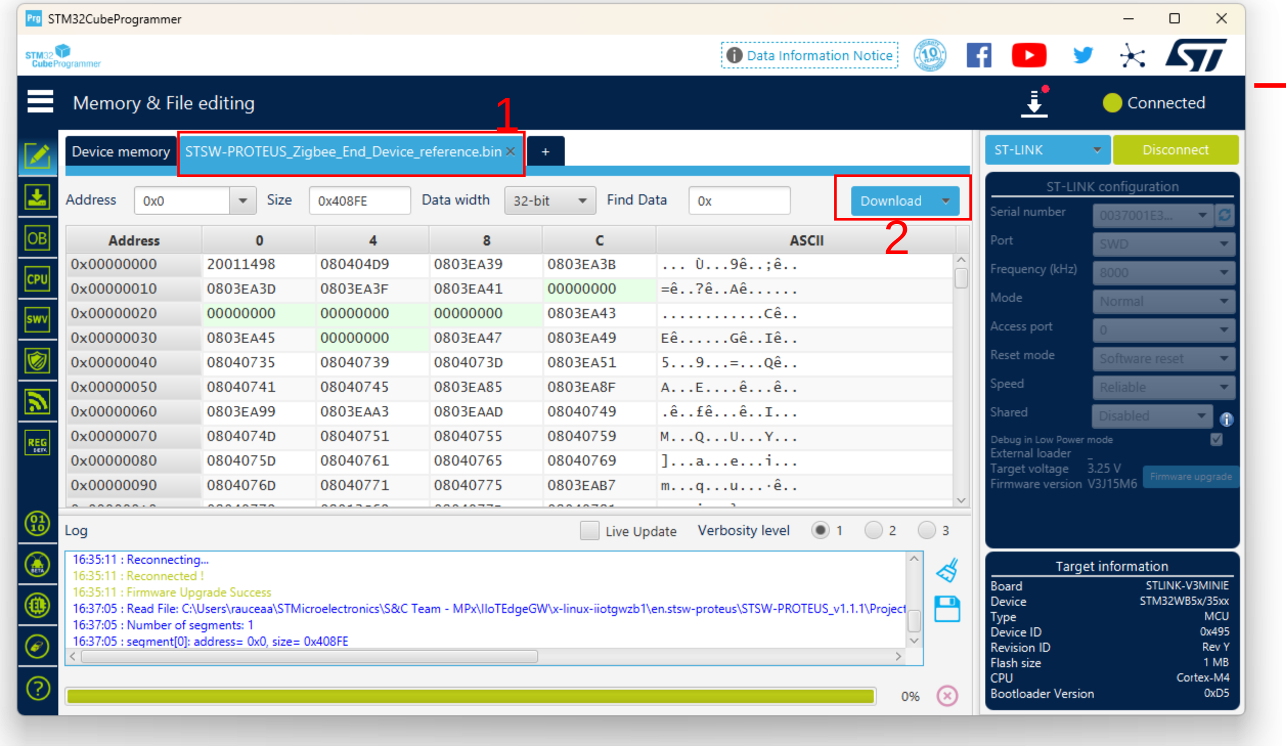
Target voltage 3.25 V

Firmware version V3J15M6

Target information	
Board	STLINK-V3MINIE
Device	STM32WB5x/35xx
Type	MCU
Device ID	0x495
Revision ID	Rev Y
Flash size	1 MB
CPU	Cortex-M4
Bootloader Version	0xD5

Update firmware application

- Folder → `STSW-PROTEUS_v1.1.1\Projects\STEVAL-PROTEUS\Applications\Zigbee\Zigbee_*\Binary`
- Coordinator → `STSW-PROTEUS_Zigbee_Coordinator_reference.bin`
- Router → `STSW-PROTEUS_Zigbee_Router_reference.bin`
- End-device → `STSW-PROTEUS_Zigbee_End_Device_reference.bin`



Address 0x0 · Size 0x408FE · Data width 32-bit · Find Data 0x

```
Address      0         4         8         C         ASCII
0x00000000   20011498  080404D9  0803EA39  0803EA3B  ... Ù...9ê..;ê..
0x00000010   0803EA3D  0803EA3F  0803EA41  00000000  =ê..?ê..Aê......
0x00000020   00000000  00000000  00000000  0803EA43  ............Cê..
0x00000030   0803EA45  00000000  0803EA47  0803EA49  Eê......Gê..Iê..
0x00000040   08040735  08040739  0804073D  0803EA51  5...9...=...Qê..
0x00000050   08040741  08040745  0803EA85  0803EA8F  A...E....ê...ê..
0x00000060   0803EA99  0803EAA3  0803EAAD  08040749  .ê..£ê...ê..I...
0x00000070   0804074D  08040751  08040755  08040759  M...Q...U...Y...
0x00000080   0804075D  08040761  08040765  08040769  ]...a...e...i...
0x00000090   0804076D  08040771  08040775  0803EAB7  m...q...u....ê..
```

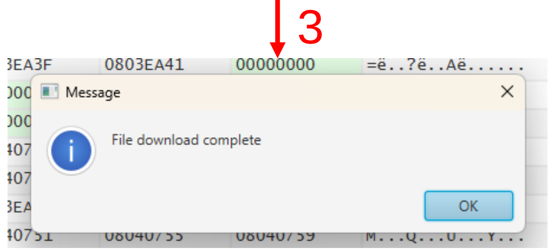
```
16:35:11 : Reconnecting...
16:35:11 : Reconnected !
16:35:11 : Firmware Upgrade Success
16:37:05 : Read File: C:\Users\rauceaa\STMicroelectronics\S&C Team - MPx\IIoTEdgeGW\x-linux-iiotgwzb1\en.stsw-proteus\STSW-PROTEUS_v1.1.1\Project
16:37:05 : Number of segments: 1
16:37:05 : segment[0]: address= 0x0, size= 0x408FE
```

ST-LINK configuration

Serial number	0037001E3...
Port	SWD
Frequency (kHz)	8000
Mode	Normal
Access port	0
Reset mode	Software reset
Speed	Reliable
Shared	Disabled
Debug in Low Power mode	☑
External loader	_
Target voltage	3.25 V
Firmware version	V3J15M6

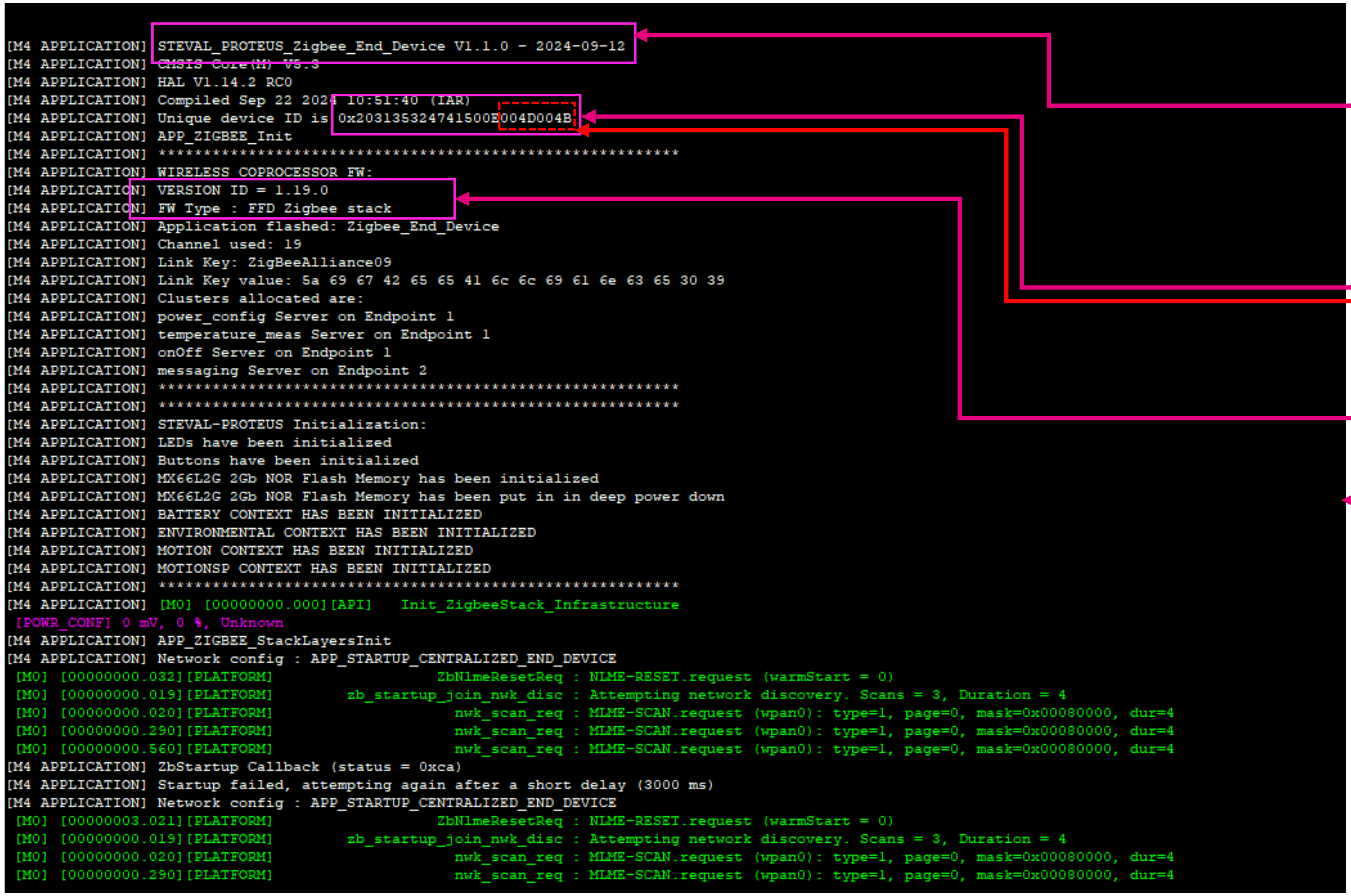
Target information

Board	STLINK-V3MINIE
Device	STM32WB5x/35xx
Type	MCU
Device ID	0x495
Revision ID	Rev Y
Flash size	1 MB
CPU	Cortex-M4
Bootloader Version	0xD5



File download complete

STLINK-V3 serial output example



```
[M4 APPLICATION] STEVAL_PROTEUS_Zigbee_End_Device V1.1.0 - 2024-09-12
[M4 APPLICATION] CMSIS Core(M) V5.3
[M4 APPLICATION] HAL V1.14.2 RC0
[M4 APPLICATION] Compiled Sep 22 2024 10:51:40 (IAR)
[M4 APPLICATION] Unique device ID is 0x203135324741500E004D004B
[M4 APPLICATION] APP_ZIGBEE_Init
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] WIRELESS COPROCESSOR FW:
[M4 APPLICATION] VERSION ID = 1.19.0
[M4 APPLICATION] FW Type : FFD Zigbee stack
[M4 APPLICATION] Application flashed: Zigbee_End_Device
[M4 APPLICATION] Channel used: 19
[M4 APPLICATION] Link Key: ZigBeeAlliance09
[M4 APPLICATION] Link Key value: 5a 69 67 42 65 65 41 6c 6c 69 61 6e 63 65 30 39
[M4 APPLICATION] Clusters allocated are:
[M4 APPLICATION] power_config Server on Endpoint 1
[M4 APPLICATION] temperature_meas Server on Endpoint 1
[M4 APPLICATION] onOff Server on Endpoint 1
[M4 APPLICATION] messaging Server on Endpoint 2
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] STEVAL-PROTEUS Initialization:
[M4 APPLICATION] LEDs have been initialized
[M4 APPLICATION] Buttons have been initialized
[M4 APPLICATION] MX66L2G 2Gb NOR Flash Memory has been initialized
[M4 APPLICATION] MX66L2G 2Gb NOR Flash Memory has been put in in deep power down
[M4 APPLICATION] BATTERY CONTEXT HAS BEEN INITIALIZED
[M4 APPLICATION] ENVIRONMENTAL CONTEXT HAS BEEN INITIALIZED
[M4 APPLICATION] MOTION CONTEXT HAS BEEN INITIALIZED
[M4 APPLICATION] MOTIONSP CONTEXT HAS BEEN INITIALIZED
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] [M0] [00000000.000][API]   Init_ZigbeeStack_Infrastructure
 [POWR_CONF] 0 mV, 0 %, Unknown
[M4 APPLICATION] APP_ZIGBEE_StackLayersInit
[M4 APPLICATION] Network config : APP_STARTUP_CENTRALIZED_END_DEVICE
 [M0] [00000000.032][PLATFORM]                  ZbNlmeResetReq : NLME-RESET.request (warmStart = 0)
 [M0] [00000000.019][PLATFORM]         zb_startup_join_nwk_disc : Attempting network discovery. Scans = 3, Duration = 4
 [M0] [00000000.020][PLATFORM]                     nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00080000, dur=4
 [M0] [00000000.290][PLATFORM]                     nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00080000, dur=4
 [M0] [00000000.560][PLATFORM]                     nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00080000, dur=4
[M4 APPLICATION] ZbStartup Callback (status = 0xca)
[M4 APPLICATION] Startup failed, attempting again after a short delay (3000 ms)
[M4 APPLICATION] Network config : APP_STARTUP_CENTRALIZED_END_DEVICE
 [M0] [00000003.021][PLATFORM]                  ZbNlmeResetReq : NLME-RESET.request (warmStart = 0)
 [M0] [00000000.019][PLATFORM]         zb_startup_join_nwk_disc : Attempting network discovery. Scans = 3, Duration = 4
 [M0] [00000000.020][PLATFORM]                     nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00080000, dur=4
 [M0] [00000000.290][PLATFORM]                     nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00080000, dur=4
```

STSW-PROTEUS FW Information:
- Zigbee
- End-device v1.1.0
- date 2024-09-12

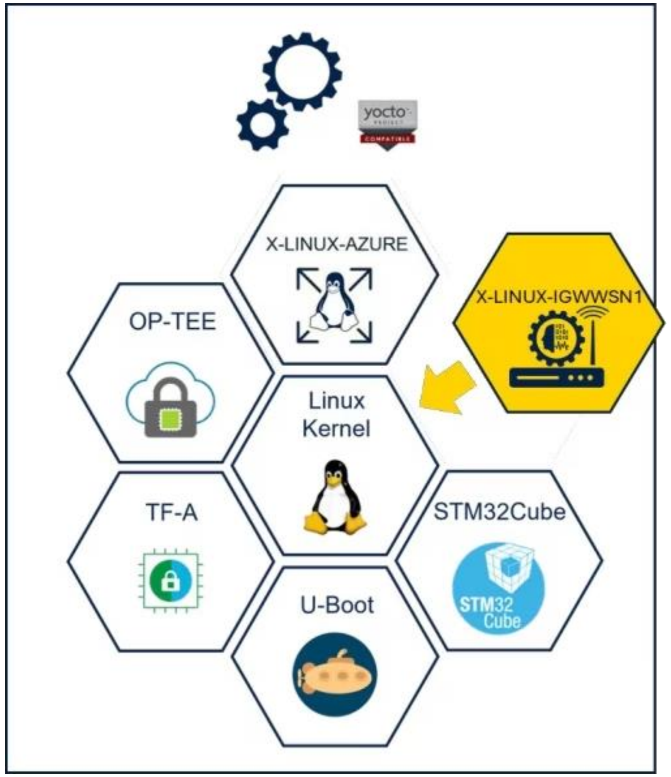
MCU Id -> Last 4 bytes are the End-node UID (available in the dashboard)

Zigbee stack information

Serial console connect to the STLink-V3 virtual com

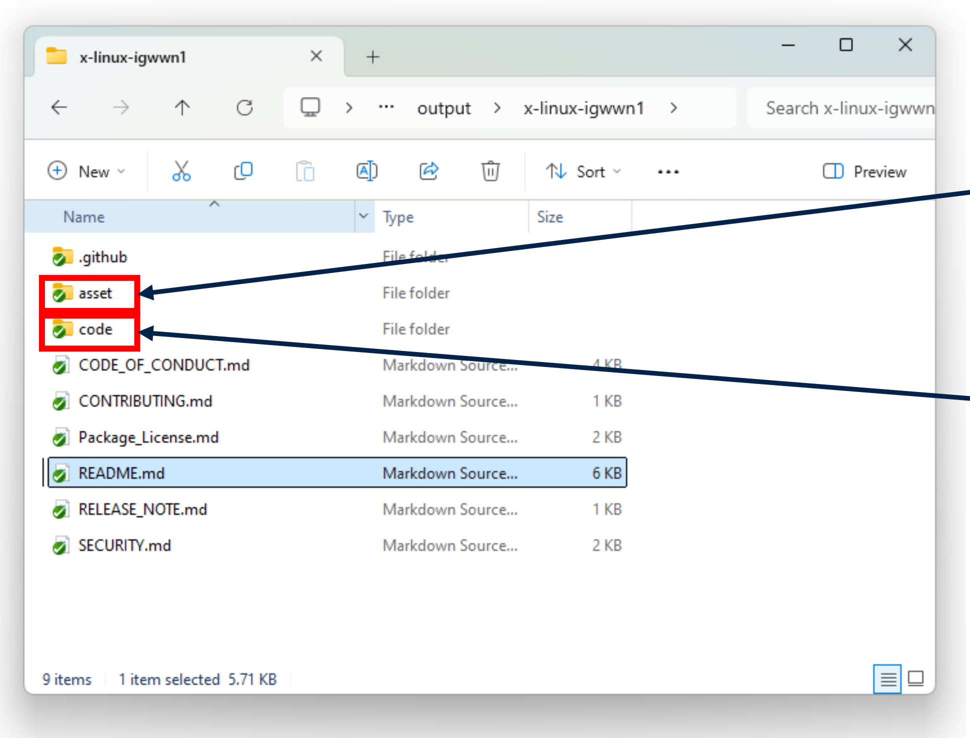
For more information refer to STEVAL-PROTEUS1 official documentation

5 X-LINUX-IGWWSN1



Download package

- Clone the source code from official STMicroelectronics Github
- PC> `git clone https://github.com/STMicroelectronics/x-linux-igwwsn1.git`



Name	Type	Size
.github	File folder	
asset	File folder	
code	File folder	
CODE_OF_CONDUCT.md	Markdown Source...	4 KB
CONTRIBUTING.md	Markdown Source...	1 KB
Package_License.md	Markdown Source...	2 KB
README.md	Markdown Source...	6 KB
RELEASE_NOTE.md	Markdown Source...	1 KB
SECURITY.md	Markdown Source...	2 KB

Configuration files

Source code: Module and Manifest (Visual Studio Code project)

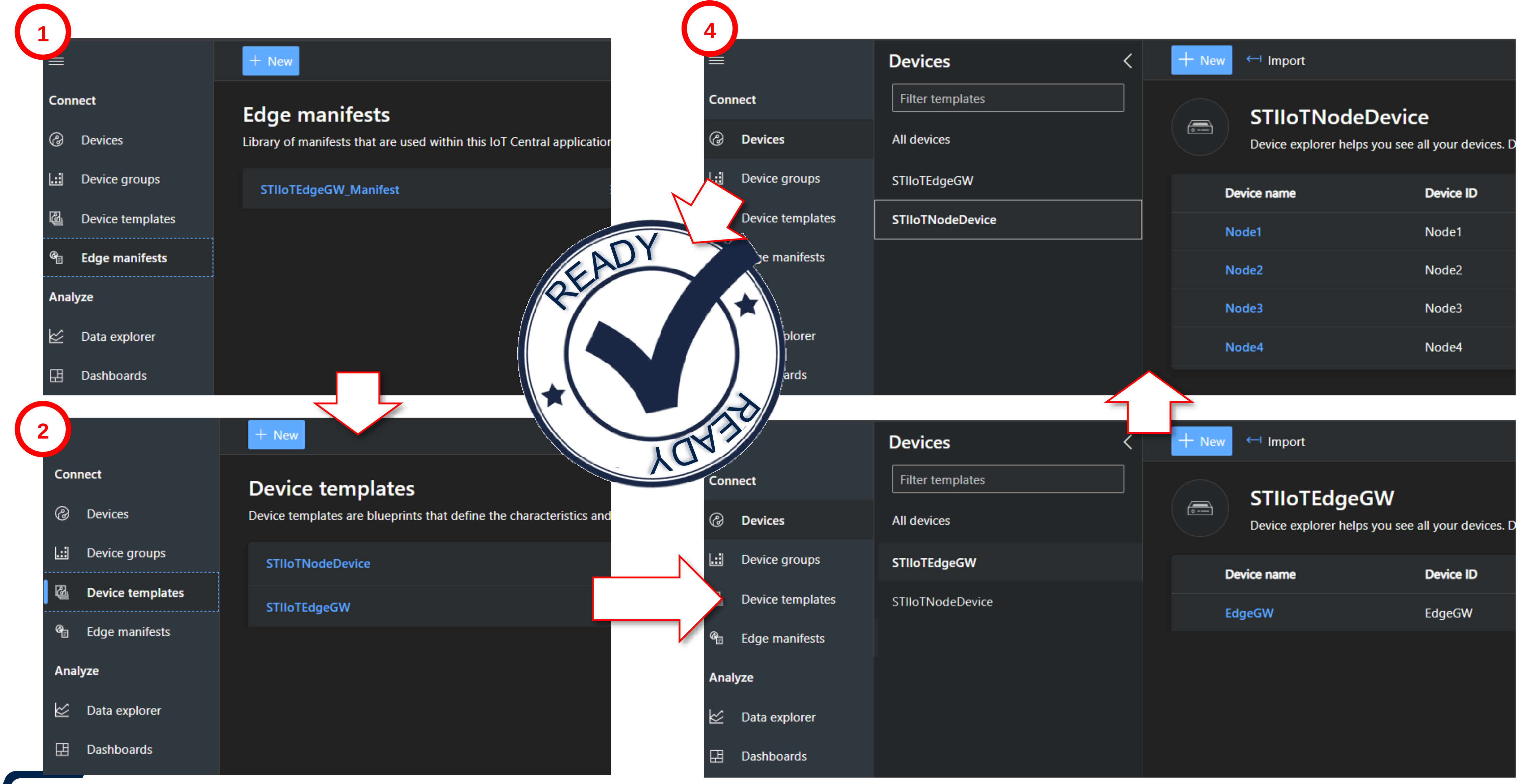
Azure IoT Central – Overview configuration

1. Add IoT Edge Manifest: STIIoTEdgeGW_Manifest
2. Create Device Templates
   a) STIIoTEdgeGW device template (modules: `edgeSerial` and `edgeIIoTGW`)
   b) STIIoTNodeDevice device template (model available)
3. Create IoT Edge Device (DT: STIIoTEdgeGW, MF: STIIoTEdgeGW_Manifest)
4. Create at least 4 Node Devices (DT device template STIIoTNodeDevice)
5. On the STM32MPU board
   a) Configure `/etc/aziot/config.toml`
   b) Configure `/var/lib/iiotedgegw/config/config_net.json`

Note: Microsoft IoT central account is mandatory

life.augmented

Azure IoT Central – Global overview



1

+ New

Connect
- Devices
- Device groups
- Device templates
- **Edge manifests**

Analyze
- Data explorer
- Dashboards

Edge manifests

Library of manifests that are used within this IoT Central applicatior

STIIoTEdgeGW_Manifest

2

+ New

Connect
- Devices
- Device groups
- **Device templates**
- Edge manifests

Analyze
- Data explorer
- Dashboards

Device templates

Device templates are blueprints that define the characteristics and

STIIoTNodeDevice

STIIoTEdgeGW

Connect
- **Devices**
- Device groups
- Device templates
- Edge manifests

Analyze
- Data explorer
- Dashboards

Devices

Filter templates

All devices

STIIoTEdgeGW

STIIoTNodeDevice

+ New ⟻ Import

STIIoTEdgeGW

Device explorer helps you see all your devices. D

Device name	Device ID
EdgeGW	EdgeGW

4

Connect
- **Devices**
- Device groups
- Device templates
- e manifests
- plorer
- ards

Devices

Filter templates

All devices

STIIoTEdgeGW

STIIoTNodeDevice

+ New ⟻ Import

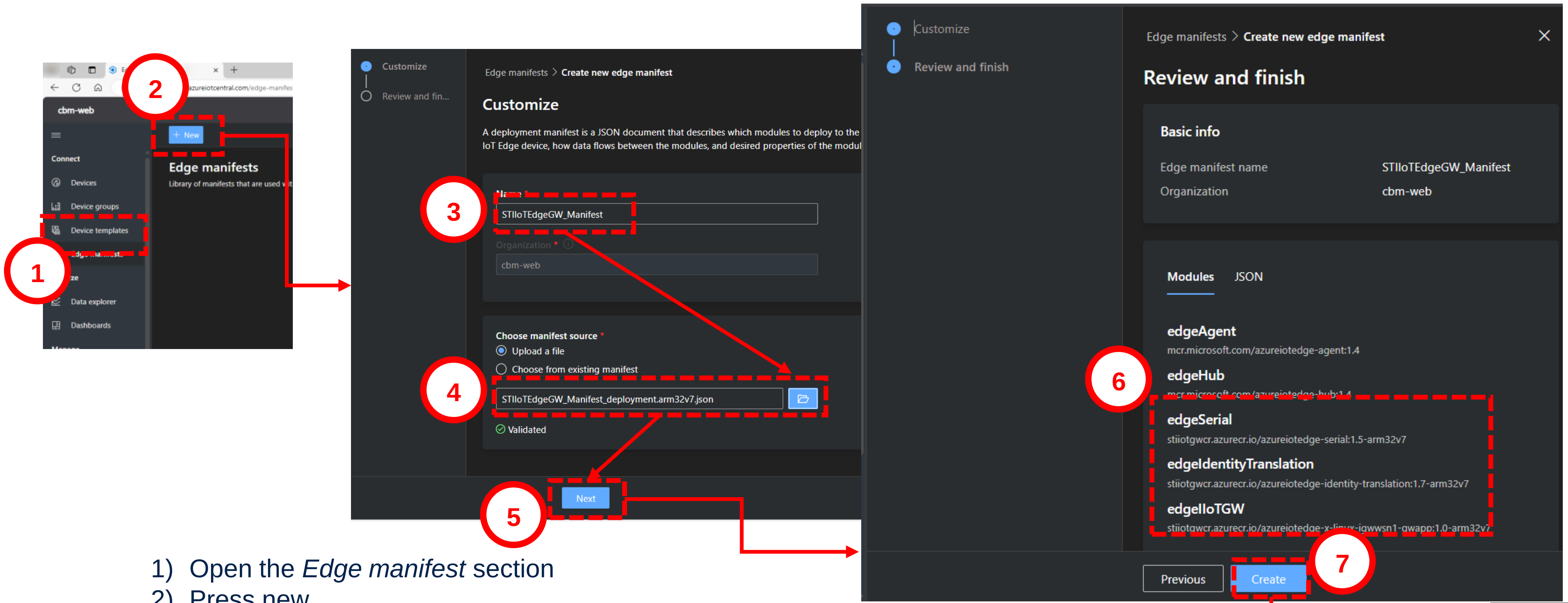
STIIoTNodeDevice

Device explorer helps you see all your devices. D

Device name	Device ID
Node1	Node1
Node2	Node2
Node3	Node3
Node4	Node4

Azure IoT Central – Add the manifest

- File: `asset/STIIoTEdgeGW_Manifest_deployment.arm32v7.json`



cbm-web

Connect

Devices

Device groups

Device templates

Edge manifests

Data explorer

Dashboards

+ New

Edge manifests

Library of manifests that are used wit

Customize

Review and fin...

Edge manifests > Create new edge manifest

Customize

A deployment manifest is a JSON document that describes which modules to deploy to the IoT Edge device, how data flows between the modules, and desired properties of the modul

Name *: STIIoTEdgeGW_Manifest

Organization *: cbm-web

Choose manifest source *

● Upload a file

○ Choose from existing manifest

STIIoTEdgeGW_Manifest_deployment.arm32v7.json

✓ Validated

Next

Customize

Review and finish

Edge manifests > Create new edge manifest

Review and finish

Basic info

Edge manifest name: STIIoTEdgeGW_Manifest

Organization: cbm-web

Modules | JSON

edgeAgent
mcr.microsoft.com/azureiotedge-agent:1.4

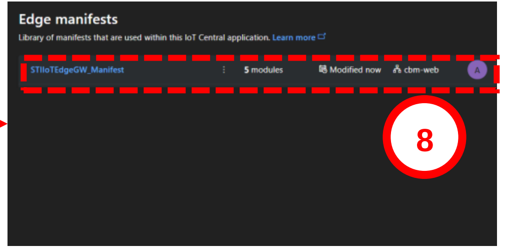
edgeHub
mcr.microsoft.com/azureiotedge-hub:1.4

edgeSerial
stiiotgwcr.azurecr.io/azureiotedge-serial:1.5-arm32v7

edgeIdentityTranslation
stiiotgwcr.azurecr.io/azureiotedge-identity-translation:1.7-arm32v7

edgeIIoTGW
stiiotgwcr.azurecr.io/azureiotedge-x-linux-igwwsn1-gwapp:1.0-arm32v7

Previous | Create



Edge manifests

Library of manifests that are used within this IoT Central application. Learn more

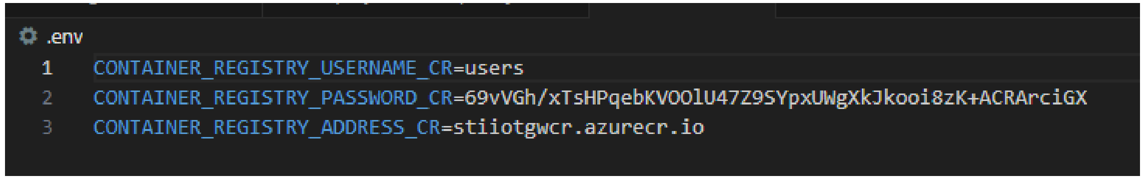
| STIIoTEdgeGW_Manifest | 5 modules | Modified now | cbm-web |

1) Open the *Edge manifest* section
2) Press new
3) Insert a manifest name (proposed *STIIoTEdgeGW_Manifest*)
4) Select the manifest file in the *asset* folder
5) Press Next
6) Check the Modules
7) Press Create
8) Find the new manifest

Azure IoT Central – Customization

- The Container Register credentials are to be set in the following files

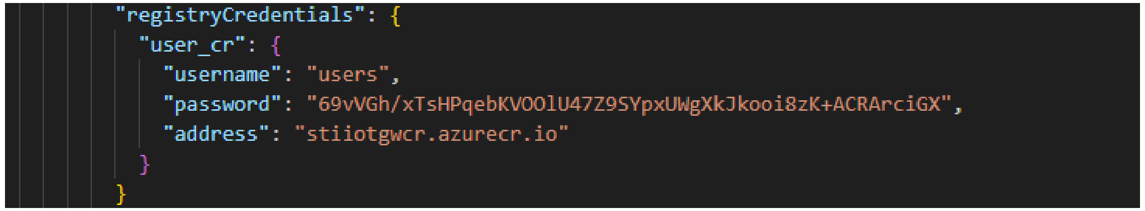
1. File: `code/.env`



.env

```
CONTAINER_REGISTRY_USERNAME_CR=users
CONTAINER_REGISTRY_PASSWORD_CR=69vVGh/xTsHPqebKVOOlU47Z9SYpxUWgXkJkooi8zK+ACRArciGX
CONTAINER_REGISTRY_ADDRESS_CR=stiiotgwcr.azurecr.io
```

2. File: `asset/STIIoTEdgeGW_Manifest_deployment.arm32v7.json`



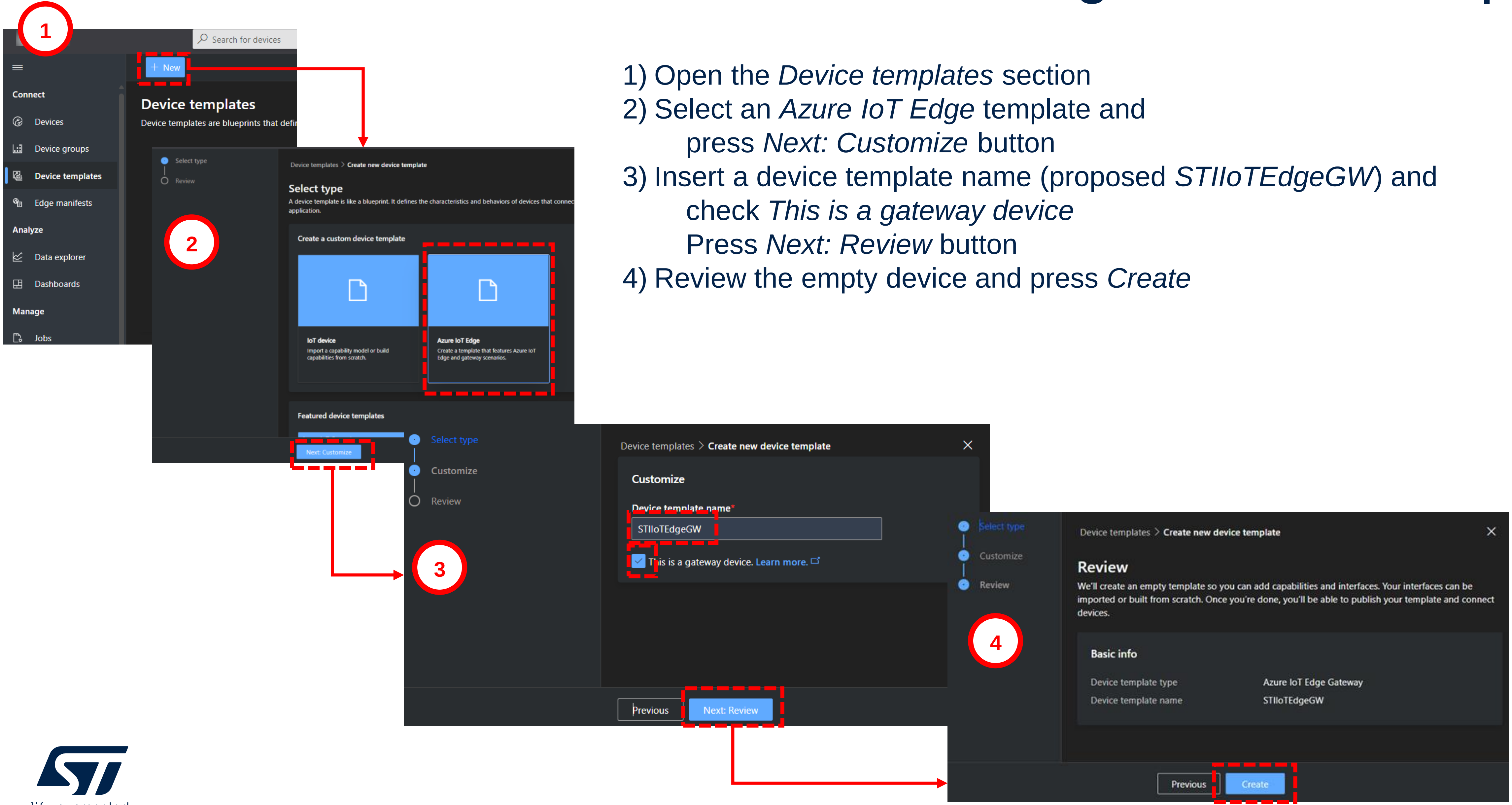
```
"registryCredentials": {
  "user_cr": {
    "username": "users",
    "password": "69vVGh/xTsHPqebKVOOlU47Z9SYpxUWgXkJkooi8zK+ACRArciGX",
    "address": "stiiotgwcr.azurecr.io"
  }
}
```

To pull the images set the following credentials:	
username	users
password	69vVGh/xTsHPqebKVOOlU47Z9SYpxUWgXkJkooi8zK+ACRArciGX
address	stiiotgwcr.azurecr.io

Note: in the repository the credentials could be empty, copy the information

Azure IoT Central – Create IoT Edge Device template

1) Open the *Device templates* section
2) Select an *Azure IoT Edge* template and
press *Next: Customize* button
3) Insert a device template name (proposed *STIIoTEdgeGW*) and
check *This is a gateway device*
Press *Next: Review* button
4) Review the empty device and press *Create*



Device templates

Device templates are blueprints that defi

Device templates > Create new device template

Select type

A device template is like a blueprint. It defines the characteristics and behaviors of devices that connec application.

Create a custom device template

IoT device
Import a capability model or build capabilities from scratch.

Azure IoT Edge
Create a template that features Azure IoT Edge and gateway scenarios.

Featured device templates

Device templates > Create new device template

Customize

Device template name*

STIIoTEdgeGW

☑ This is a gateway device. Learn more.

Device templates > Create new device template

Review

We'll create an empty template so you can add capabilities and interfaces. Your interfaces can be imported or built from scratch. Once you're done, you'll be able to publish your template and connect devices.

Basic info

Device template type	Azure IoT Edge Gateway
Device template name	STIIoTEdgeGW

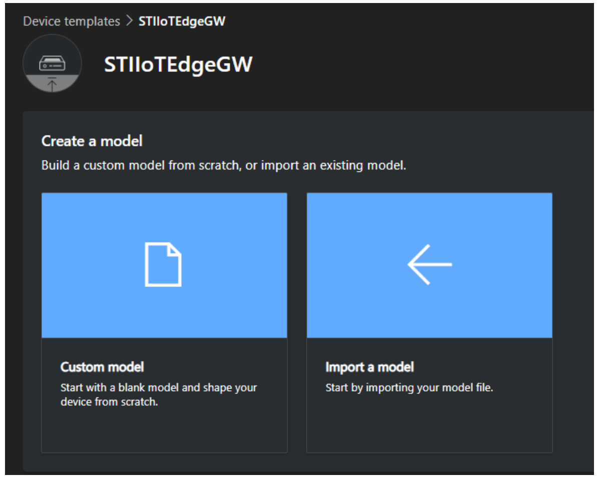
life.augmented

Azure IoT Central – Add the modules

Add the modules:

- **`edgeSerial`**
- **`edgeIIoTGW`**

with the corresponding file in asset folder.

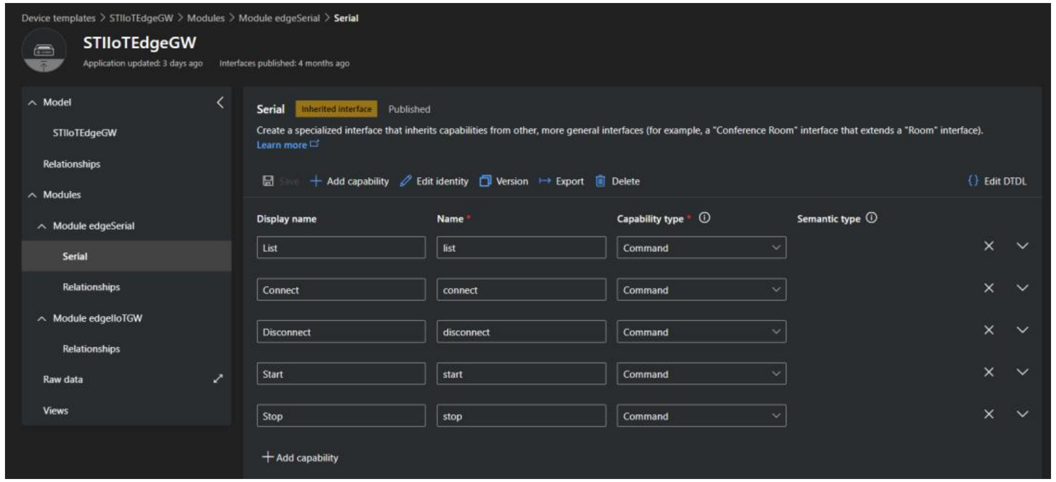


Create a model

Build a custom model from scratch, or import an existing model.

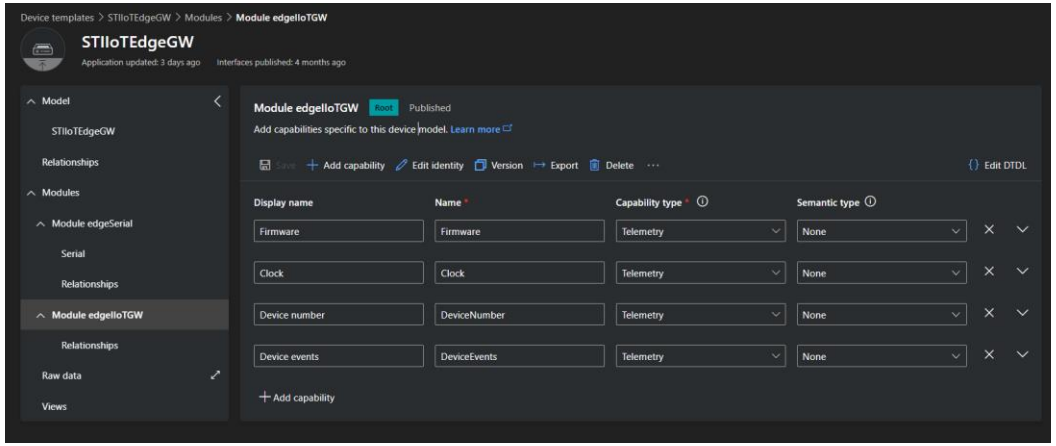
Custom model
Start with a blank model and shape your device from scratch.

Import a model
Start by importing your model file.



Create a specialized interface that inherits capabilities from other, more general interfaces (for example, a "Conference Room" interface that extends a "Room" interface). Learn more

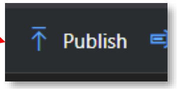
Display name	Name *	Capability type *	Semantic type
List	list	Command	
Connect	connect	Command	
Disconnect	disconnect	Command	
Start	start	Command	
Stop	stop	Command	



Add capabilities specific to this device model. Learn more

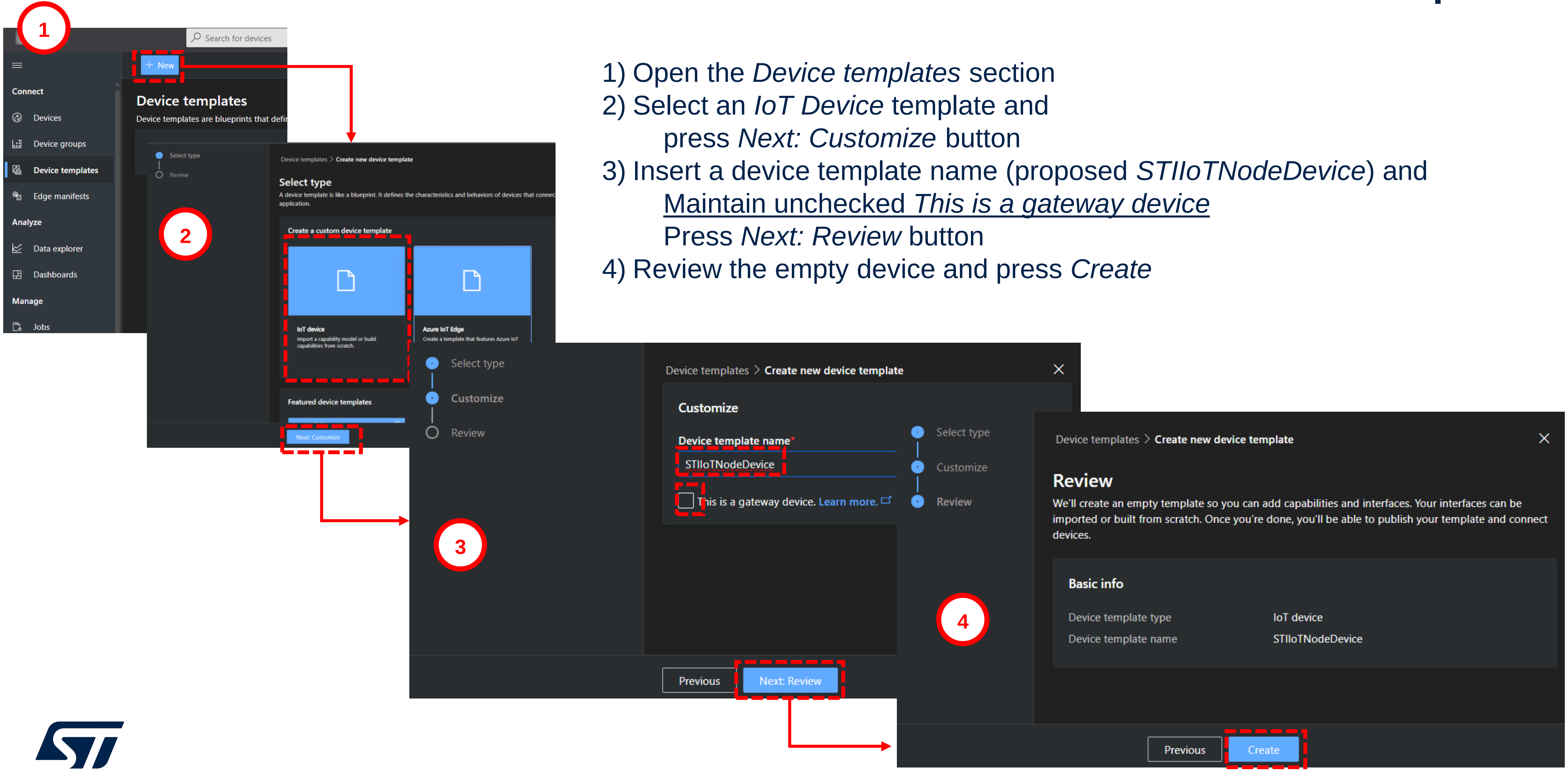
Display name	Name *	Capability type *	Semantic type
Firmware	Firmware	Telemetry	None
Clock	Clock	Telemetry	None
Device number	DeviceNumber	Telemetry	None
Device events	DeviceEvents	Telemetry	None

Important: publish the device template before assigning it to a device



Azure IoT Central – Create Node Device template

1) Open the *Device templates* section
2) Select an *IoT Device* template and
press *Next: Customize* button
3) Insert a device template name (proposed *STIIoTNodeDevice*) and
Maintain unchecked *This is a gateway device*
Press *Next: Review* button
4) Review the empty device and press *Create*



Device templates

Device templates are blueprints that defi

Device templates > Create new device template

Select type

A device template is like a blueprint. It defines the characteristics and behaviors of devices that connec application.

Create a custom device template

IoT device
Import a capability model or build capabilities from scratch.

Azure IoT Edge
Create a template that features Azure IoT

Featured device templates

Device templates > Create new device template

Customize

Device template name*: STIIoTNodeDevice

□ This is a gateway device. Learn more.

Device templates > Create new device template

Review

We'll create an empty template so you can add capabilities and interfaces. Your interfaces can be imported or built from scratch. Once you're done, you'll be able to publish your template and connect devices.

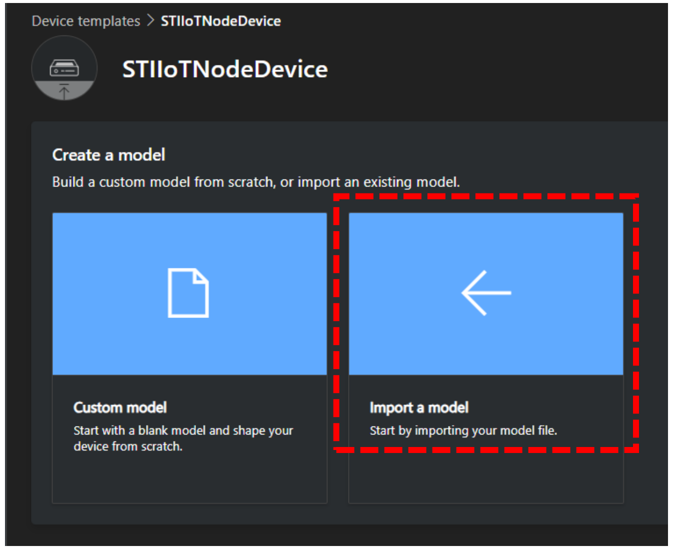
Basic info

Device template type: IoT device
Device template name: STIIoTNodeDevice

Azure IoT Central – Add the model to device template

1) Press Import a model and select the file *STIIoTEndNode_DeviceTemplate_module.json* into the *asset* folder
2) Publish the create device template

Note: the device template name have to be set into the *config_net.json* file



Device templates > STIIoTNodeDevice

STIIoTNodeDevice

Create a model

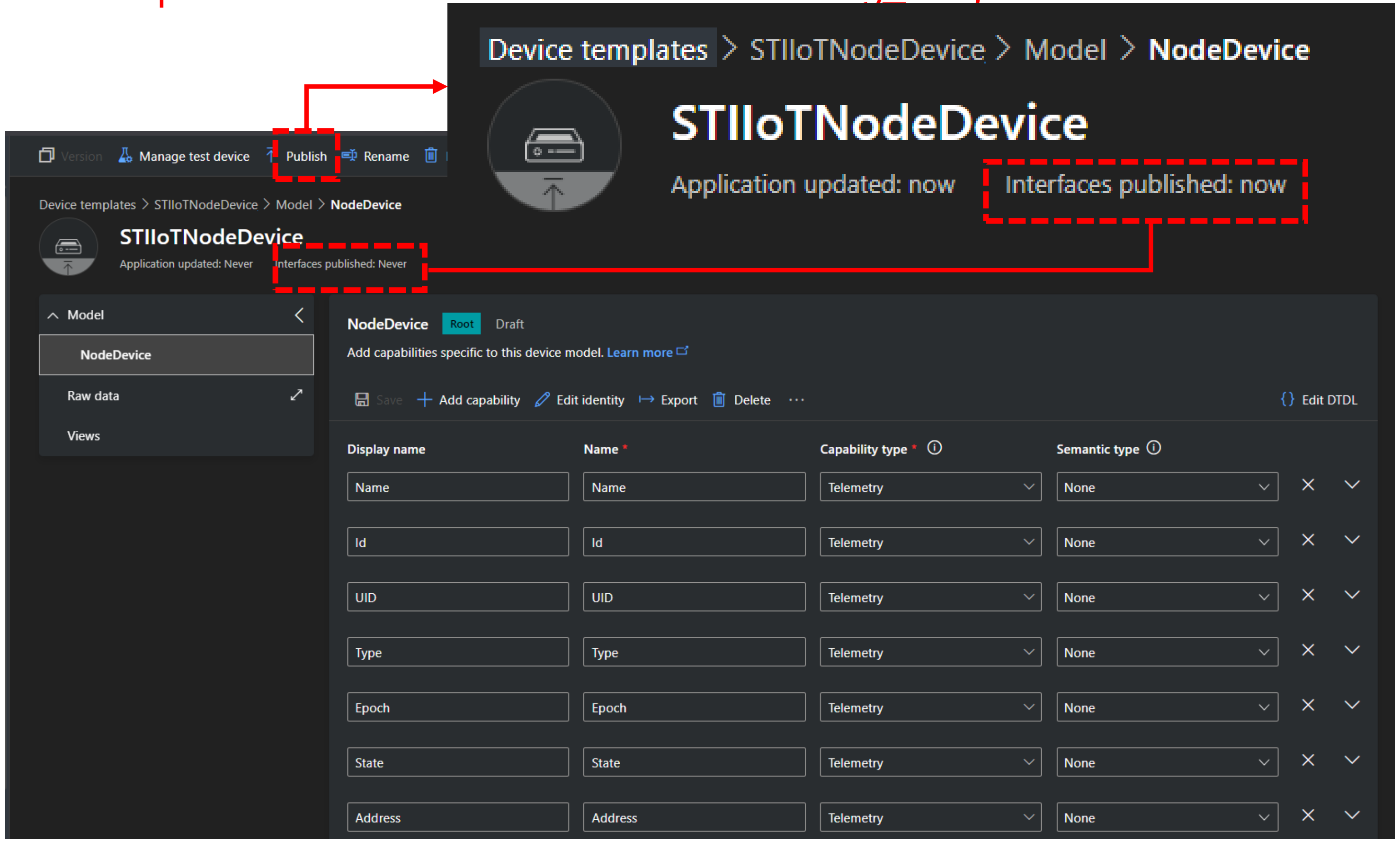
Build a custom model from scratch, or import an existing model.

Custom model

Start with a blank model and shape your device from scratch.

Import a model

Start by importing your model file.



Device templates > STIIoTNodeDevice > Model > NodeDevice

STIIoTNodeDevice

Application updated: Never Interfaces published: Never

Application updated: now Interfaces published: now

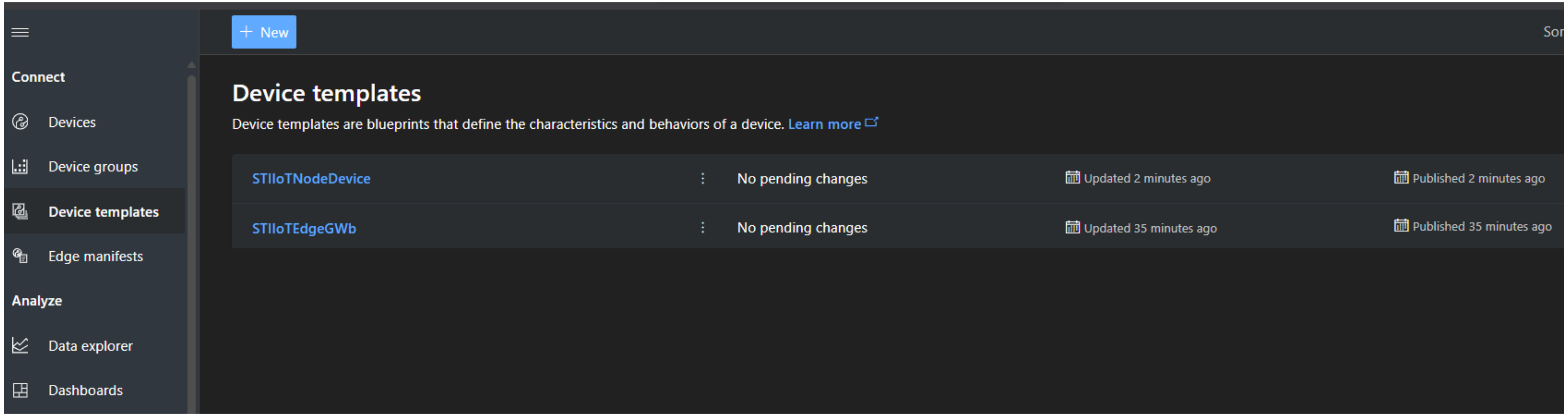
NodeDevice Root Draft

Add capabilities specific to this device model. Learn more

Display name	Name *	Capability type *	Semantic type
Name	Name	Telemetry	None
Id	Id	Telemetry	None
UID	UID	Telemetry	None
Type	Type	Telemetry	None
Epoch	Epoch	Telemetry	None
State	State	Telemetry	None
Address	Address	Telemetry	None

life.augmented

Azure IoT Central – Device templates overview



Device templates

Device templates are blueprints that define the characteristics and behaviors of a device. Learn more

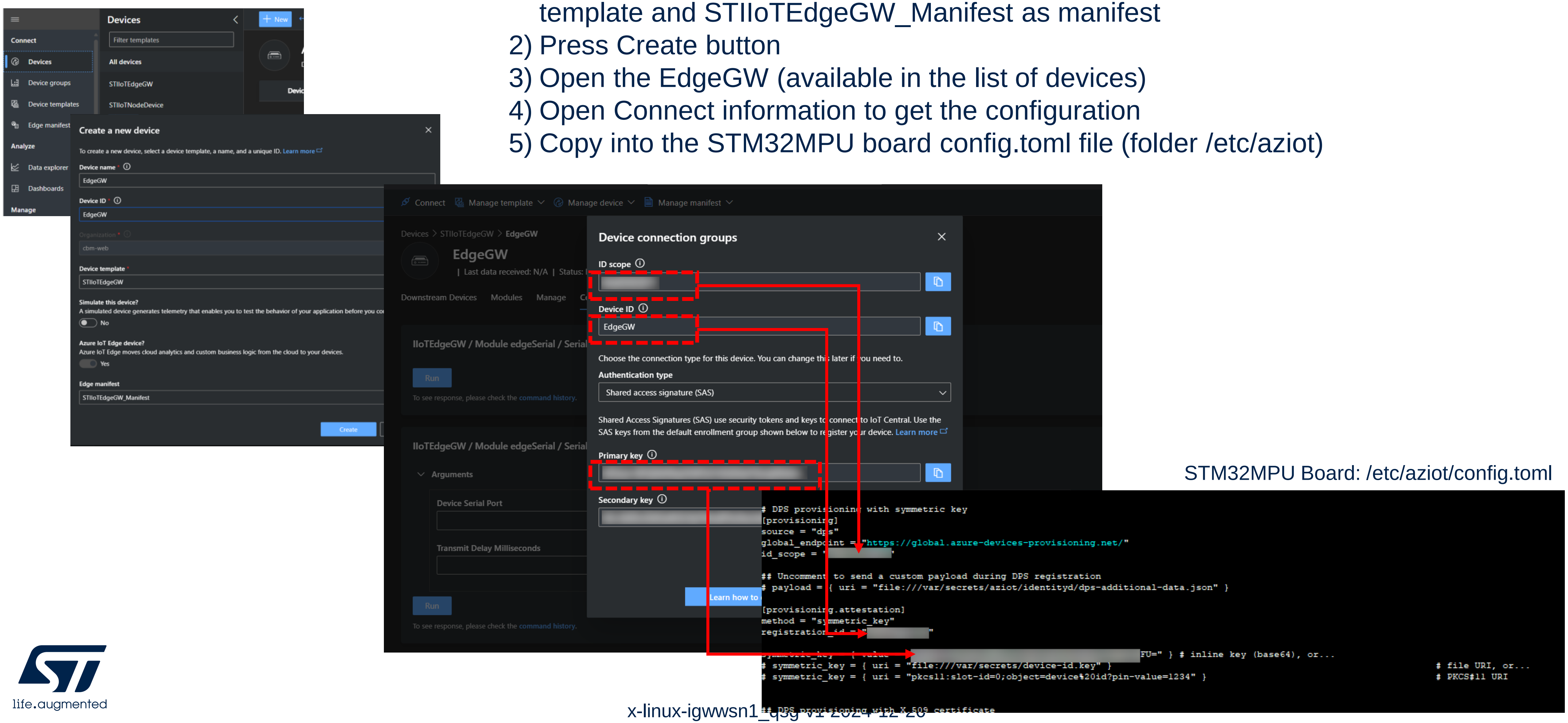
STIIoTNodeDevice	No pending changes	Updated 2 minutes ago	Published 2 minutes ago
STIIoTEdgeGWb	No pending changes	Updated 35 minutes ago	Published 35 minutes ago

Now ready to create the devices

Azure IoT Central – Create IoT Edge Device

and configure the config.toml in the STM32MPU Board

1) Press New button and fill the fields, important select the STIIoTEdgeGW device template and STIIoTEdgeGW_Manifest as manifest
2) Press Create button
3) Open the EdgeGW (available in the list of devices)
4) Open Connect information to get the configuration
5) Copy into the STM32MPU board config.toml file (folder /etc/aziot)



Create a new device

To create a new device, select a device template, a name, and a unique ID. Learn more

Device name *: EdgeGW

Device ID *: EdgeGW

Organization *: cbm-web

Device template *: STIIoTEdgeGW

Simulate this device?
A simulated device generates telemetry that enables you to test the behavior of your application before you co
No

Azure IoT Edge device?
Azure IoT Edge moves cloud analytics and custom business logic from the cloud to your devices.
Yes

Edge manifest: STIIoTEdgeGW_Manifest

Create

To see response, please check the command history.

Device connection groups

ID scope: [redacted]

Device ID: EdgeGW

Choose the connection type for this device. You can change this later if you need to.

Authentication type: Shared access signature (SAS)

Shared Access Signatures (SAS) use security tokens and keys to connect to IoT Central. Use the SAS keys from the default enrollment group shown below to register your device. Learn more

Primary key: [redacted]

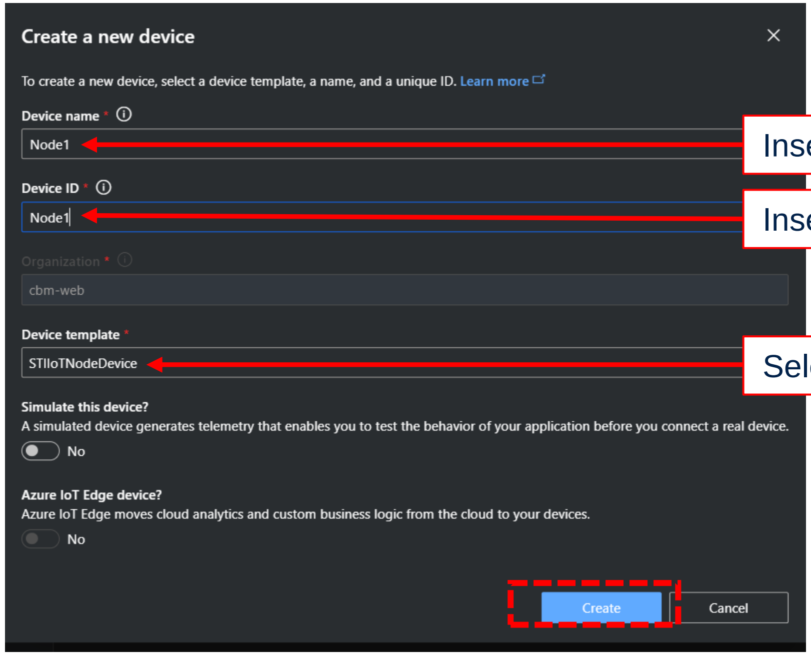
Secondary key: [redacted]

STM32MPU Board: /etc/aziot/config.toml

```
# DPS provisioning with symmetric key
[provisioning]
source = "dps"
global_endpoint = "https://global.azure-devices-provisioning.net/"
id_scope = "[redacted]"

## Uncomment to send a custom payload during DPS registration
# payload = { uri = "file:///var/secrets/aziotd/identityd/dps-additional-data.json" }

[provisioning.attestation]
method = "symmetric_key"
registration_id = "[redacted]"

symmetric_key = { value = "[redacted]FU=" } # inline key (base64), or...
# symmetric_key = { uri = "file:///var/secrets/device-id.key" }                                      # file URI, or...
# symmetric_key = { uri = "pkcs11:slot-id=0;object=device%20id?pin-value=1234" }                     # PKCS#11 URI

## DPS provisioning with X.509 certificate
```

Azure IoT Central – Create Node Devices

Create 4 devices with the STIIoTNodeDevice Device Template (created before)



Create a new device

To create a new device, select a device template, a name, and a unique ID. Learn more

Device name *: Node1

Device ID *: Node1

Organization *: cbm-web

Device template *: STIIoTNodeDevice

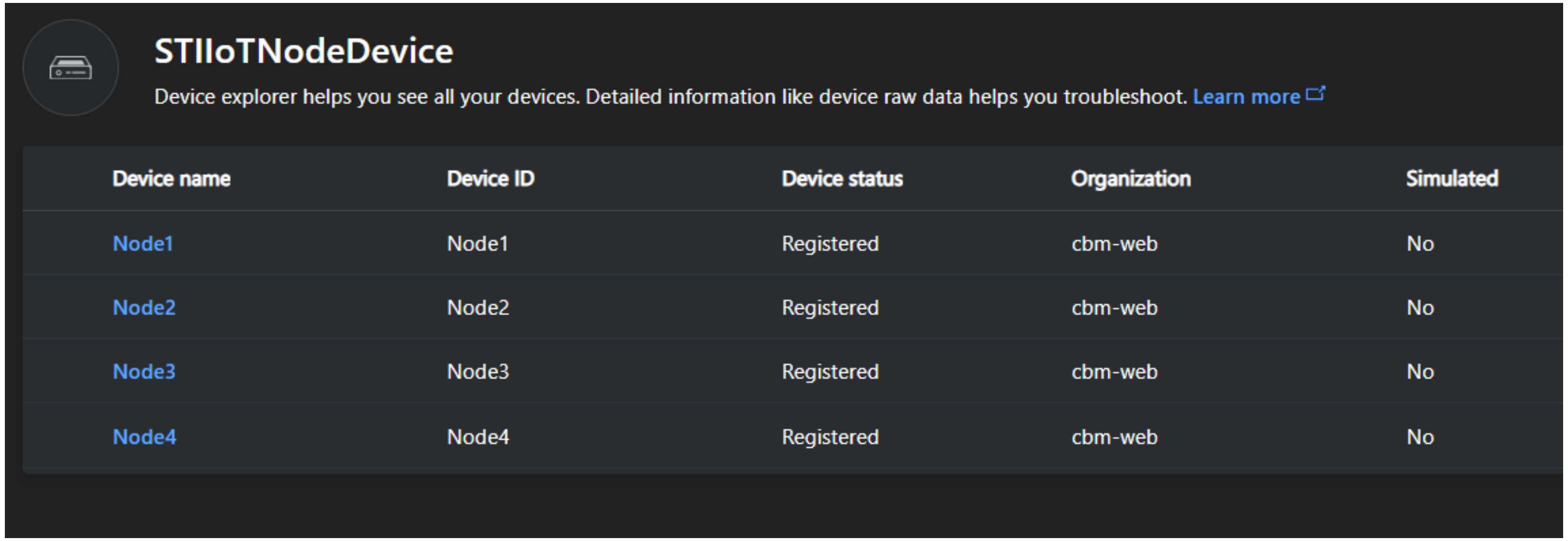
Simulate this device?
A simulated device generates telemetry that enables you to test the behavior of your application before you connect a real device.
No

Azure IoT Edge device?
Azure IoT Edge moves cloud analytics and custom business logic from the cloud to your devices.
No

Insert the preferred device name Node1 (proposed)

Insert the preferred device id: Node1

Select the Device Template: STIIoTNodeDevice



STIIoTNodeDevice

Device explorer helps you see all your devices. Detailed information like device raw data helps you troubleshoot. Learn more

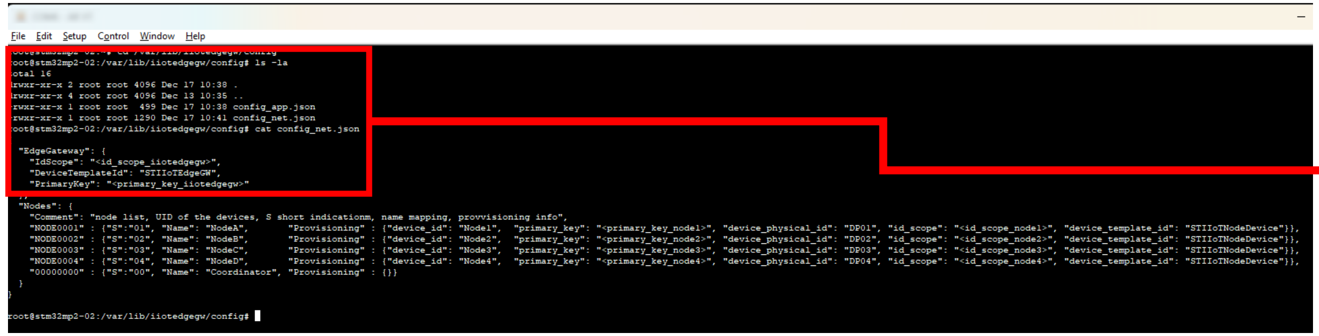
Device name	Device ID	Device status	Organization	Simulated
Node1	Node1	Registered	cbm-web	No
Node2	Node2	Registered	cbm-web	No
Node3	Node3	Registered	cbm-web	No
Node4	Node4	Registered	cbm-web	No

The node device information are copied into the config_net.json file (see next section)

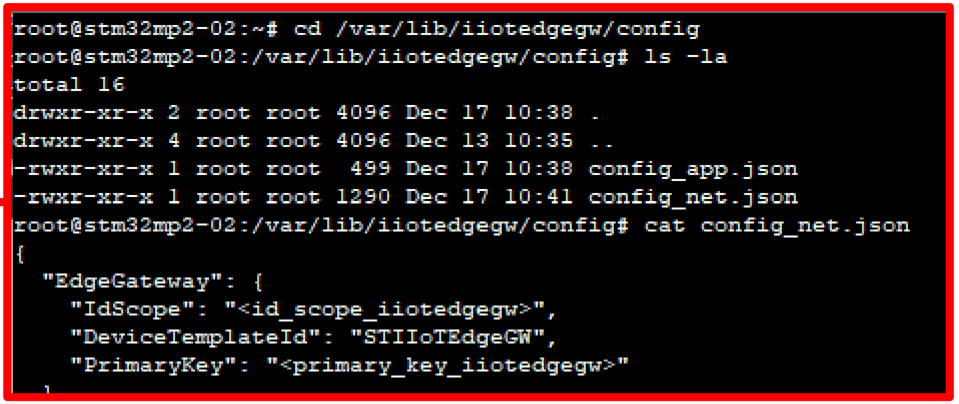
Network configuration

config_net.json (folder */var/lib/iitedgegw/config/)*

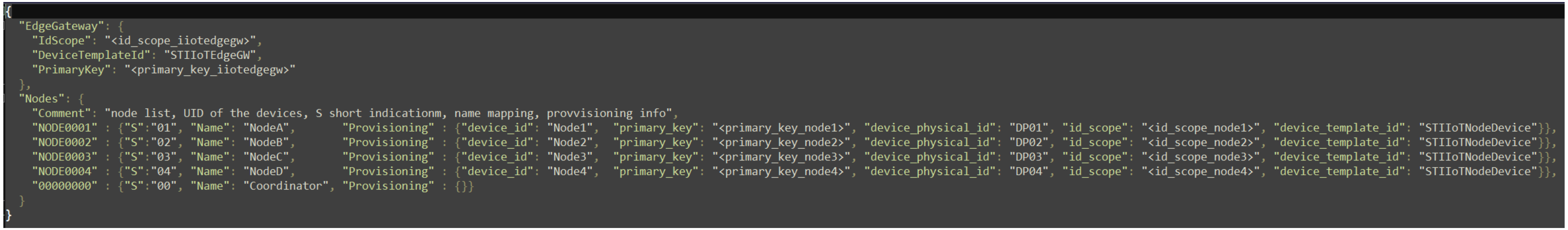
This file must be configured by the user to connect each Node with a digital twin



```
root@stm32mp2-02:~# cd /var/lib/iiotedgegw/config
oot@stm32mp2-02:/var/lib/iiotedgegw/config# ls -la
otal 16
lrwxr-xr-x 2 root root 4096 Dec 17 10:38 .
lrwxr-xr-x 4 root root 4096 Dec 13 10:35 ..
rwxr-xr-x 1 root root  499 Dec 17 10:38 config_app.json
rwxr-xr-x 1 root root 1290 Dec 17 10:41 config_net.json
oot@stm32mp2-02:/var/lib/iiotedgegw/config# cat config_net.json

 "EdgeGateway": {
   "IdScope": "<id_scope_iiotedgegw>",
   "DeviceTemplateId": "STIIoTEdgeGW",
   "PrimaryKey": "<primary_key_iiotedgegw>"
 },
 "Nodes": {
   "Comment": "node list, UID of the devices, S short indicationm, name mapping, provvisioning info",
   "NODE0001" : {"S":"01", "Name": "NodeA",       "Provisioning" : {"device_id": "Node1",  "primary_key": "<primary_key_node1>", "device_physical_id": "DP01", "id_scope": "<id_scope_node1>", "device_template_id": "STIIoTNodeDevice"}},
   "NODE0002" : {"S":"02", "Name": "NodeB",       "Provisioning" : {"device_id": "Node2",  "primary_key": "<primary_key_node2>", "device_physical_id": "DP02", "id_scope": "<id_scope_node2>", "device_template_id": "STIIoTNodeDevice"}},
   "NODE0003" : {"S":"03", "Name": "NodeC",       "Provisioning" : {"device_id": "Node3",  "primary_key": "<primary_key_node3>", "device_physical_id": "DP03", "id_scope": "<id_scope_node3>", "device_template_id": "STIIoTNodeDevice"}},
   "NODE0004" : {"S":"04", "Name": "NodeD",       "Provisioning" : {"device_id": "Node4",  "primary_key": "<primary_key_node4>", "device_physical_id": "DP04", "id_scope": "<id_scope_node4>", "device_template_id": "STIIoTNodeDevice"}},
   "00000000" : {"S":"00", "Name": "Coordinator", "Provisioning" : {}}
 }
}

root@stm32mp2-02:/var/lib/iiotedgegw/config#
```



```
root@stm32mp2-02:~# cd /var/lib/iiotedgegw/config
root@stm32mp2-02:/var/lib/iiotedgegw/config# ls -la
total 16
drwxr-xr-x 2 root root 4096 Dec 17 10:38 .
drwxr-xr-x 4 root root 4096 Dec 13 10:35 ..
-rwxr-xr-x 1 root root  499 Dec 17 10:38 config_app.json
-rwxr-xr-x 1 root root 1290 Dec 17 10:41 config_net.json
root@stm32mp2-02:/var/lib/iiotedgegw/config# cat config_net.json
{
  "EdgeGateway": {
    "IdScope": "<id_scope_iiotedgegw>",
    "DeviceTemplateId": "STIIoTEdgeGW",
    "PrimaryKey": "<primary_key_iiotedgegw>"
```



```
{
  "EdgeGateway": {
    "IdScope": "<id_scope_iiotedgegw>",
    "DeviceTemplateId": "STIIoTEdgeGW",
    "PrimaryKey": "<primary_key_iiotedgegw>"
  },
  "Nodes": {
    "Comment": "node list, UID of the devices, S short indicationm, name mapping, provvisioning info",
    "NODE0001" : {"S":"01", "Name": "NodeA",       "Provisioning" : {"device_id": "Node1",  "primary_key": "<primary_key_node1>", "device_physical_id": "DP01", "id_scope": "<id_scope_node1>", "device_template_id": "STIIoTNodeDevice"}},
    "NODE0002" : {"S":"02", "Name": "NodeB",       "Provisioning" : {"device_id": "Node2",  "primary_key": "<primary_key_node2>", "device_physical_id": "DP02", "id_scope": "<id_scope_node2>", "device_template_id": "STIIoTNodeDevice"}},
    "NODE0003" : {"S":"03", "Name": "NodeC",       "Provisioning" : {"device_id": "Node3",  "primary_key": "<primary_key_node3>", "device_physical_id": "DP03", "id_scope": "<id_scope_node3>", "device_template_id": "STIIoTNodeDevice"}},
    "NODE0004" : {"S":"04", "Name": "NodeD",       "Provisioning" : {"device_id": "Node4",  "primary_key": "<primary_key_node4>", "device_physical_id": "DP04", "id_scope": "<id_scope_node4>", "device_template_id": "STIIoTNodeDevice"}},
    "00000000" : {"S":"00", "Name": "Coordinator", "Provisioning" : {}}
  }
}
```

*this folder is created by the iotedge run-time and the file is created by the application gw module on run

life.augmented

Network configuration

Node1 configuration example

```
"NODE0001" : {
   "S":"01",
   "Name": "Node1",
   "Provisioning" : {
          "device_id": "Node1",
          "primary_key": "<primary key>",
          "device_physical_id": "DP01",
          "id_scope": "<id scope>",
          "device_template_id": "STIIoTNodeDevice"
          }
}
```

Short id (for future feature)

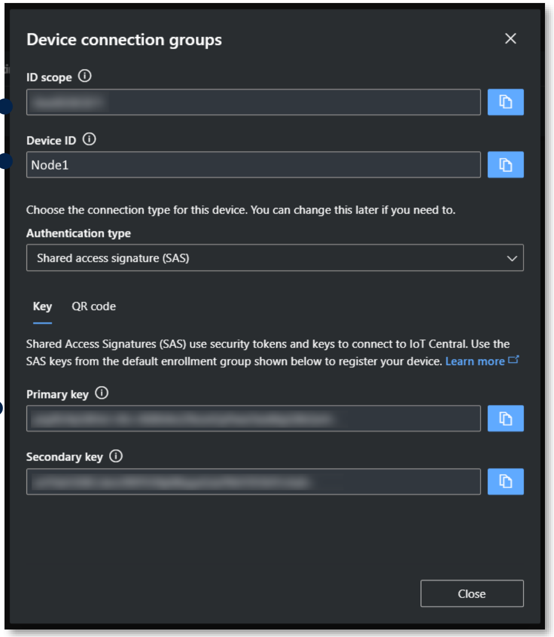
Custom name

Provisioning configuration

Device id

Primary key

Id scope



Device connection groups

ID scope: [redacted]

Device ID: Node1

Choose the connection type for this device. You can change this later if you need to.

Authentication type: Shared access signature (SAS)

Shared Access Signatures (SAS) use security tokens and keys to connect to IoT Central. Use the SAS keys from the default enrollment group shown below to register your device. Learn more

Primary key: [redacted]

Secondary key: [redacted]

- Leave *NODEx* to dynamic association of the STEVAL-PROTEUS1 (in order of connection)
- Edit with the Node UID** to associate a specific STEVAL-PROTEUS1

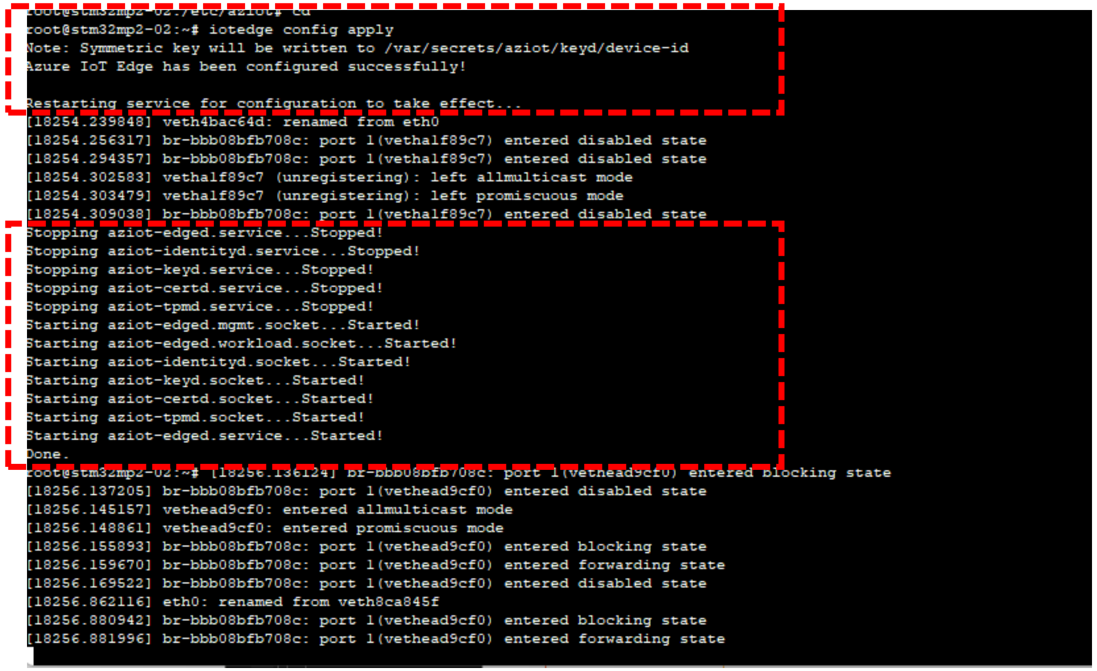
- * The *device_physical_id* is the *digital twin node identification*.
- ** The Node UID is the last 4 bytes of the MCU UID (available in the log STLINK-V3 serial console)

life.augmented

Apply the new configuration

Launch the iotedge config apply

The iotedge stops all services, applies the new configuration and restarts again the services … together there are messages on linux system debug output

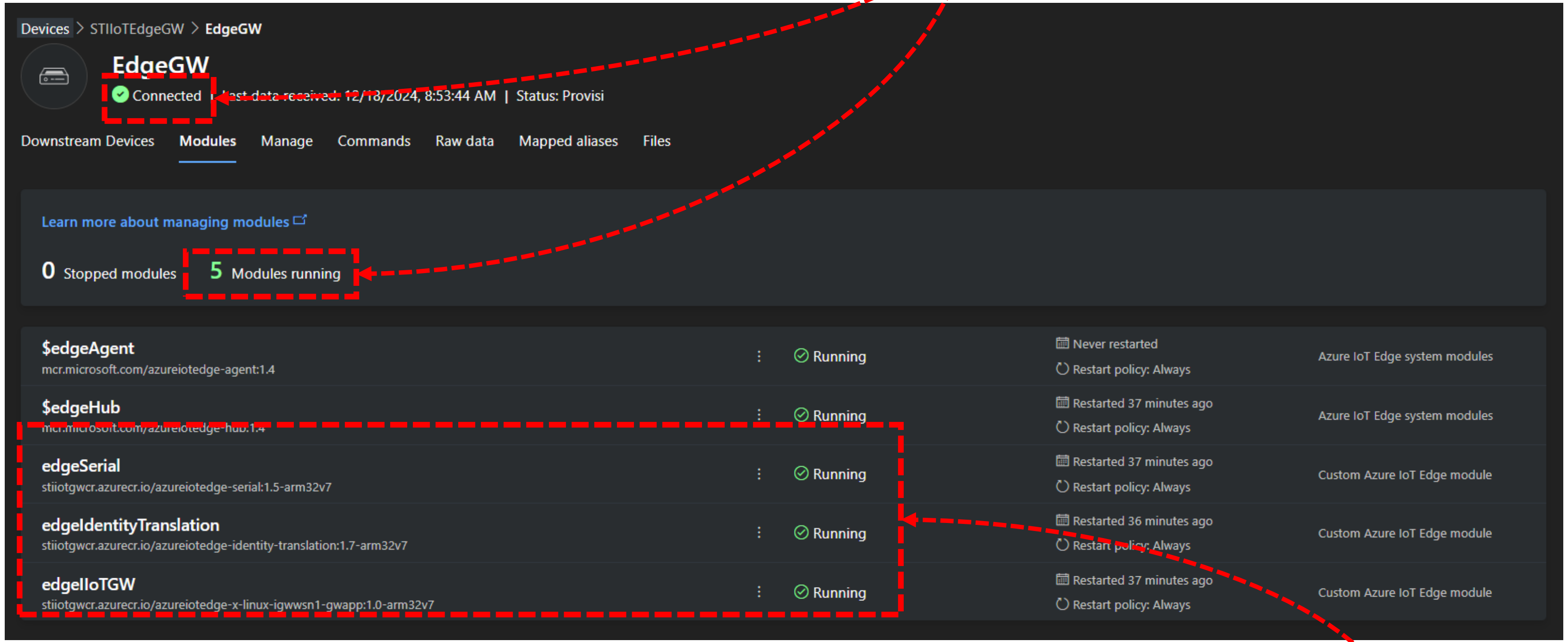


```
root@stm32mp2-02:/etc/aziot# cd
root@stm32mp2-02:~# iotedge config apply
Note: Symmetric key will be written to /var/secrets/aziot/keyd/device-id
Azure IoT Edge has been configured successfully!

Restarting service for configuration to take effect...
[18254.239848] veth4bac64d: renamed from eth0
[18254.256317] br-bbb08bfb708c: port 1(vethalf89c7) entered disabled state
[18254.294357] br-bbb08bfb708c: port 1(vethalf89c7) entered disabled state
[18254.302583] vethalf89c7 (unregistering): left allmulticast mode
[18254.303479] vethalf89c7 (unregistering): left promiscuous mode
[18254.309038] br-bbb08bfb708c: port 1(vethalf89c7) entered disabled state
Stopping aziot-edged.service...Stopped!
Stopping aziot-identityd.service...Stopped!
Stopping aziot-keyd.service...Stopped!
Stopping aziot-certd.service...Stopped!
Stopping aziot-tpmd.service...Stopped!
Starting aziot-edged.mgmt.socket...Started!
Starting aziot-edged.workload.socket...Started!
Starting aziot-identityd.socket...Started!
Starting aziot-keyd.socket...Started!
Starting aziot-certd.socket...Started!
Starting aziot-tpmd.socket...Started!
Starting aziot-edged.service...Started!
Done.
root@stm32mp2-02:~# [18256.136124] br-bbb08bfb708c: port 1(vethead9cf0) entered blocking state
[18256.137205] br-bbb08bfb708c: port 1(vethead9cf0) entered disabled state
[18256.145157] vethead9cf0: entered allmulticast mode
[18256.148861] vethead9cf0: entered promiscuous mode
[18256.155893] br-bbb08bfb708c: port 1(vethead9cf0) entered blocking state
[18256.159670] br-bbb08bfb708c: port 1(vethead9cf0) entered forwarding state
[18256.169522] br-bbb08bfb708c: port 1(vethead9cf0) entered disabled state
[18256.862116] eth0: renamed from veth8ca845f
[18256.880942] br-bbb08bfb708c: port 1(vethead9cf0) entered blocking state
[18256.881996] br-bbb08bfb708c: port 1(vethead9cf0) entered forwarding state
```

life.augmented

System ready

After waiting a little while, the system will be ready.



Devices > STIIoTEdgeGW > EdgeGW

EdgeGW

Connected | Last data received: 12/18/2024, 8:53:44 AM | Status: Provisi

Downstream Devices | Modules | Manage | Commands | Raw data | Mapped aliases | Files

Learn more about managing modules

0 Stopped modules 5 Modules running

$edgeAgent
mcr.microsoft.com/azureiotedge-agent:1.4
Running
Never restarted
Restart policy: Always
Azure IoT Edge system modules

$edgeHub
mcr.microsoft.com/azureiotedge-hub:1.4
Running
Restarted 37 minutes ago
Restart policy: Always
Azure IoT Edge system modules

edgeSerial
stiiotgwcr.azurecr.io/azureiotedge-serial:1.5-arm32v7
Running
Restarted 37 minutes ago
Restart policy: Always
Custom Azure IoT Edge module

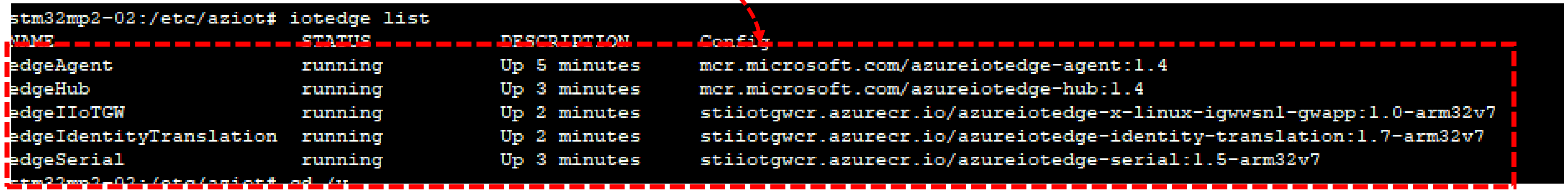
edgeIdentityTranslation
stiiotgwcr.azurecr.io/azureiotedge-identity-translation:1.7-arm32v7
Running
Restarted 36 minutes ago
Restart policy: Always
Custom Azure IoT Edge module

edgeIIoTGW
stiiotgwcr.azurecr.io/azureiotedge-x-linux-igwwsn1-gwapp:1.0-arm32v7
Running
Restarted 37 minutes ago
Restart policy: Always
Custom Azure IoT Edge module

Azure IoTC view

In both system are available the same information

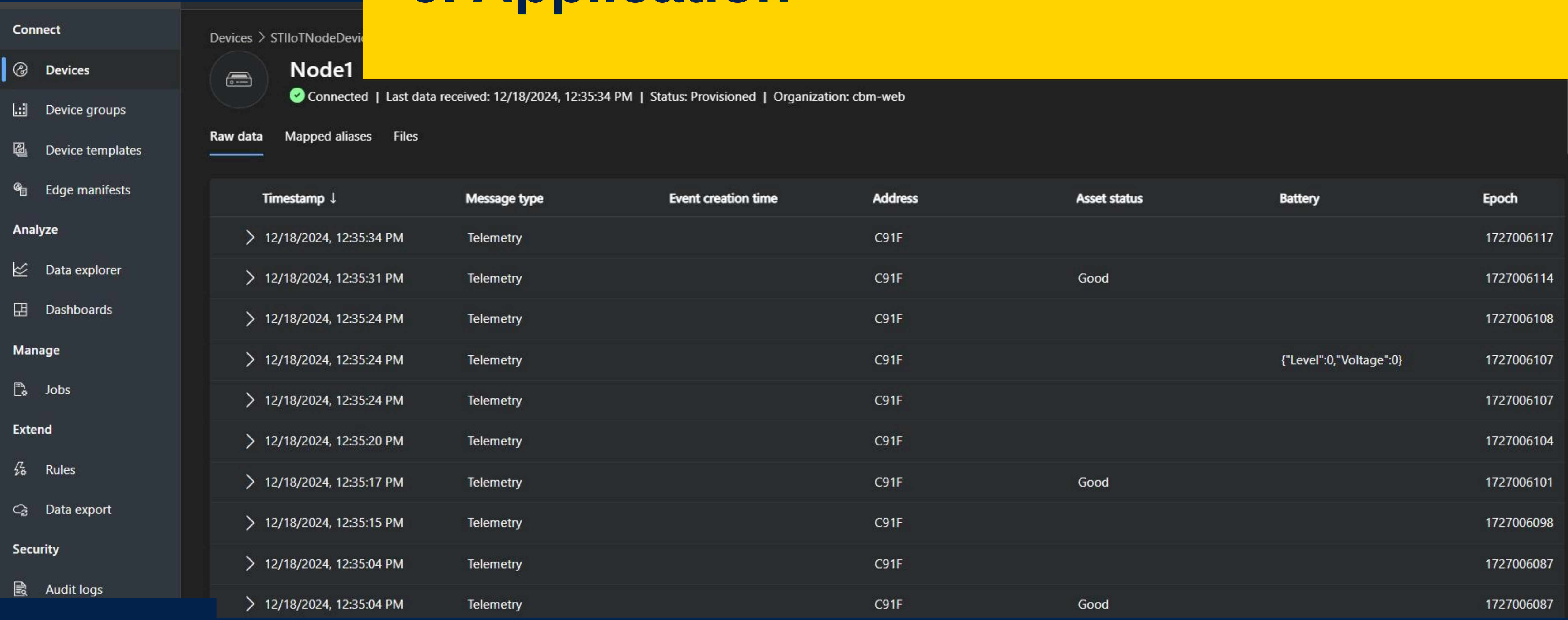
STM32MPU Board view



```
stm32mp2-02:/etc/aziot# iotedge list
NAME                        STATUS      DESCRIPTION     Config
edgeAgent                   running     Up 5 minutes    mcr.microsoft.com/azureiotedge-agent:1.4
edgeHub                     running     Up 3 minutes    mcr.microsoft.com/azureiotedge-hub:1.4
edgeIIoTGW                  running     Up 2 minutes    stiiotgwcr.azurecr.io/azureiotedge-x-linux-igwwsn1-gwapp:1.0-arm32v7
edgeIdentityTranslation     running     Up 2 minutes    stiiotgwcr.azurecr.io/azureiotedge-identity-translation:1.7-arm32v7
edgeSerial                  running     Up 3 minutes    stiiotgwcr.azurecr.io/azureiotedge-serial:1.5-arm32v7
stm32mp2-02:/etc/aziot# cd /u
```

life.augmented

6. Application



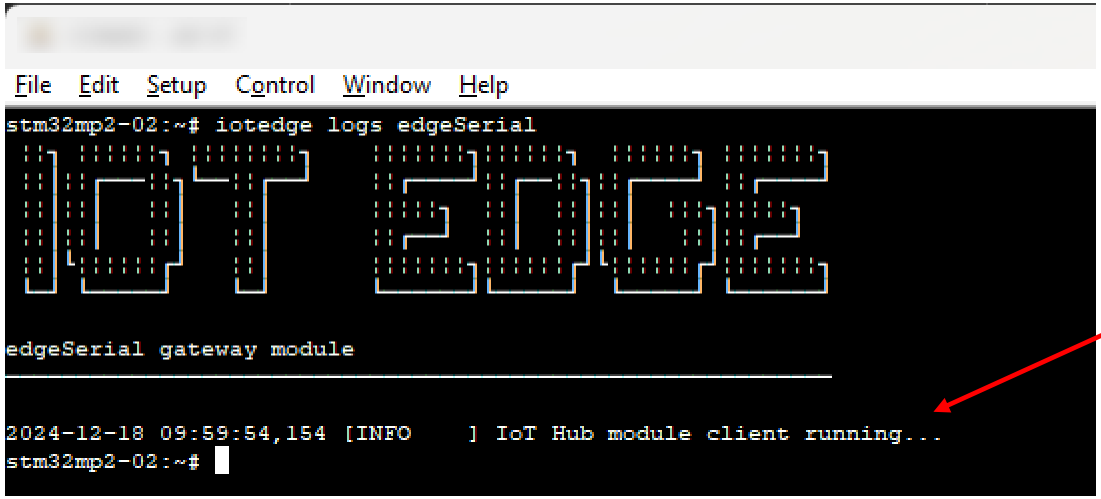
Node1

Connected | Last data received: 12/18/2024, 12:35:34 PM | Status: Provisioned | Organization: cbm-web

Timestamp ↓	Message type	Event creation time	Address	Asset status	Battery	Epoch
12/18/2024, 12:35:34 PM	Telemetry		C91F			1727006117
12/18/2024, 12:35:31 PM	Telemetry		C91F	Good		1727006114
12/18/2024, 12:35:24 PM	Telemetry		C91F			1727006108
12/18/2024, 12:35:24 PM	Telemetry		C91F		{"Level":0,"Voltage":0}	1727006107
12/18/2024, 12:35:24 PM	Telemetry		C91F			1727006107
12/18/2024, 12:35:20 PM	Telemetry		C91F			1727006104
12/18/2024, 12:35:17 PM	Telemetry		C91F	Good		1727006101
12/18/2024, 12:35:15 PM	Telemetry		C91F			1727006098
12/18/2024, 12:35:04 PM	Telemetry		C91F			1727006087
12/18/2024, 12:35:04 PM	Telemetry		C91F	Good		1727006087

General check

- On board: it is possible to know the status of the modules running in the iotedge environment, types:
  - `iotedge logs edgeSerial --tail 10 -f` → to check the serial port status
  - `iotedge logs edgeIdentityTranslation --tail 10 -f` → to show the identity translation activity
  - `iotedge logs edgeIIoTGW --tail 10 -f` → to monitor the GW activity



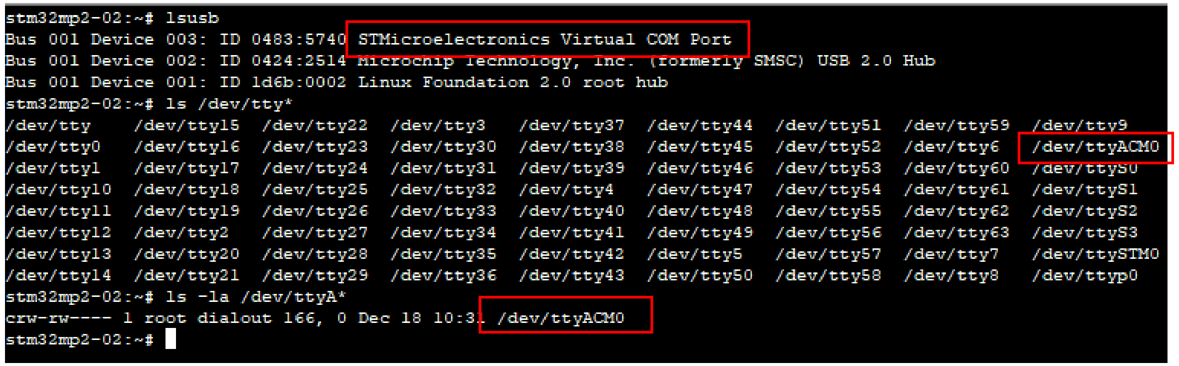
```
stm32mp2-02:~# iotedge logs edgeSerial

edgeSerial gateway module
________________________________________________________________

2024-12-18 09:59:54,154 [INFO    ] IoT Hub module client running...
stm32mp2-02:~#
```

edgeSerial: it is working fine

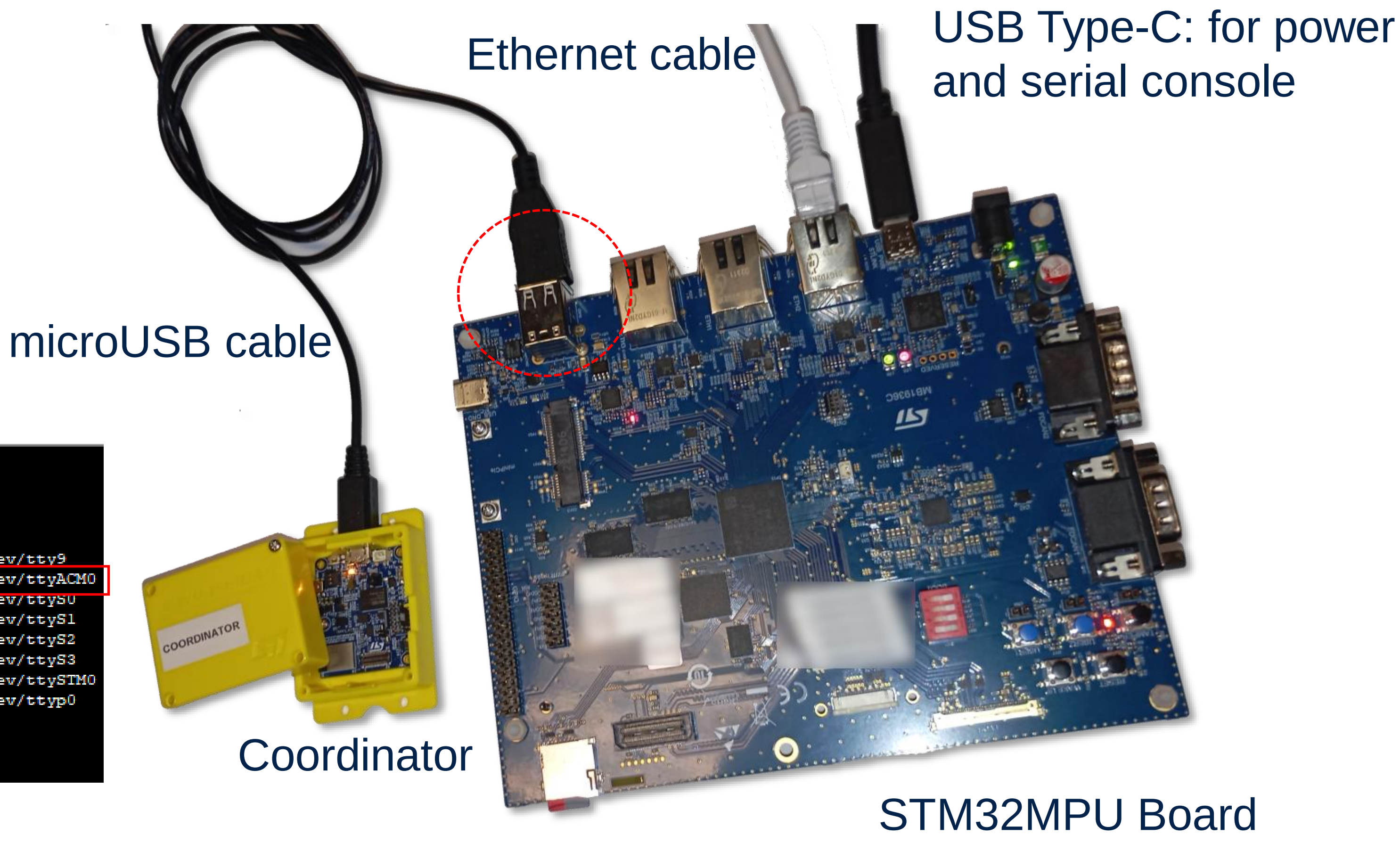
Coordinator: physical connection

The coordinator is a STEVAL-PROTEUS1 connected to the STM32MPU Board with a microUSB cable

Console output



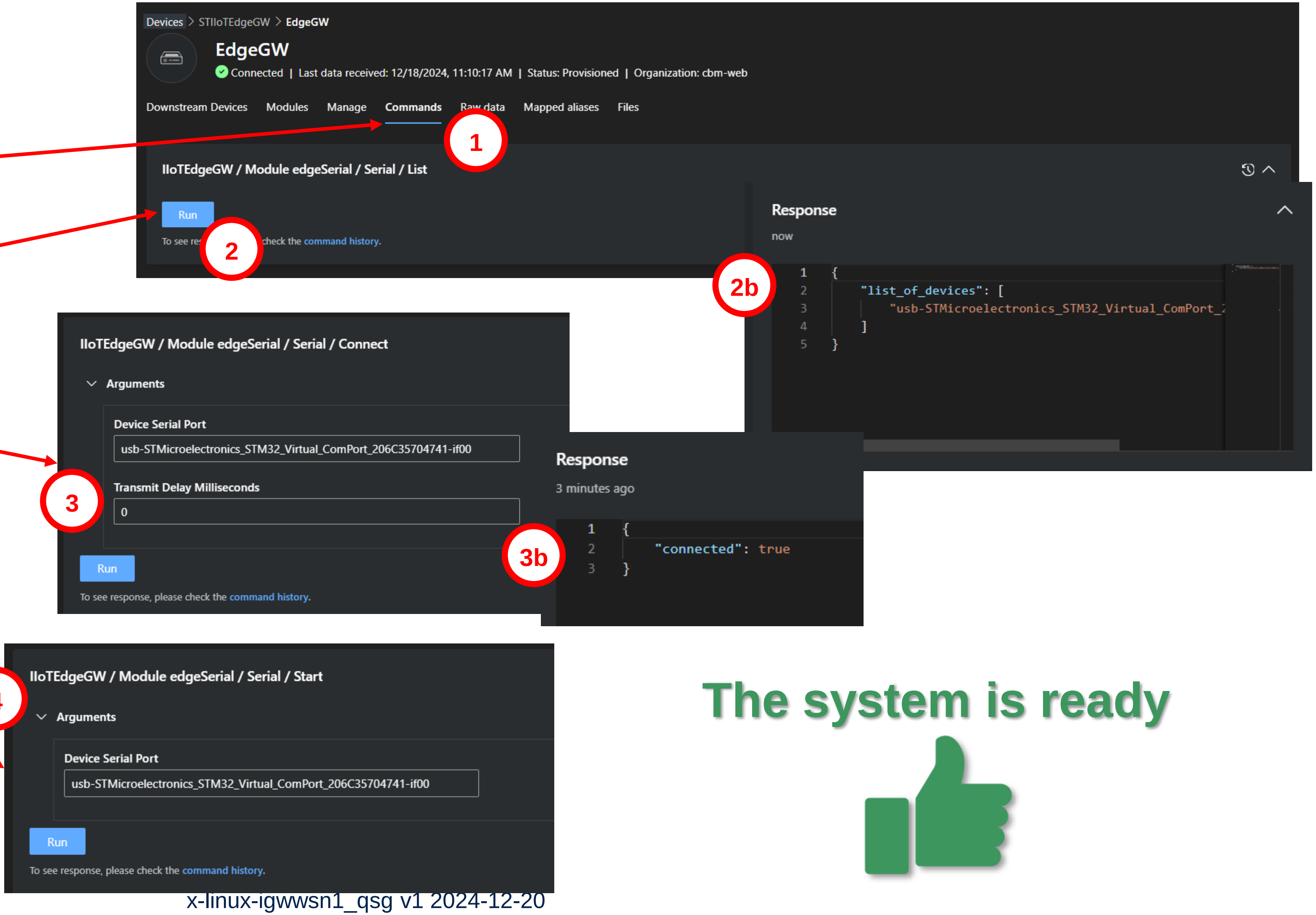
```
stm32mp2-02:~# lsusb
Bus 001 Device 003: ID 0483:5740 STMicroelectronics Virtual COM Port
Bus 001 Device 002: ID 0424:2514 Microchip Technology, Inc. (formerly SMSC) USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
stm32mp2-02:~# ls /dev/tty*
/dev/tty    /dev/tty15  /dev/tty22  /dev/tty3   /dev/tty37  /dev/tty44  /dev/tty51  /dev/tty59  /dev/tty9
/dev/tty0   /dev/tty16  /dev/tty23  /dev/tty30  /dev/tty38  /dev/tty45  /dev/tty52  /dev/tty6   /dev/ttyACM0
/dev/tty1   /dev/tty17  /dev/tty24  /dev/tty31  /dev/tty39  /dev/tty46  /dev/tty53  /dev/tty60  /dev/ttyS0
/dev/tty10  /dev/tty18  /dev/tty25  /dev/tty32  /dev/tty4   /dev/tty47  /dev/tty54  /dev/tty61  /dev/ttyS1
/dev/tty11  /dev/tty19  /dev/tty26  /dev/tty33  /dev/tty40  /dev/tty48  /dev/tty55  /dev/tty62  /dev/ttyS2
/dev/tty12  /dev/tty2   /dev/tty27  /dev/tty34  /dev/tty41  /dev/tty49  /dev/tty56  /dev/tty63  /dev/ttyS3
/dev/tty13  /dev/tty20  /dev/tty28  /dev/tty35  /dev/tty42  /dev/tty5   /dev/tty57  /dev/tty7   /dev/ttySTM0
/dev/tty14  /dev/tty21  /dev/tty29  /dev/tty36  /dev/tty43  /dev/tty50  /dev/tty58  /dev/tty8   /dev/ttyp0
stm32mp2-02:~# ls -la /dev/ttyA*
crw-rw---- 1 root dialout 166, 0 Dec 18 10:31 /dev/ttyACM0
stm32mp2-02:~#
```



Coordinator: open serial from remotely

In the Azure IoT Central

1. Select the device and open the **commands** tab
2. Then press **Run** in Serial / List command and open the response
3. Copy the *vcom id* and press **Run** in Serial / Connect, then check the response
4. Copy the *vcom id* and in Serial / Start and press **Run**



✓ Connected | Last data received: 12/18/2024, 11:10:17 AM | Status: Provisioned | Organization: cbm-web

IIoTEdgeGW / Module edgeSerial / Serial / List

To see res check the command history.

Response

now

```
{
    "list_of_devices": [
        "usb-STMicroelectronics_STM32_Virtual_ComPort_2
    ]
}
```

IIoTEdgeGW / Module edgeSerial / Serial / Connect

Device Serial Port: usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00

Transmit Delay Milliseconds: 0

To see response, please check the command history.

Response

3 minutes ago

```
{
    "connected": true
}
```

IIoTEdgeGW / Module edgeSerial / Serial / Start

Device Serial Port: usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00

To see response, please check the command history.

The system is ready

Coordinator: debug show

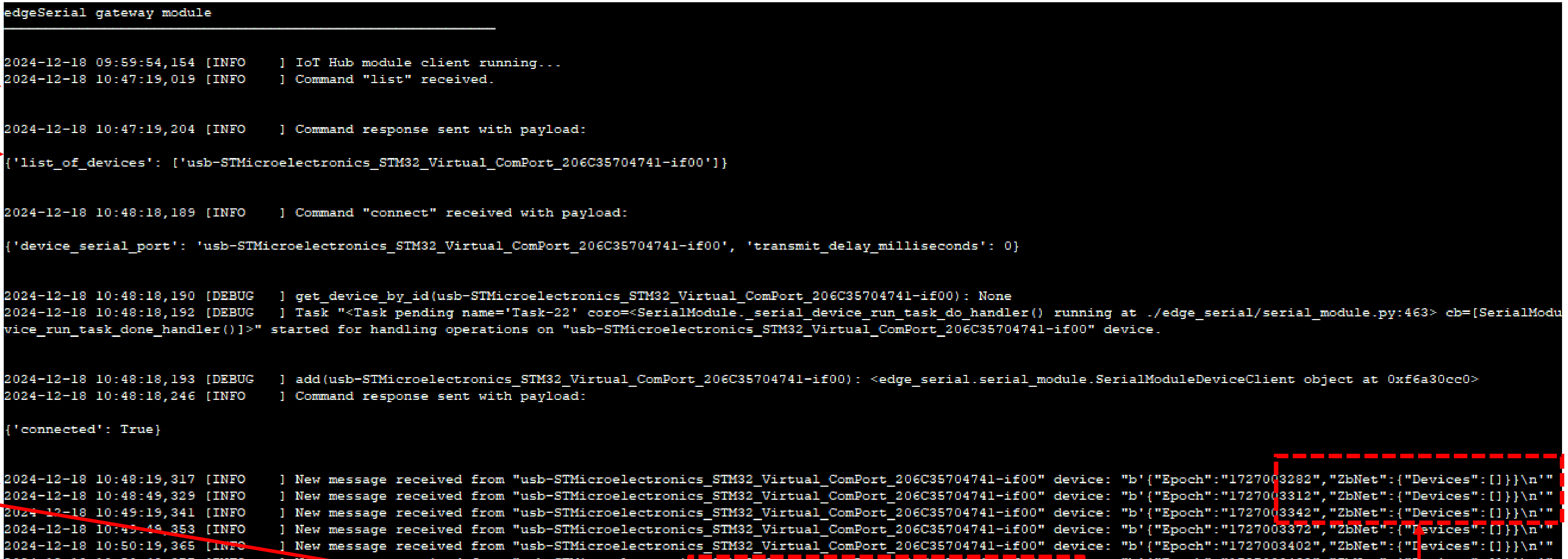
Log commands

List command and response

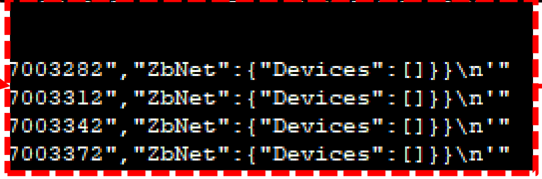
Connect command and response

Start command and response

List node received from the serial port (empty on start-up)



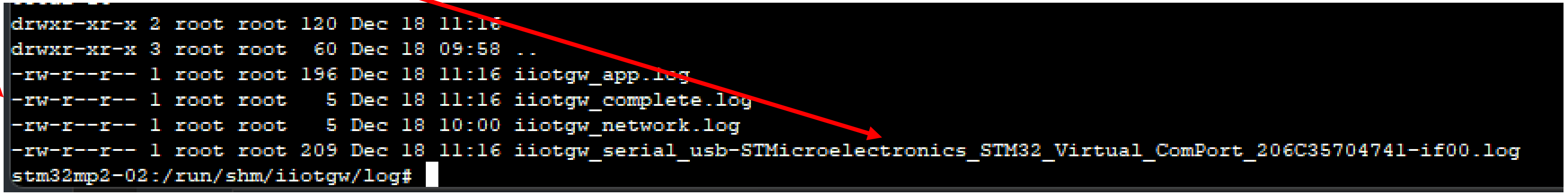
```
edgeSerial gateway module
_______________________________________________________________

2024-12-18 09:59:54,154 [INFO    ] IoT Hub module client running...
2024-12-18 10:47:19,019 [INFO    ] Command "list" received.


2024-12-18 10:47:19,204 [INFO    ] Command response sent with payload:

{'list_of_devices': ['usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00']}


2024-12-18 10:48:18,189 [INFO    ] Command "connect" received with payload:

{'device_serial_port': 'usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00', 'transmit_delay_milliseconds': 0}


2024-12-18 10:48:18,190 [DEBUG   ] get_device_by_id(usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00): None
2024-12-18 10:48:18,192 [DEBUG   ] Task "<Task pending name='Task-22' coro=<SerialModule._serial_device_run_task_do_handler() running at ./edge_serial/serial_module.py:463> cb=[SerialModu
vice_run_task_done_handler()]>" started for handling operations on "usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00" device.


2024-12-18 10:48:18,193 [DEBUG   ] add(usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00): <edge_serial.serial_module.SerialModuleDeviceClient object at 0xf6a30cc0>
2024-12-18 10:48:18,246 [INFO    ] Command response sent with payload:

{'connected': True}


2024-12-18 10:48:19,317 [INFO    ] New message received from "usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00" device: "b'{"Epoch":"1727003282","ZbNet":{"Devices":[]}}\n'"
2024-12-18 10:48:49,329 [INFO    ] New message received from "usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00" device: "b'{"Epoch":"1727003312","ZbNet":{"Devices":[]}}\n'"
2024-12-18 10:49:19,341 [INFO    ] New message received from "usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00" device: "b'{"Epoch":"1727003342","ZbNet":{"Devices":[]}}\n'"
2024-12-18 10:49:49,353 [INFO    ] New message received from "usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00" device: "b'{"Epoch":"1727003372","ZbNet":{"Devices":[]}}\n'"
2024-12-18 10:50:19,365 [INFO    ] New message received from "usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00" device: "b'{"Epoch":"1727003402","ZbNet":{"Devices":[]}}\n'"
```



```
7003282","ZbNet":{"Devices":[]}}\n'"
7003312","ZbNet":{"Devices":[]}}\n'"
7003342","ZbNet":{"Devices":[]}}\n'"
7003372","ZbNet":{"Devices":[]}}\n'"
```

Log folder /run/shm/iiotgw/log/

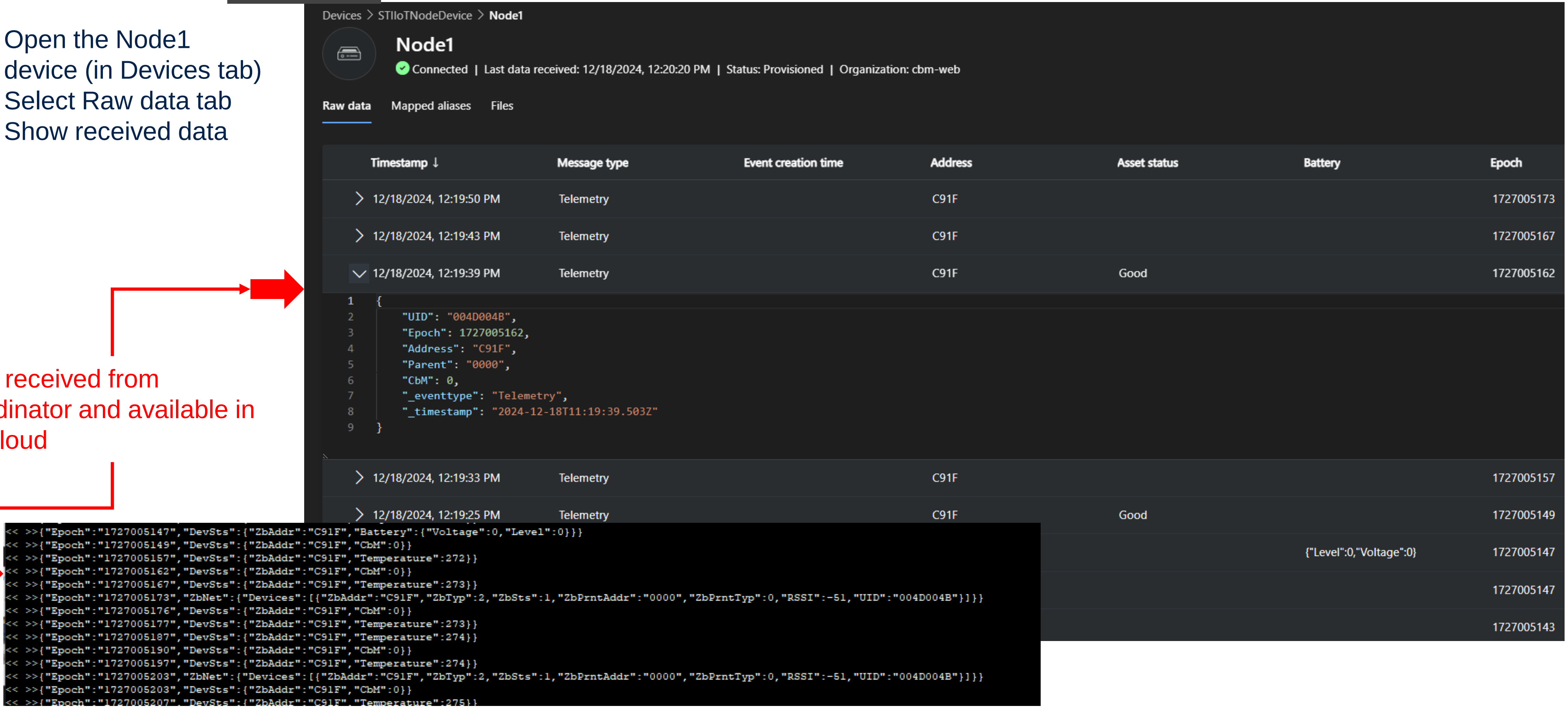
Serial output log



```
drwxr-xr-x 2 root root 120 Dec 18 11:16 .
drwxr-xr-x 3 root root  60 Dec 18 09:58 ..
-rw-r--r-- 1 root root 196 Dec 18 11:16 iiotgw_app.log
-rw-r--r-- 1 root root   5 Dec 18 11:16 iiotgw_complete.log
-rw-r--r-- 1 root root   5 Dec 18 10:00 iiotgw_network.log
-rw-r--r-- 1 root root 209 Dec 18 11:16 iiotgw_serial_usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00.log
stm32mp2-02:/run/shm/iiotgw/log#
```

Show data

- Open the Node1 device (in Devices tab)
- Select Raw data tab
- Show received data



Node1

Connected | Last data received: 12/18/2024, 12:20:20 PM | Status: Provisioned | Organization: cbm-web

	Timestamp ↓	Message type	Event creation time	Address	Asset status	Battery	Epoch
>	12/18/2024, 12:19:50 PM	Telemetry		C91F			1727005173
>	12/18/2024, 12:19:43 PM	Telemetry		C91F			1727005167
∨	12/18/2024, 12:19:39 PM	Telemetry		C91F	Good		1727005162

```
{
    "UID": "004D004B",
    "Epoch": 1727005162,
    "Address": "C91F",
    "Parent": "0000",
    "CbM": 0,
    "_eventtype": "Telemetry",
    "_timestamp": "2024-12-18T11:19:39.503Z"
}
```

	Timestamp ↓	Message type	Event creation time	Address	Asset status	Battery	Epoch
>	12/18/2024, 12:19:33 PM	Telemetry		C91F			1727005157
>	12/18/2024, 12:19:25 PM	Telemetry		C91F	Good		1727005149
						{"Level":0,"Voltage":0}	1727005147
							1727005147
							1727005143

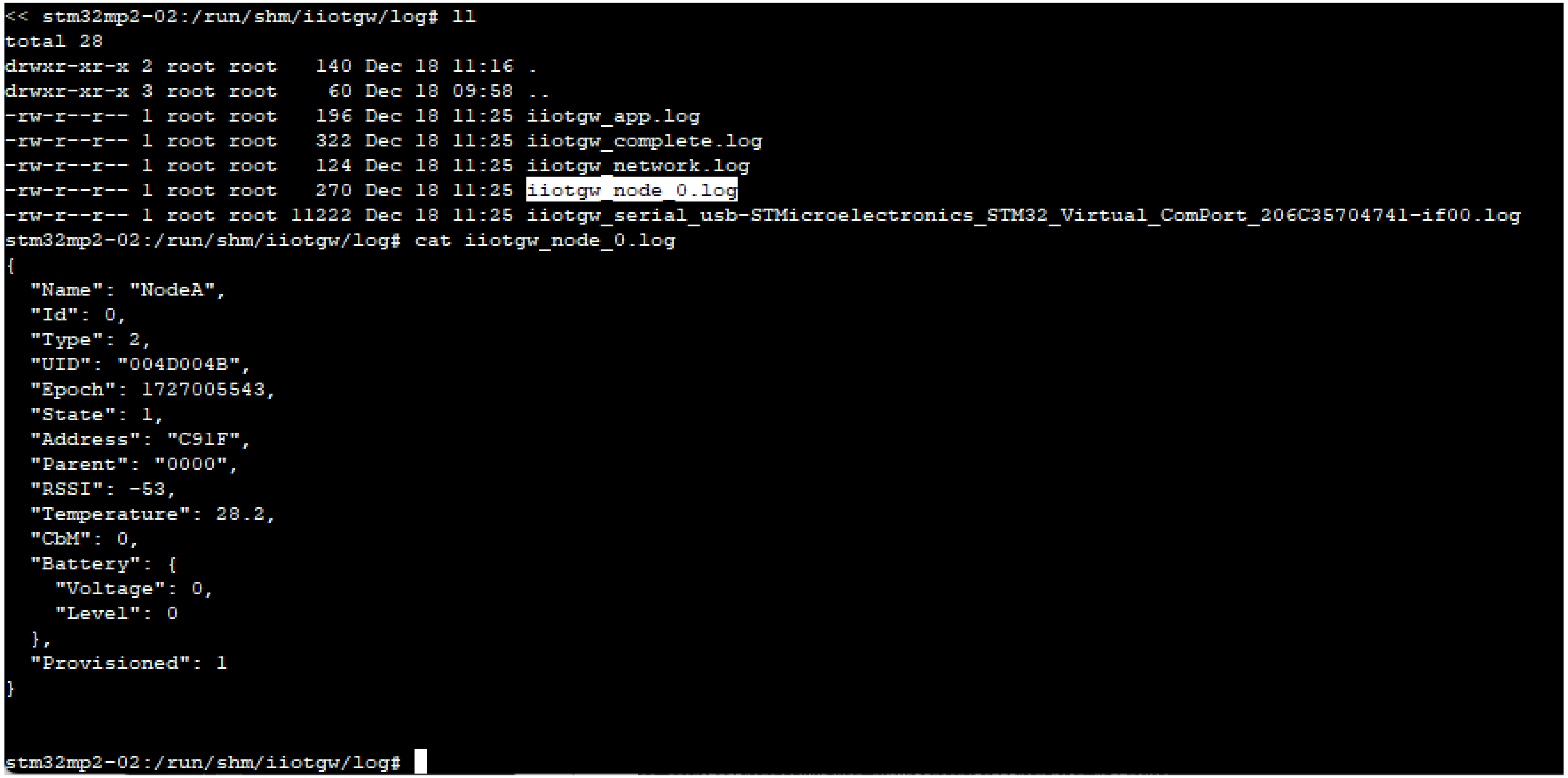
Data received from coordinator and available in the cloud

```
<< >>{"Epoch":"1727005147","DevSts":{"ZbAddr":"C91F","Battery":{"Voltage":0,"Level":0}}}
<< >>{"Epoch":"1727005149","DevSts":{"ZbAddr":"C91F","CbM":0}}
<< >>{"Epoch":"1727005157","DevSts":{"ZbAddr":"C91F","Temperature":272}}
<< >>{"Epoch":"1727005162","DevSts":{"ZbAddr":"C91F","CbM":0}}
<< >>{"Epoch":"1727005167","DevSts":{"ZbAddr":"C91F","Temperature":273}}
<< >>{"Epoch":"1727005173","ZbNet":{"Devices":[{"ZbAddr":"C91F","ZbTyp":2,"ZbSts":1,"ZbPrntAddr":"0000","ZbPrntTyp":0,"RSSI":-51,"UID":"004D004B"}]}}
<< >>{"Epoch":"1727005176","DevSts":{"ZbAddr":"C91F","CbM":0}}
<< >>{"Epoch":"1727005177","DevSts":{"ZbAddr":"C91F","Temperature":273}}
<< >>{"Epoch":"1727005187","DevSts":{"ZbAddr":"C91F","Temperature":274}}
<< >>{"Epoch":"1727005190","DevSts":{"ZbAddr":"C91F","CbM":0}}
<< >>{"Epoch":"1727005197","DevSts":{"ZbAddr":"C91F","Temperature":274}}
<< >>{"Epoch":"1727005203","ZbNet":{"Devices":[{"ZbAddr":"C91F","ZbTyp":2,"ZbSts":1,"ZbPrntAddr":"0000","ZbPrntTyp":0,"RSSI":-51,"UID":"004D004B"}]}}
<< >>{"Epoch":"1727005203","DevSts":{"ZbAddr":"C91F","CbM":0}}
<< >>{"Epoch":"1727005207","DevSts":{"ZbAddr":"C91F","Temperature":275}}
```

life.augmented

Show Data

The data are available in real time directly in the STM32MPU board for all or for a specific node (iiotgw_node_x.log *x 0-based index*)



```
<< stm32mp2-02:/run/shm/iiotgw/log# ll
total 28
drwxr-xr-x 2 root root   140 Dec 18 11:16 .
drwxr-xr-x 3 root root    60 Dec 18 09:58 ..
-rw-r--r-- 1 root root   196 Dec 18 11:25 iiotgw_app.log
-rw-r--r-- 1 root root   322 Dec 18 11:25 iiotgw_complete.log
-rw-r--r-- 1 root root   124 Dec 18 11:25 iiotgw_network.log
-rw-r--r-- 1 root root   270 Dec 18 11:25 iiotgw_node_0.log
-rw-r--r-- 1 root root 11222 Dec 18 11:25 iiotgw_serial_usb-STMicroelectronics_STM32_Virtual_ComPort_206C35704741-if00.log
stm32mp2-02:/run/shm/iiotgw/log# cat iiotgw_node_0.log
{
  "Name": "NodeA",
  "Id": 0,
  "Type": 2,
  "UID": "004D004B",
  "Epoch": 1727005543,
  "State": 1,
  "Address": "C91F",
  "Parent": "0000",
  "RSSI": -53,
  "Temperature": 28.2,
  "CbM": 0,
  "Battery": {
    "Voltage": 0,
    "Level": 0
  },
  "Provisioned": 1
}


stm32mp2-02:/run/shm/iiotgw/log# 
```

More information on

- www.st.com
- www.st.com/en/embedded-software/x-linux-igwwsn.html
  - DB5376
- STMicroelectronics/x-linux-igwwsn1 →source code

Our technology starts with You

Find out more at www.st.com

life.augmented