life.augmented

ST8500 Hybrid PLC&RF connectivity development kit – User Terminal quick start guide

Jun 2021

Introduction

- Scope of this presentation is to introduce the user Terminal available from EVLKST8500GH868 or EVLKST8500GH915 STM32 Nucleo board USB port.
- This terminal allows user to get into the evaluation of some features like UDP data transfers via some launchable tests/demo.

- Required elements:
  - PC with a terminal tool installed like Teraterm.
  - Get 2 of EVLKST8500GH868 or 2 of EVLKST8500GH915 boards in order to be able to run data transfers from one PLC/RF node to the other.
  - Update boards with latest STSW-ST8500GH package available at: https://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-st8500gh.html

Sections

1 Preliminary steps

2 General handling

3 Terminal UDP data transfer tests: ‘simple’ UDP Data Req test

4 Terminal UDP data transfer tests: ‘loopback’ UDP Data Req test

life.augmented

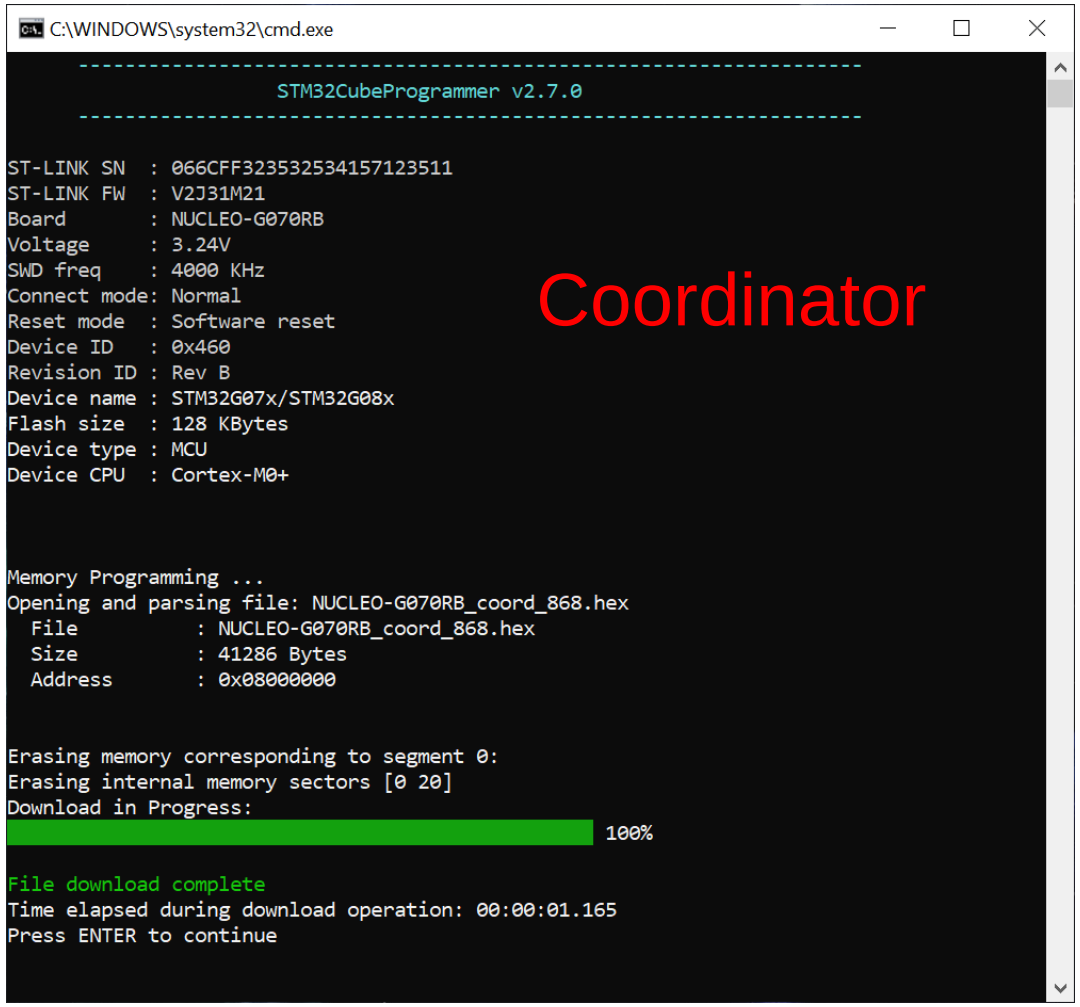
Preliminary steps

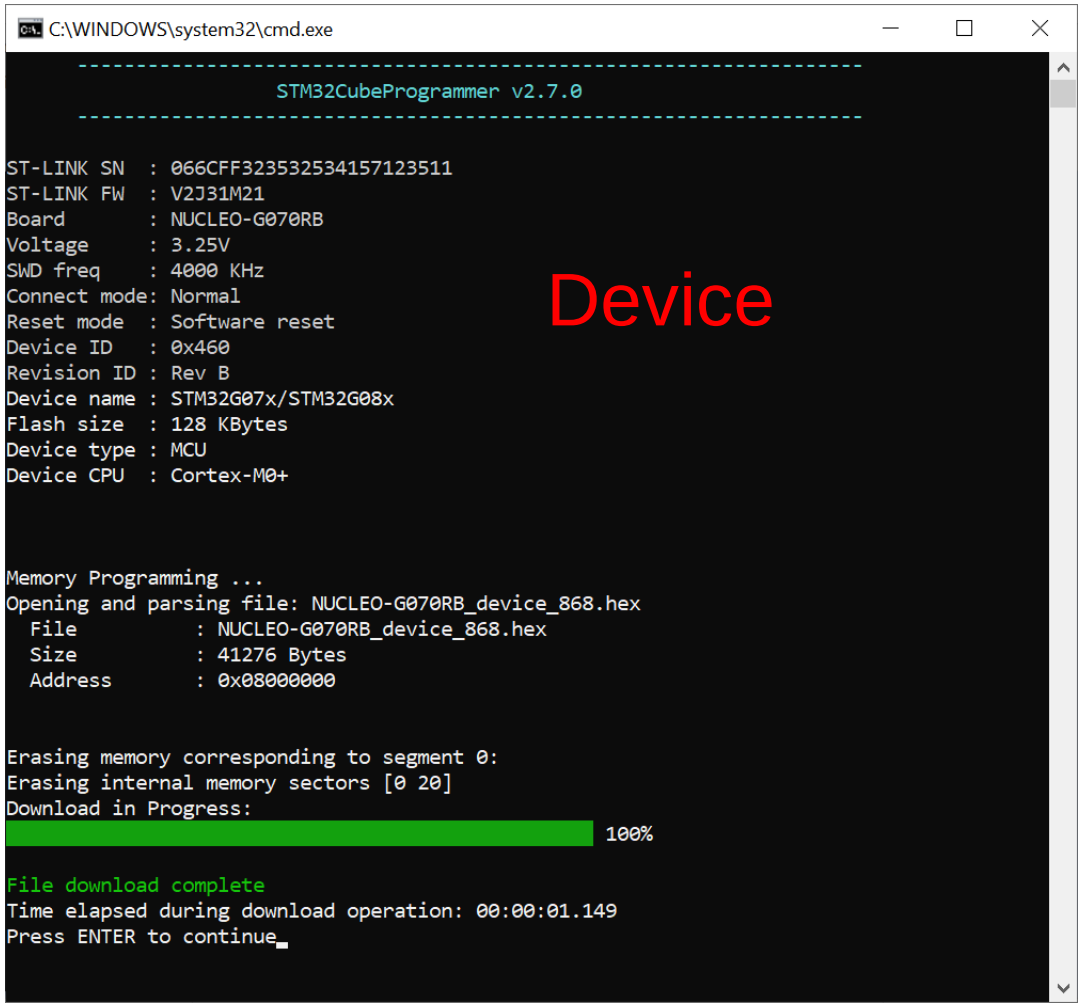
NUCLEO-G070RB FW upgrade

- After installing the STM32 CubeProgrammer (available on https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stm32cubeprog.html), connect the USB cable to the NUCLEO-G070RB USB connector and launch the appropriate batch (.bat) file, based on the network role and kit type (EVLKST8500GH868 or EVLKST8500GH915).
- Please note that for each G3-PLC network only one node must be programmed as Coordinator.
- Please connect one EVLKST8500GH at a time.

life.augmented

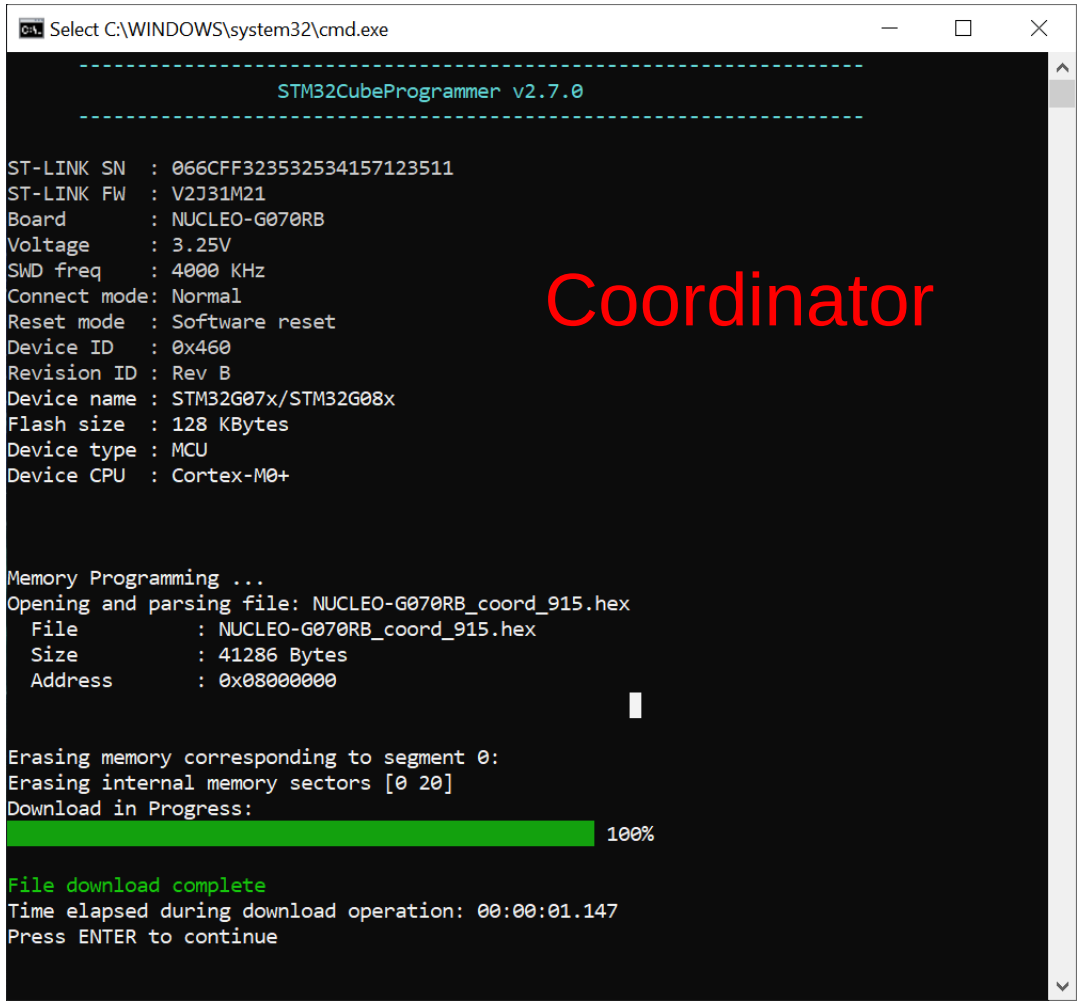
NUCLEO-G070RB FW upgrade (868 MHz)

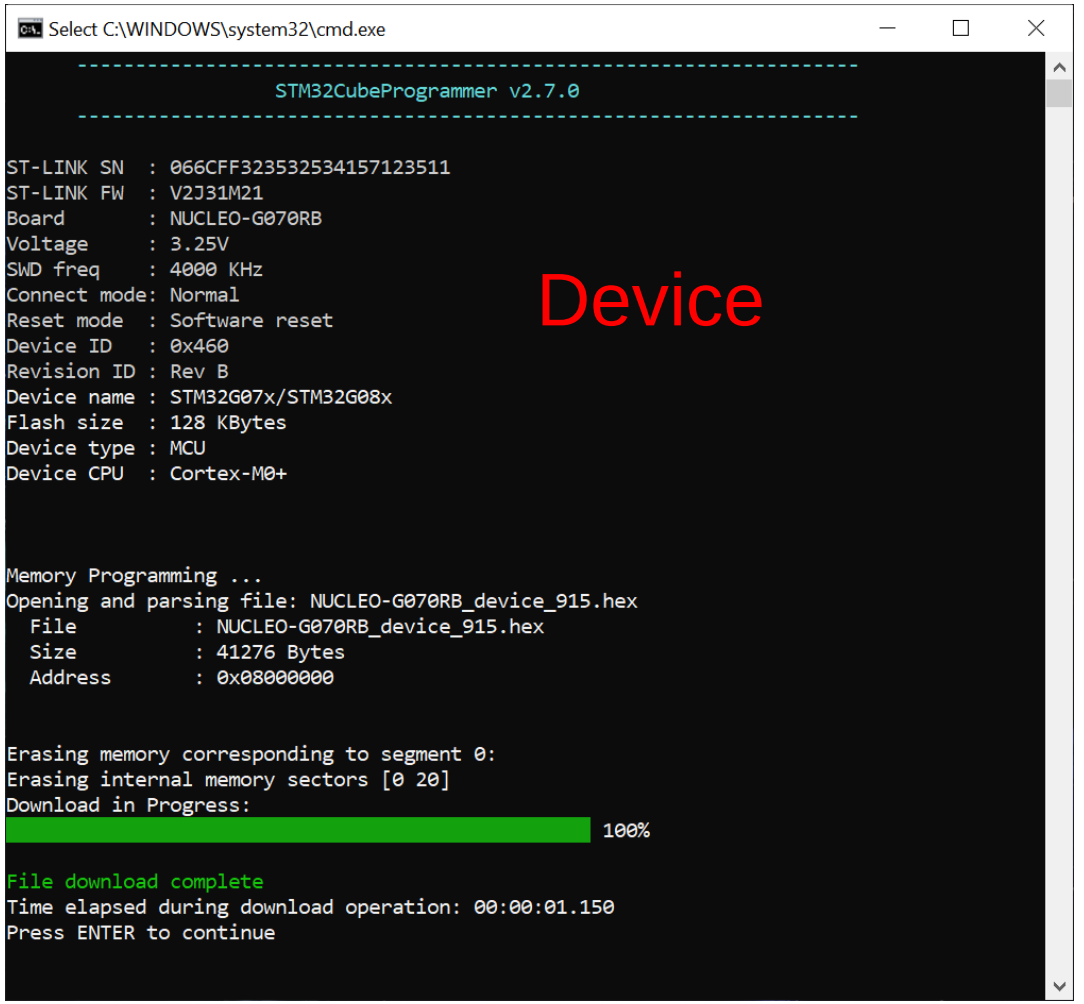




life.augmented

NUCLEO-G070RB FW upgrade (915 MHz)

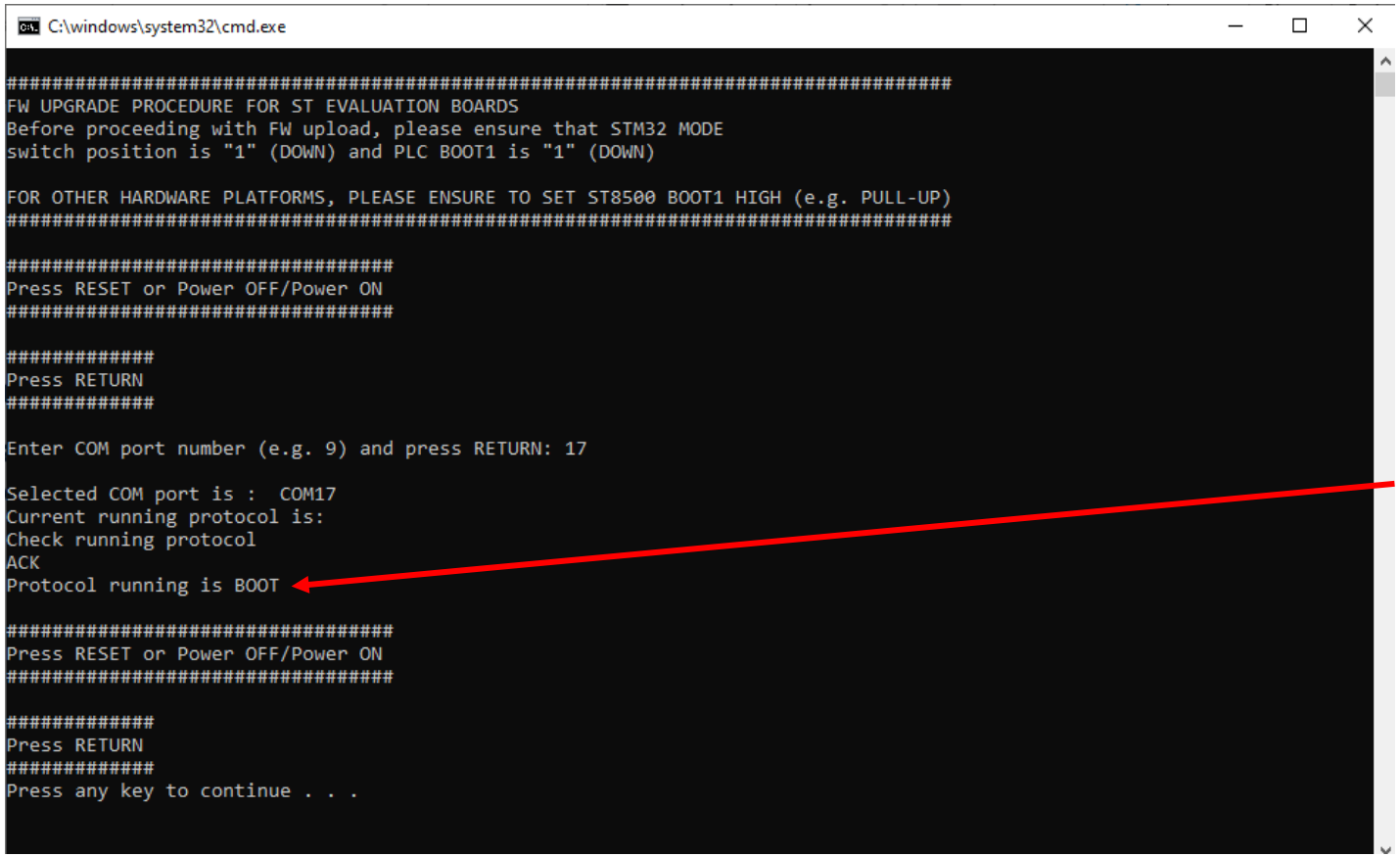




life.augmented

ST8500 FW upgrade steps

- Launch *ST8500_G3_Hybrid_PE_RTE_upload.bat* file and follow the instructions.

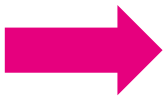


Running protocol is BOOT if ST8500 SPI flash is empty

ST8500 FW upgrade steps

- *ST8500_SPI_Loader.img* file upload on ST8500 RAM.
- After completion, FW images are written into ST8500 SPI flash.

```
Upload on ST8500 RAM a boot software on-going...
ACK
True
ACK
BOOT Version read:True
Loading IMG Header
ACK
Header OK... start sending file
ACK
Done Write 2.8%
ACK
Done Write 5.4%
ACK
Done Write 8.0%
ACK
Done Write 10.6%
ACK
Done Write 13.2%
ACK
Done Write 15.7%
ACK
Done Write 18.3%
ACK
Done Write 20.9%
```

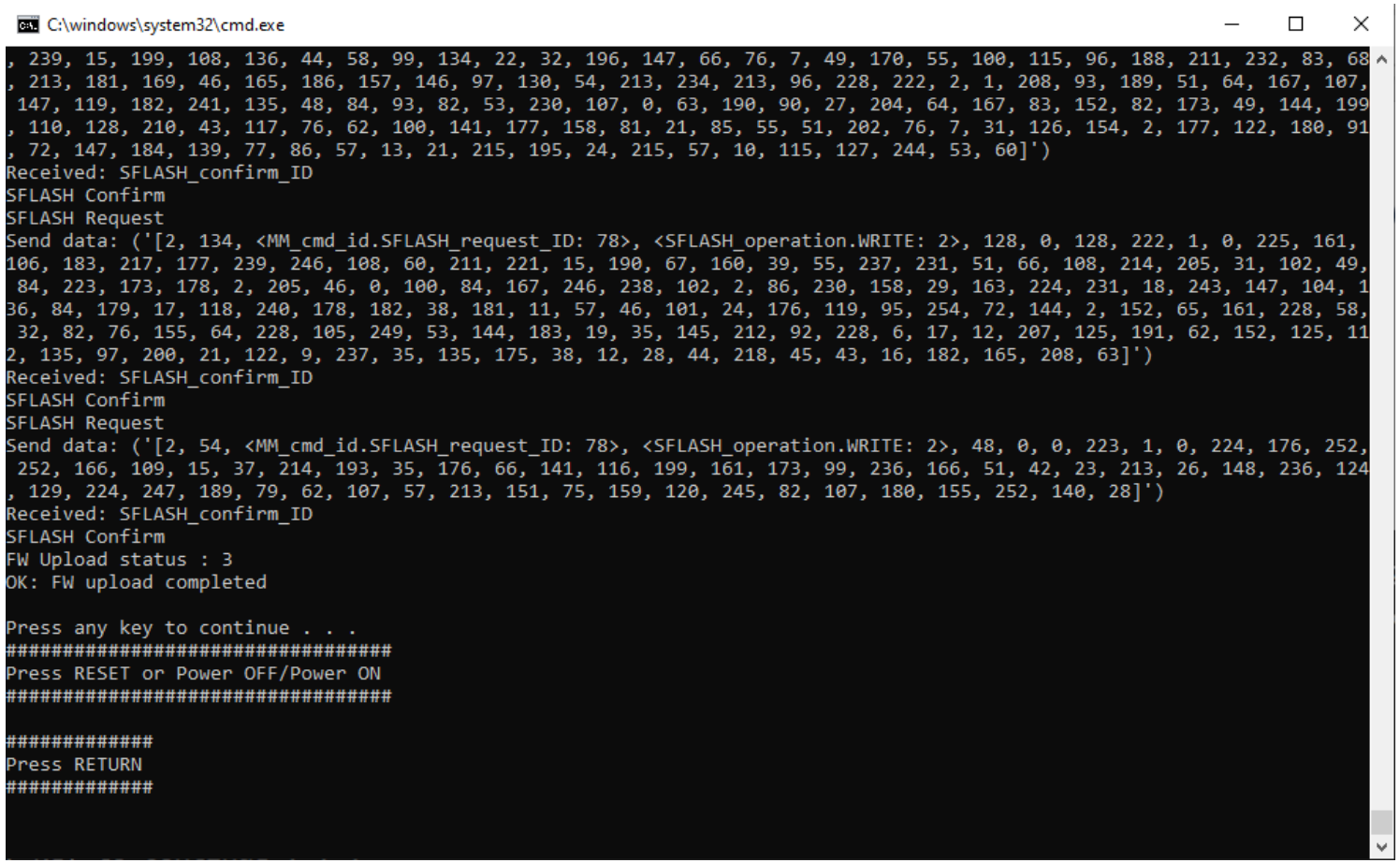


```
Done Write 82.9%
ACK
Done Write 85.5%
ACK
Done Write 88.1%
ACK
Done Write 90.6%
ACK
Done Write 93.2%
ACK
Done Write 95.8%
ACK
Done Write 98.4%
ACK
Done Write 100.0%
PE IMG File Loaded: imgs\ST8500_SPI_Loader.img
Send IMG Start Request
ACK
START DONE
Press any key to continue . . .
```

life.augmented

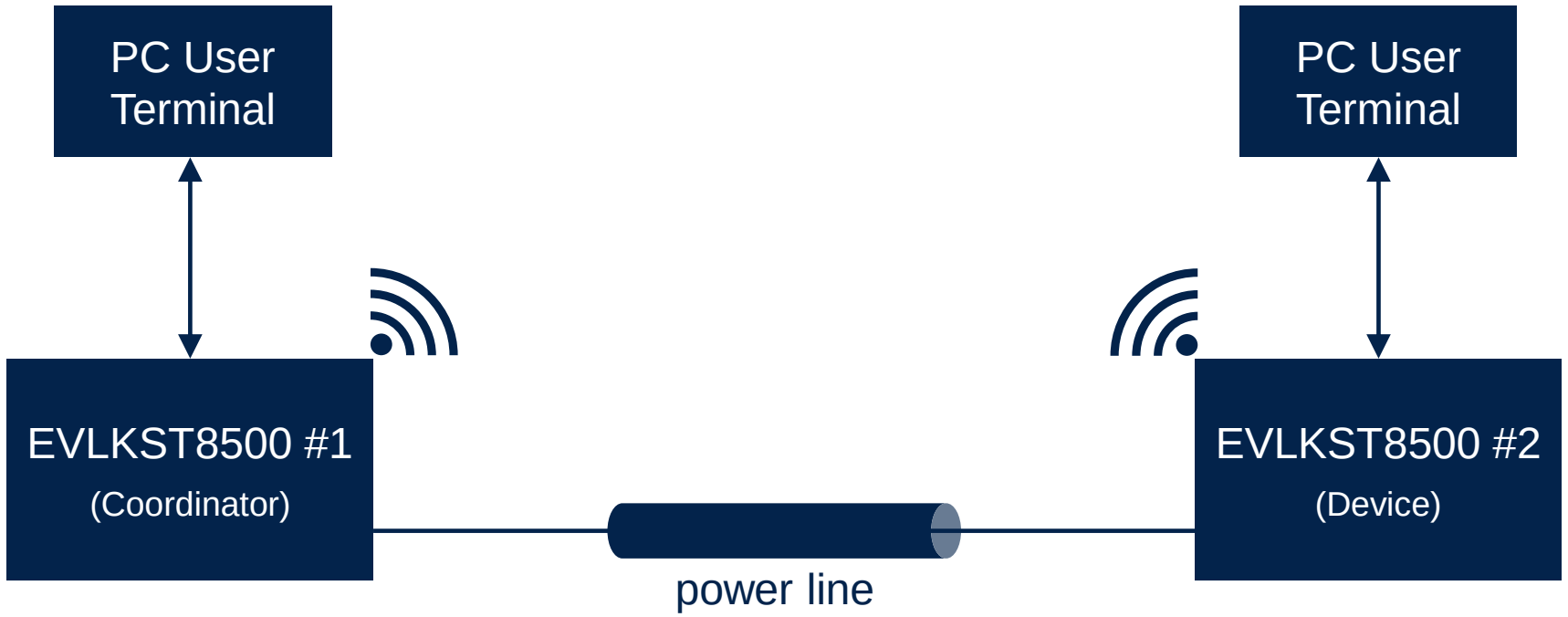
ST8500 FW upgrade steps

- Write of FW images into ST8500 SPI flash (final part and program termination).
- After RESET, the ST8500 is ready.



Demo/test setup

- Basic/standard setup for the user terminal demo/test comes with 2 PLC/RF nodes connected as follows:



General handling

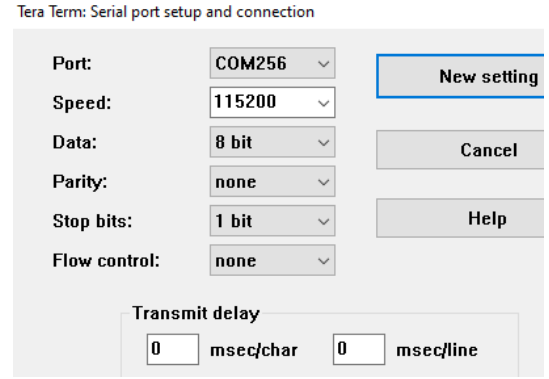
Board configuration

- S1 switch shall be set UP (= STM32 normal mode) on all boards.
- S2 switch shall be set DOWN (= ST8500 boot from SPI flash) on all boards.
- Boards will by default communicate via RF and PLC.
- Each board must be programmed with the proper NUCLEO-G070RB FW type, depending on the RF module. If 868 MHz modules are used, use the 868 MHz version, otherwise, if 915 MHz modules are used, use the 915 MHz version.
- If a 868 MHz module has to communicate with a 915 MHz module, a NUCLEO-G070RB FW modification is necessary to align the RF frequency (*rfconfig_req.data.RadioBaseFreq*).

PC Terminal configuration

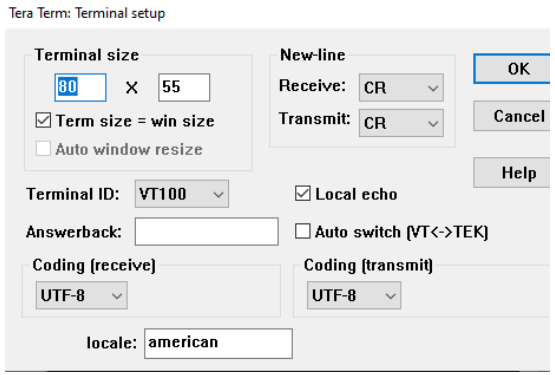
- Open one serial terminal for each board PC side.
- Serial terminals port configuration:
  - Port: *<COM nb>*
  - Speed: *115200*
  - Data: *8bit*
  - Parity: *none*
  - Stop bits: *1 bit*
  - Flow control: *none*

Using Teraterm: menu “Setup -> Serial port…”



- Terminal setup:
  - New line: *Receive: CR, Transmit: CR*
  - Coding: *Receive: UTF-8, Transmit: UTF-8*
  - For convenience “Local echo” can be activated (terminal live prints what is typed)

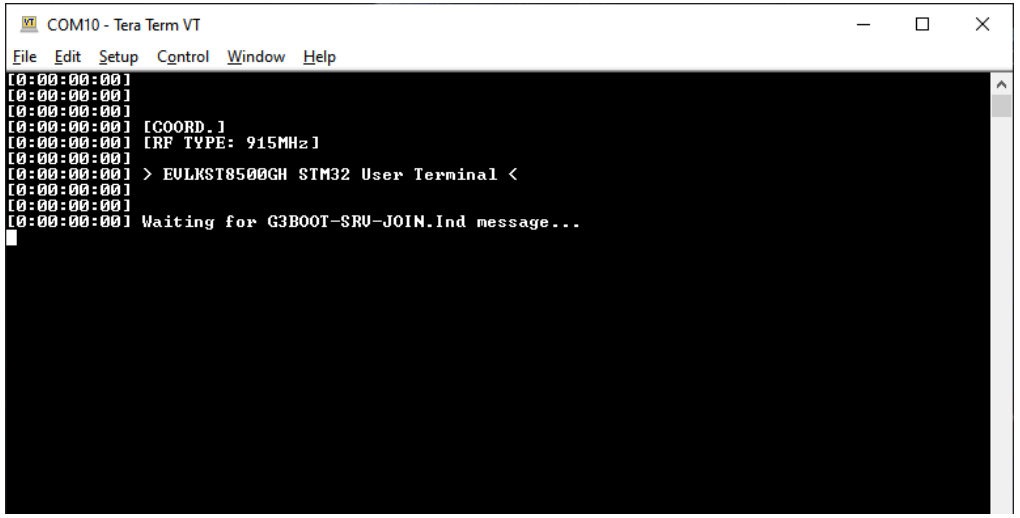
Using Teraterm: menu “Setup -> Terminal…”

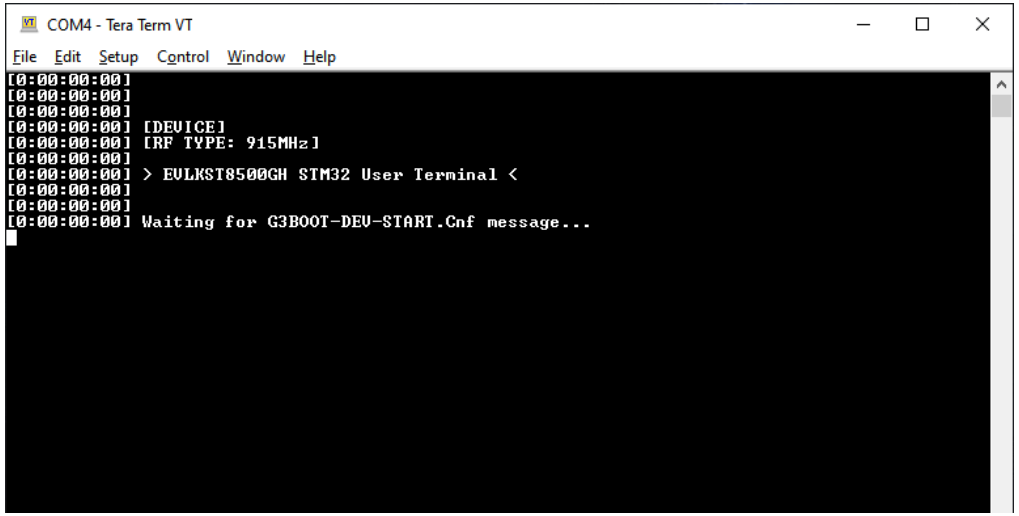


life.augmented

Terminal welcome menu

- Press the NUCLEO board reset button on all boards: the EVLKST8500GH STM32 Terminal welcome prompt appears.
- First, the user is invited to wait (~1min) for the end of the G3-PLC Bootstrap phase…





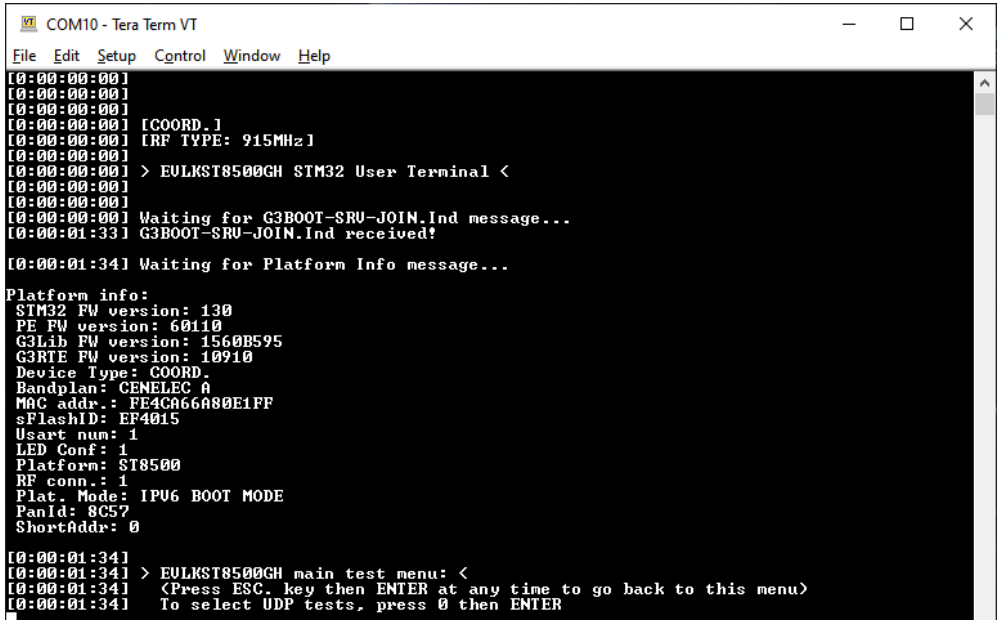
Left side terminal: linked to EVLKST8500 #1 (Coordinator)

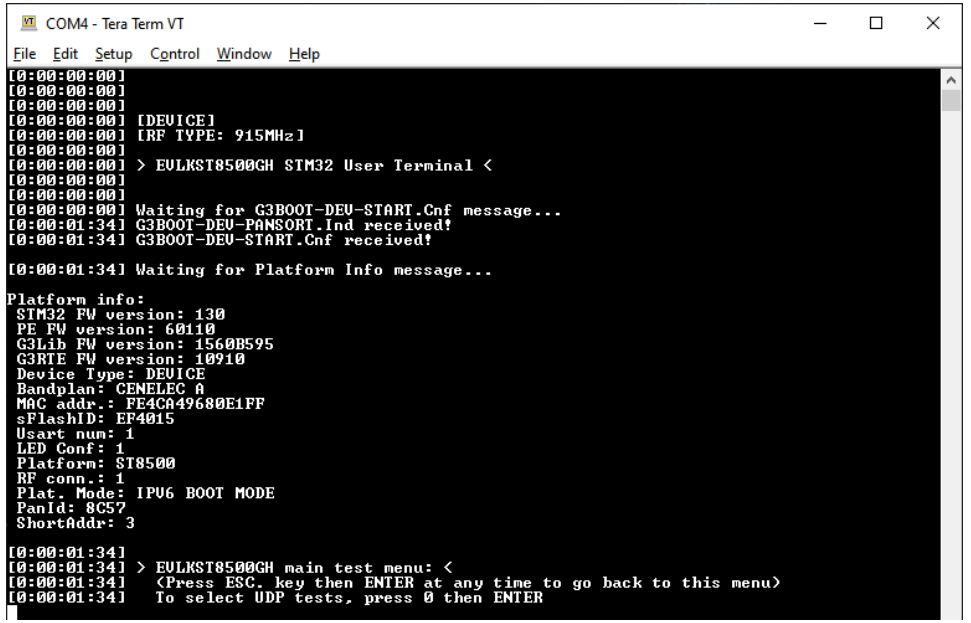
Right side terminal: linked to EVLKST8500 #2 (Device)

life.augmented

Terminal main test menu 1/2

- As soon as the final Bootstrap messages are received (i.e G3BOOT-SRV-JOIN.Ind on Coord. side and G3BOOT-DEV-START.Cnf on Device side), access to the terminal test menu is given:





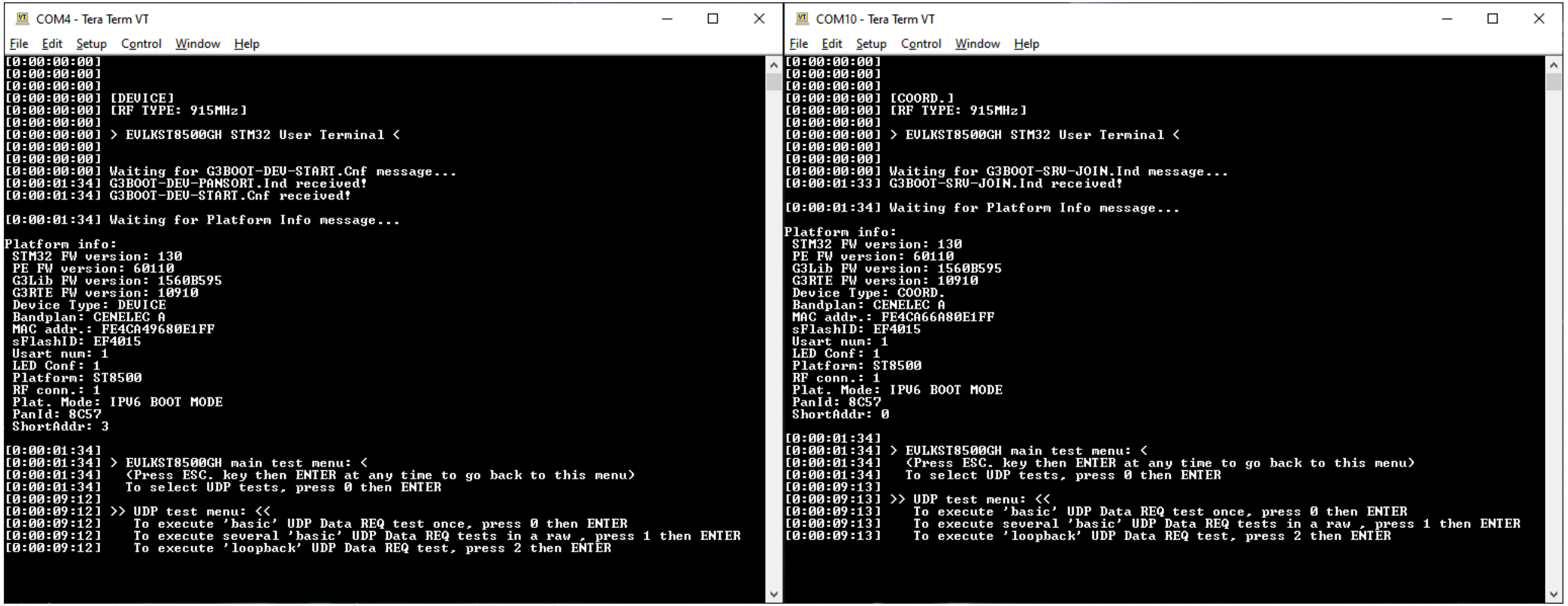
- The user is invited to access further sub-menu and choose a test by typing the corresponding number and then pressing ENTER.

life.augmented

Terminal main test menu 2/2

- Tip: to run a specific test, the corresponding number must be selected on each board terminal through the menu hierarchy.

For example: to reach “UDP test menu”, selections through menu are the same (‘0’ then ‘ENTER’) for both terminals.



life.augmented

Terminal UDP data transfer tests: ‘basic’ UDP Data Req test

‘Basic’ UDP Data Req test launching

- This test aims at making a single UDP data transfer from one node to another.
- The test can be selected from terminals as follows:
  - From *EVLKST8500GH test menu*: press **0** to select *UDP tests.*
  - Then, from *UDP test menu*: press **0** to select *‘basic’ UDP Data REQ test.*

- In the test menu, the user is then invited to choose which node will play the role of UDP Data transfer originator and which node will play the role of UDP data transfer recipient.

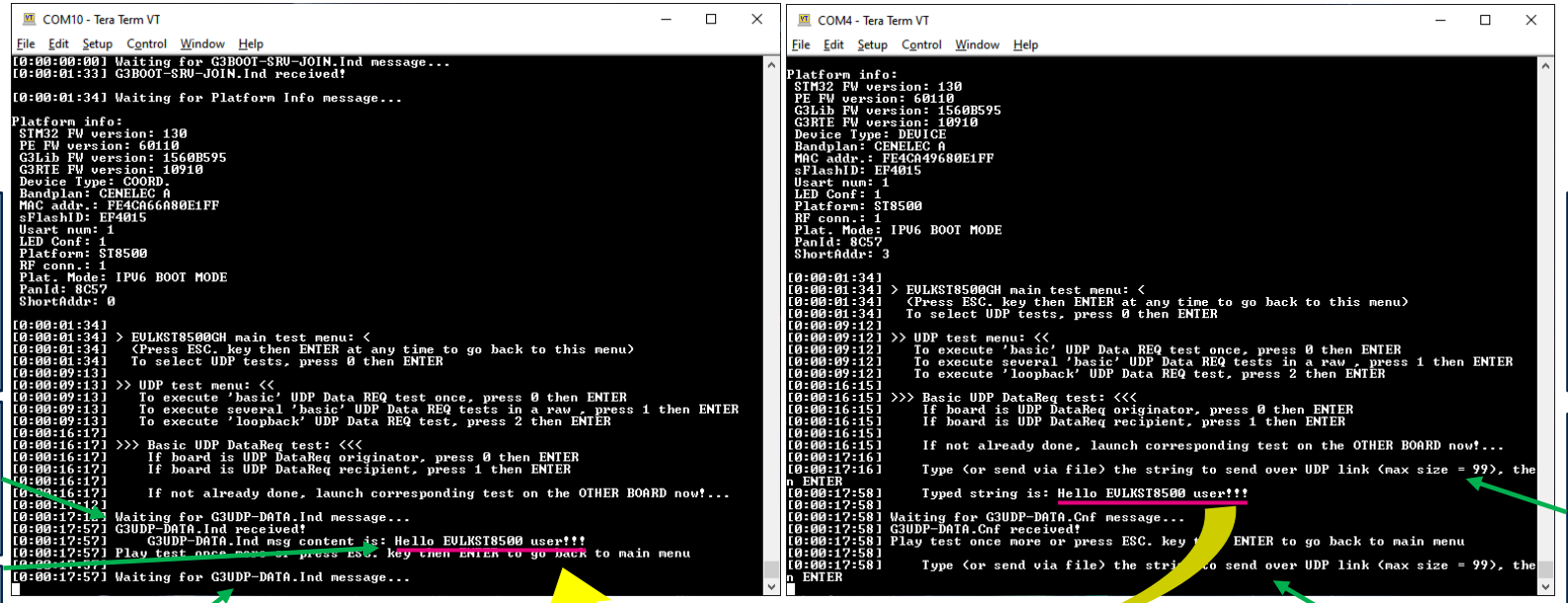
‘Basic’ UDP Data Req test example

‘1’ is selected: EVLKST8500GH #1 is UDP data transfer ***recipient***

Then EVLKST8500GH #1 waits for message reception…

Message is received and displayed

Test is finished, system waits for another test string



‘0’ is selected: EVLKST8500GH #2 is UDP data transfer ***originator***

Then, the user is invited to enter the string to be sent (string must be ended by pressing ENTER)

Test is finished, system prompts for another test string

life.augmented

Terminal UDP data transfer tests: several ‘basic’ UDP Data Req test

life.augmented

Several ‘basic’ UDP Data Req test launching

- This test aims at making a series of basic UDP data transfer from one node to another.
- The test can be selected from terminals as follows:
  - From *EVLKST8500GH test menu*: press **0** to select *UDP tests.*
  - Then, from *UDP test menu*: press **1** to select *several ‘basic’ UDP Data REQ test.*

- In the test menu, the user is then invited to choose which node will play the role of UDP Data transfer originator and which node will play the role of UDP data transfer recipient.

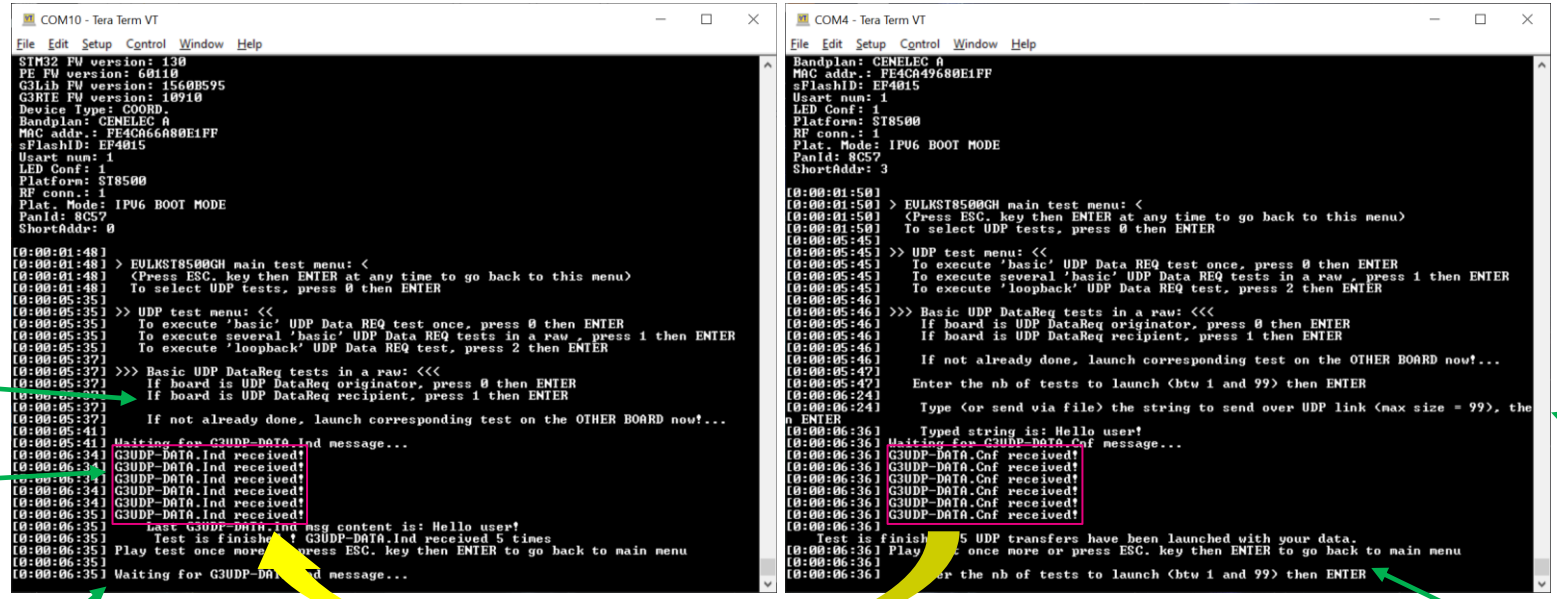
Several ‘Basic’ UDP Data Req test example

‘1’ is selected: EVLKST8500GH #1 is UDP data transfer ***recipient***

Then EVLKST8500GH #1 waits for message reception…

Messages are received and displayed

Test is finished, system waits for another test string



‘0’ is selected: EVLKST8500GH #2 is UDP data transfer ***originator***

Then, the user is invited to enter the string to be sent (string must be ended by pressing ENTER)

Test is finished, system prompts for another test string

life.augmented

Terminal UDP data transfer tests: ‘loopback’ UDP Data Req test

‘Loopback’ UDP Data Req test launching

- This test aims at making a UDP data transfer from one node to another, then data are sent back to the originator. The data loopback transfer duration and number of exchanged bytes are reported.
- The test can be selected from terminals as follows:
  - From *EVLKST8500GH test menu*: press **0** to select *UDP tests.*
  - Then, from *UDP test menu*: press **2** to select *‘loopback’ UDP Data REQ test.*
- In the test menu, the user is then invited to choose which node will play the role of UDP Data transfer originator and which node will play the role of UDP data transfer mirror.

‘Loopback’ UDP Data Req test example

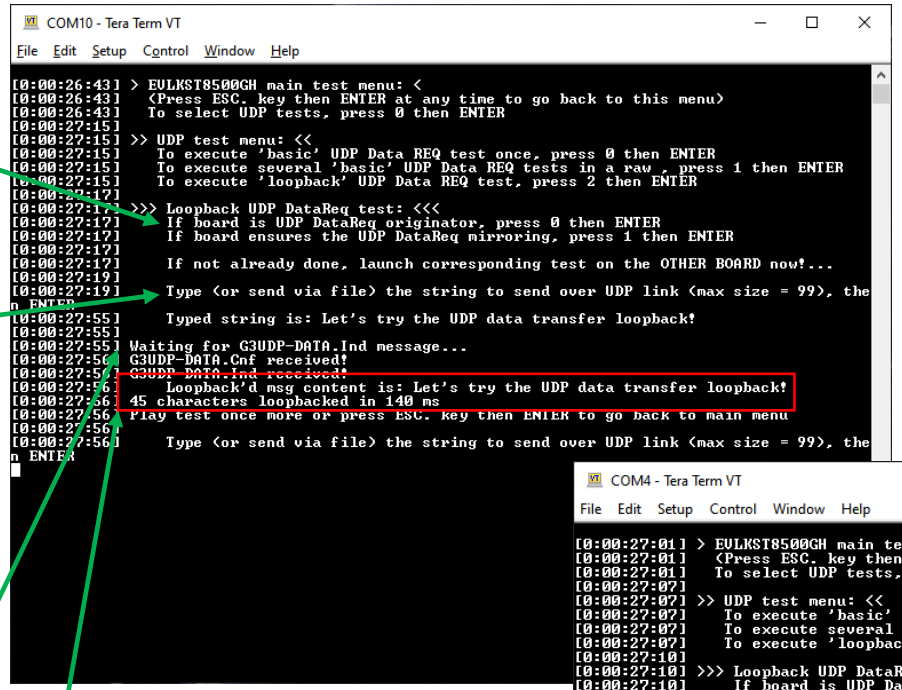
‘0’ is selected: EVLKST8500GH #1 is UDP data transfer ***originator***

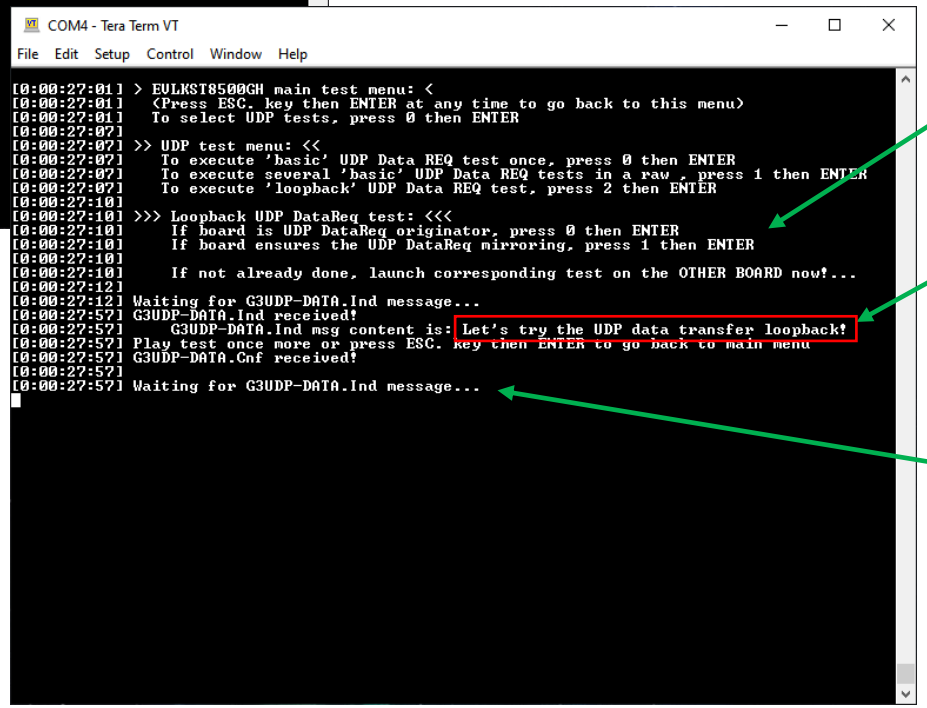
Then, the user is invited to enter string to be sent (string must be ended by pressing ENTER)

Then EVLKST8500GH #1 waits for message loopback…

Message is received back, displaying:

- ***Msg content***
- ***Nb of exchanged chars***
- ***transfer duration***





‘1’ is selected: EVLKST8500GH #2 is UDP data transfer ***“mirror”***

Data from originator is received and sent back

Test is finished, system waits for another test string

life.augmented

Thank you

life.augmented