life.augmented

STM32H5 USB DFU Dual Bank Example

ST Microcontrollers

Overview

- Presenting the STM32H5 USB DFU Dual Bank Example code

Hardware used:

- Nucleo-H563 (using STM32H5 MCUs)
- 2x USB Cables

Software:

- STM32CubeProgrammer (V1.15.0, do not use version V2.16.0 due to a limitation)
- Terminal Software like TeraTerm

Materials delivered:

- STM32H5 USB DFU Dual Bank Project (source and binary)
- STM32H5 Application Example Code (source and binary)
- STM32H5 USB DFU Dual Bank.pdf

Overview

- The Example code demonstrates the bank swapping mechanism when doing firmware upgrade of the application code using USB DFU (Device Firmware Upgrade).
- The code checks if application code is present at address 0x0802 2000, if there is code it will jump to it.
- The code also checks if the user button is pressed after releasing the reset button on the board, if so, it will run the USB DFU code.
- The Example shows how to do a firmware upgrade of the application code on the opposite bank of the active one.
- The example code shows the swap bank mechanism.

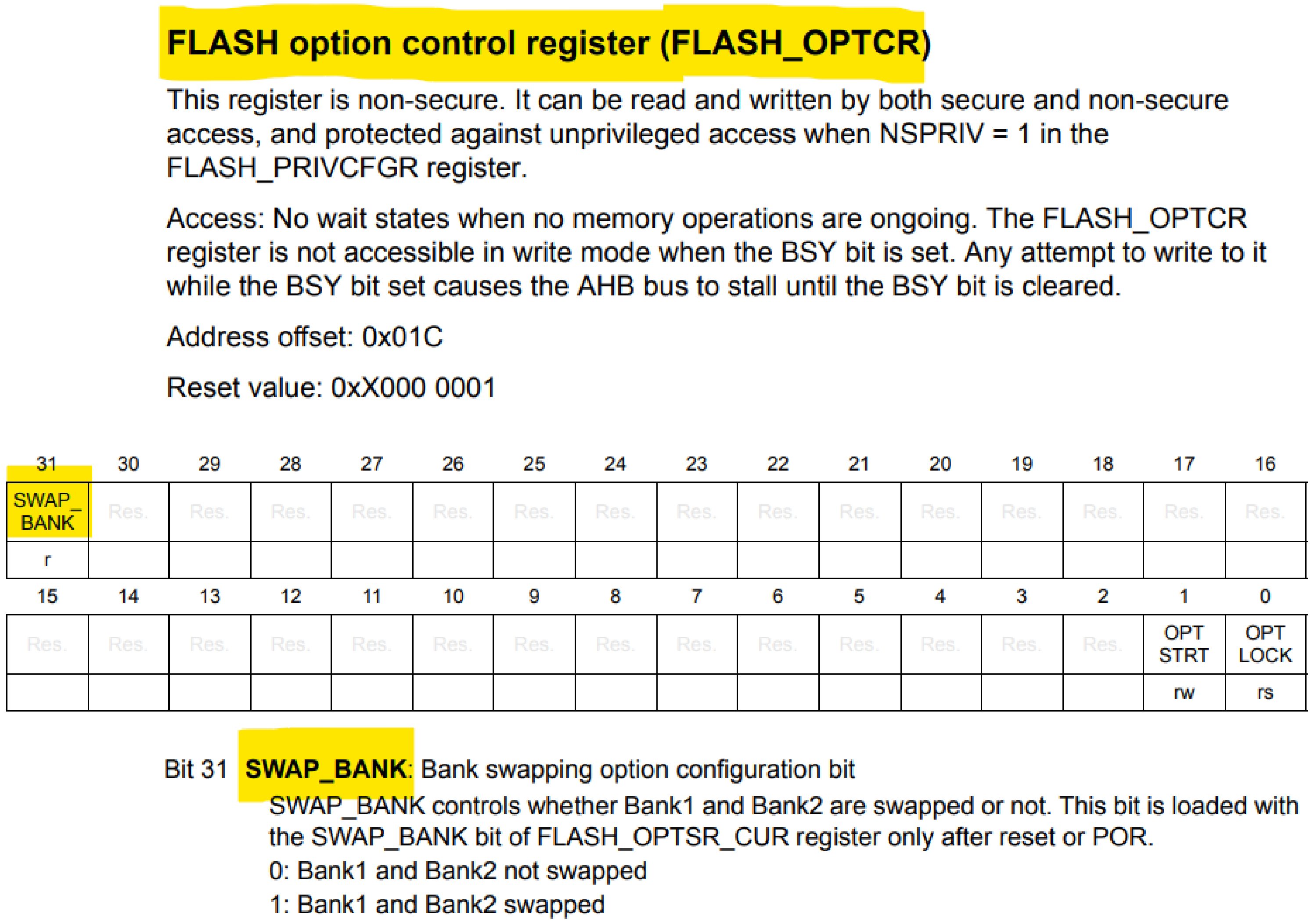
life.augmented

Memory map and swapping options of the STM32H563

Area	Corresponding bank		Start address	End address	Size (bytes)	Region name
	SWAP_BANK = 0	SWAP_BANK = 1				
User main memory	Bank1	Bank2	0x0800 0000	0x0800 1FFF	8 K	Sector 0
			0x0800 2000	0x0800 3FFF	8 K	Sector 1
			...	...	...	...
			0x080F E000	0x080F FFFF	8 K	Sector 127
	Bank2	Bank1	0x0810 0000	0x0810 1FFF	8 K	Sector 0
			0x0810 2000	0x0810 3FFF	8 K	Sector 1
			...	...	...	...
			0x081F E000	0x081F FFFF	8 K	Sector 127

life.augmented

SWAP_BANK bit in User Option Bytes



FLASH option control register (FLASH_OPTCR)

This register is non-secure. It can be read and written by both secure and non-secure access, and protected against unprivileged access when NSPRIV = 1 in the FLASH_PRIVCFGR register.

Access: No wait states when no memory operations are ongoing. The FLASH_OPTCR register is not accessible in write mode when the BSY bit is set. Any attempt to write to it while the BSY bit set causes the AHB bus to stall until the BSY bit is cleared.

Address offset: 0x01C

Reset value: 0xX000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWAP_ BANK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPT STRT	OPT LOCK
														rw	rs

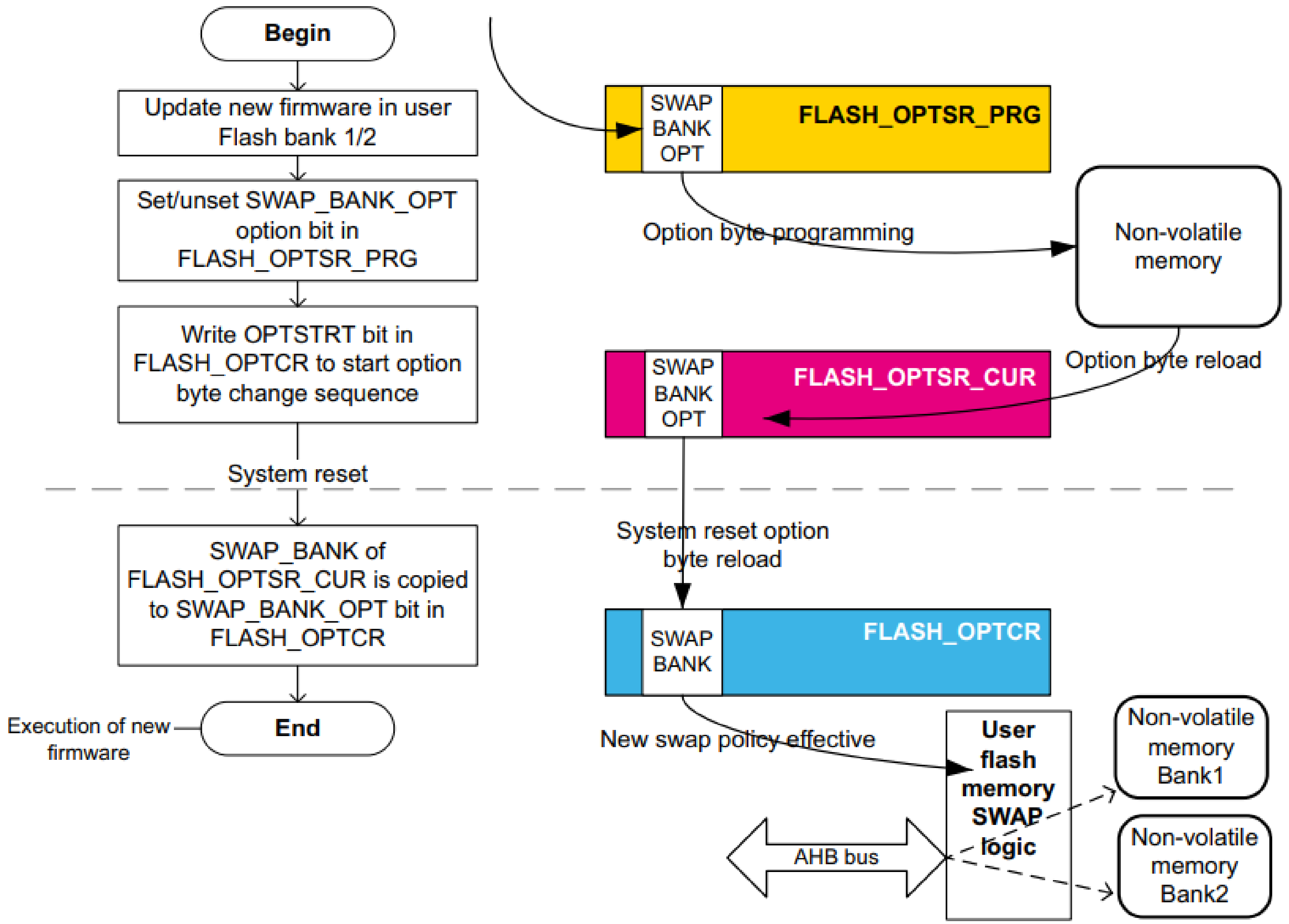
Bit 31 **SWAP_BANK**: Bank swapping option configuration bit

SWAP_BANK controls whether Bank1 and Bank2 are swapped or not. This bit is loaded with the SWAP_BANK bit of FLASH_OPTSR_CUR register only after reset or POR.

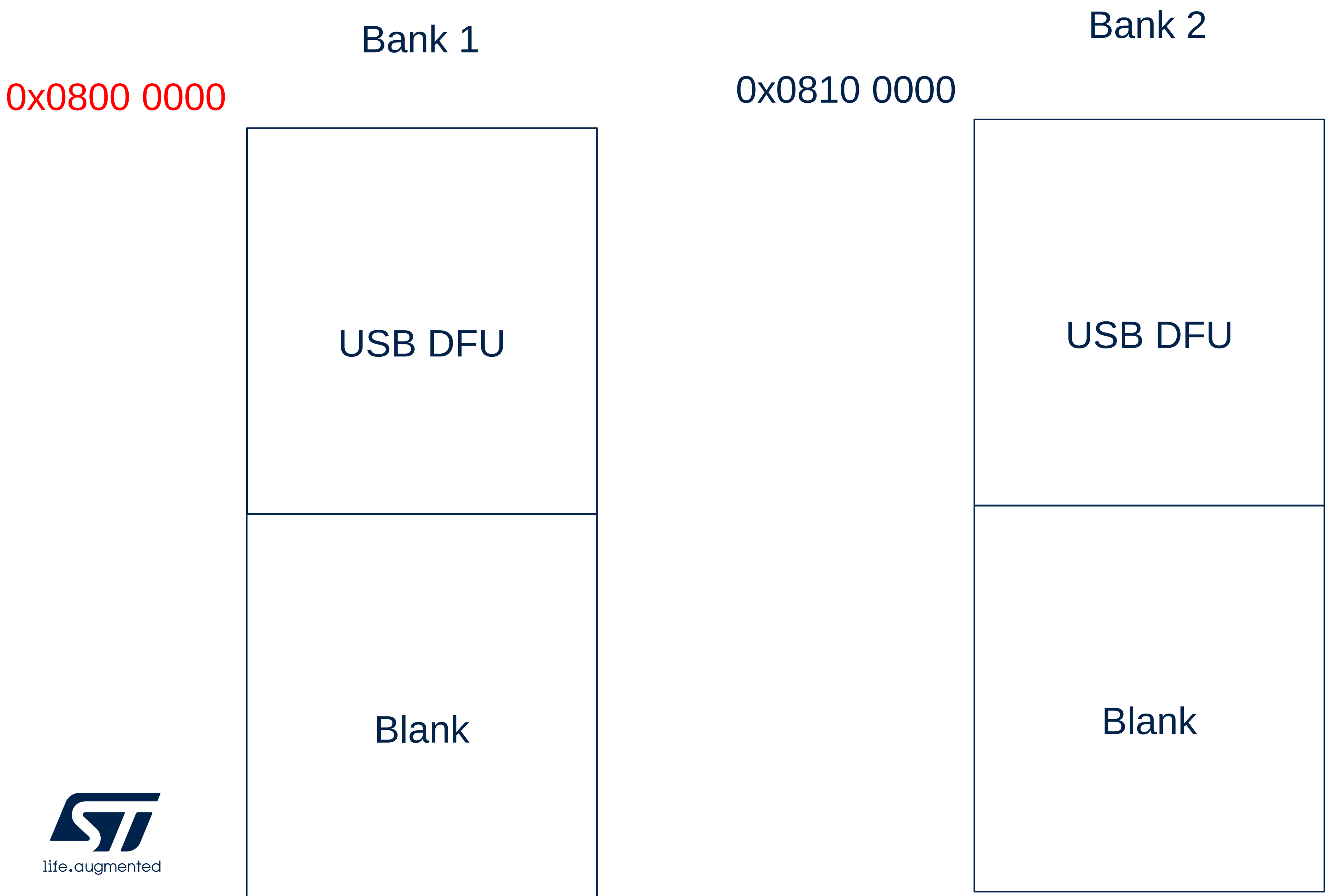
0: Bank1 and Bank2 not swapped

1: Bank1 and Bank2 swapped

Flash bank swapping sequence


Begin
Update new firmware in user Flash bank 1/2
Set/unset SWAP_BANK_OPT option bit in FLASH_OPTSR_PRG
Write OPTSTRT bit in FLASH_OPTCR to start option byte change sequence
System reset
SWAP_BANK of FLASH_OPTSR_CUR is copied to SWAP_BANK_OPT bit in FLASH_OPTCR
End
Execution of new firmware
SWAP BANK OPT
FLASH_OPTSR_PRG
Option byte programming
Non-volatile memory
Option byte reload
SWAP BANK OPT
FLASH_OPTSR_CUR
System reset option byte reload
SWAP BANK
FLASH_OPTCR
New swap policy effective
User flash memory SWAP logic
Non-volatile memory Bank1
Non-volatile memory Bank2
AHB bus

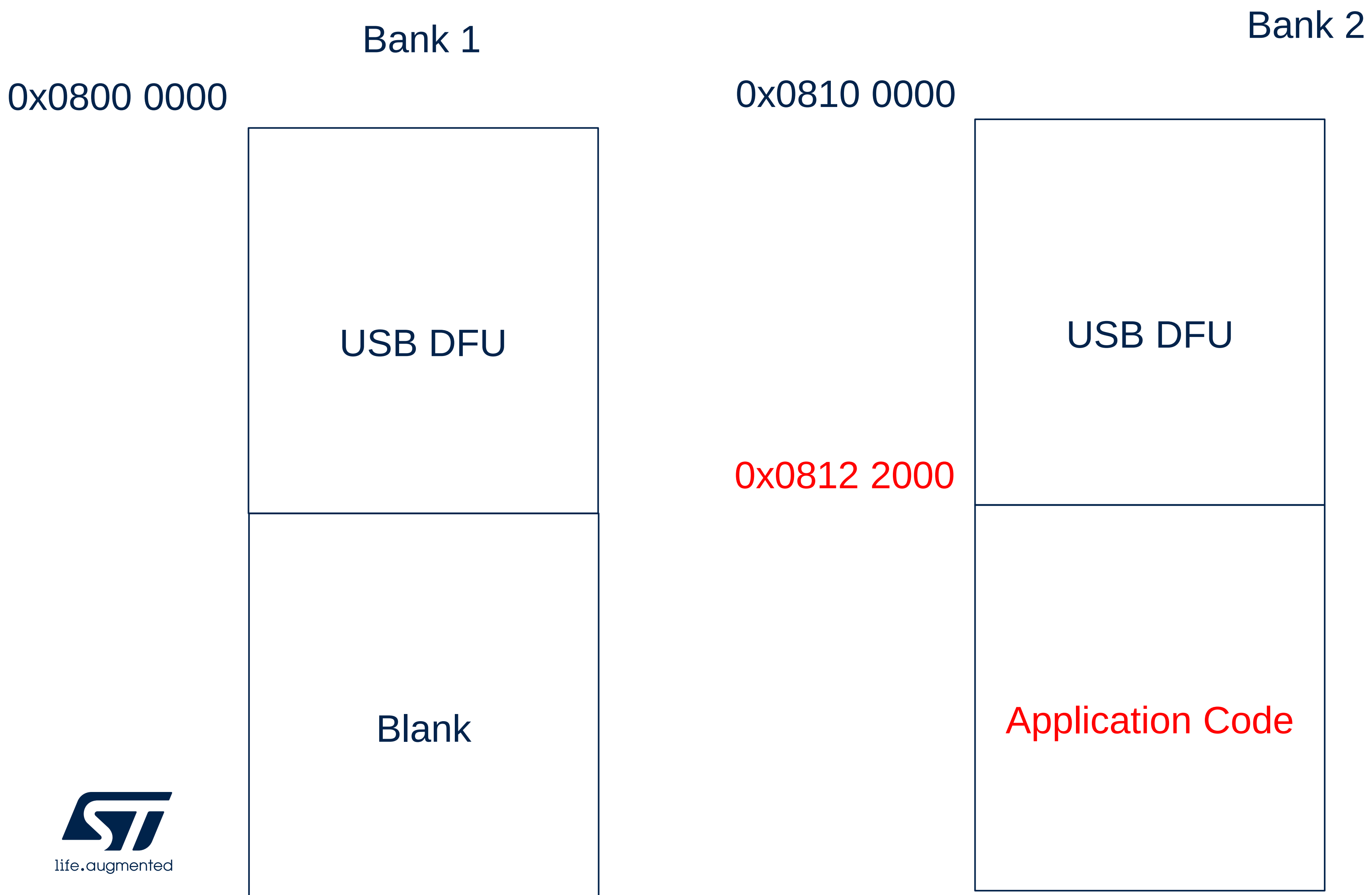
Memory Map – original state – Bank 1 active



- Original state has the USB DFU code programmed at the beginning of each bank.
- Original state has the BANK_SWAP disabled so the active bank is bank 1.

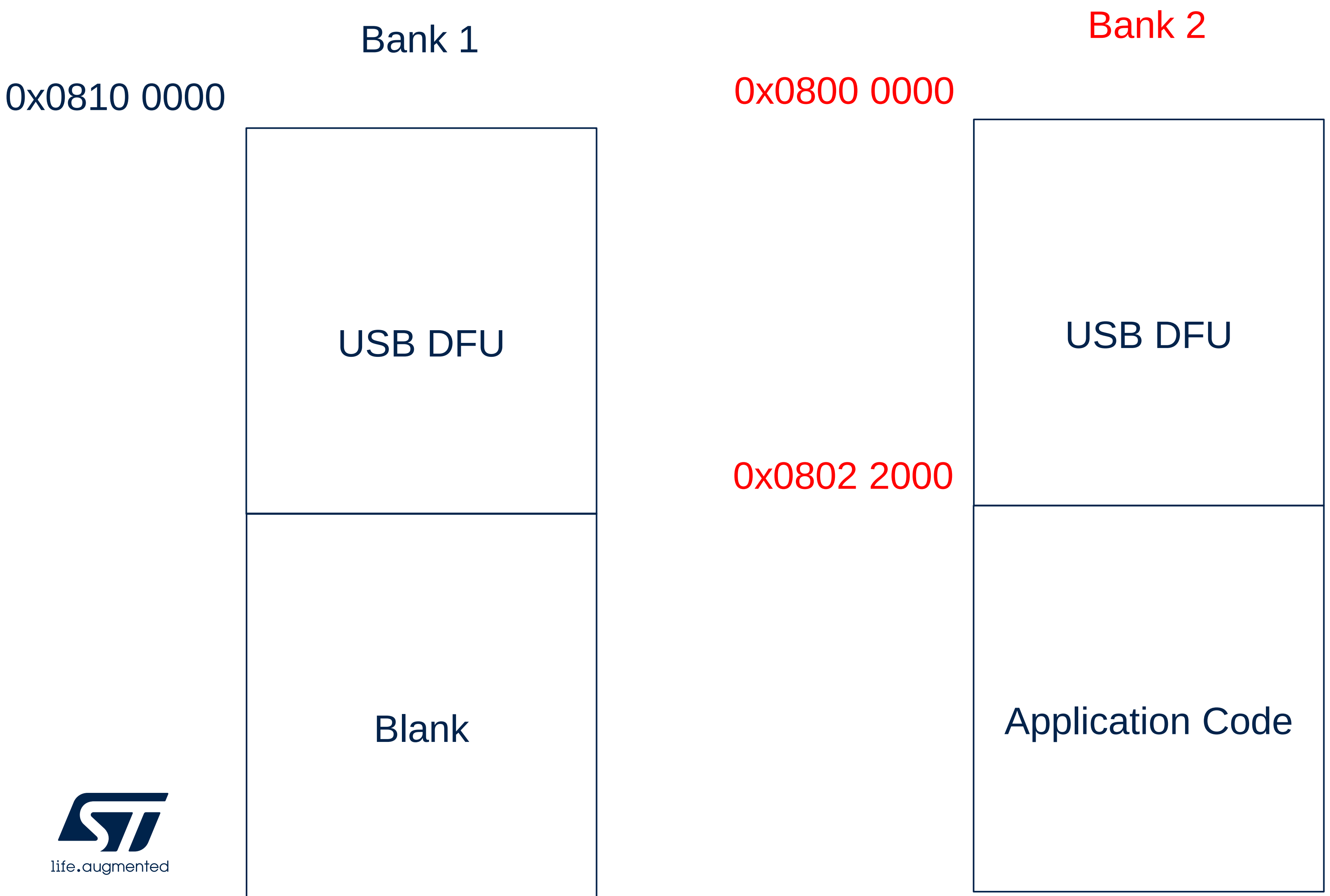
- The application codes in each bank are not present and shown as blank here in the diagrams.

Memory Map – Upgrading code - first time



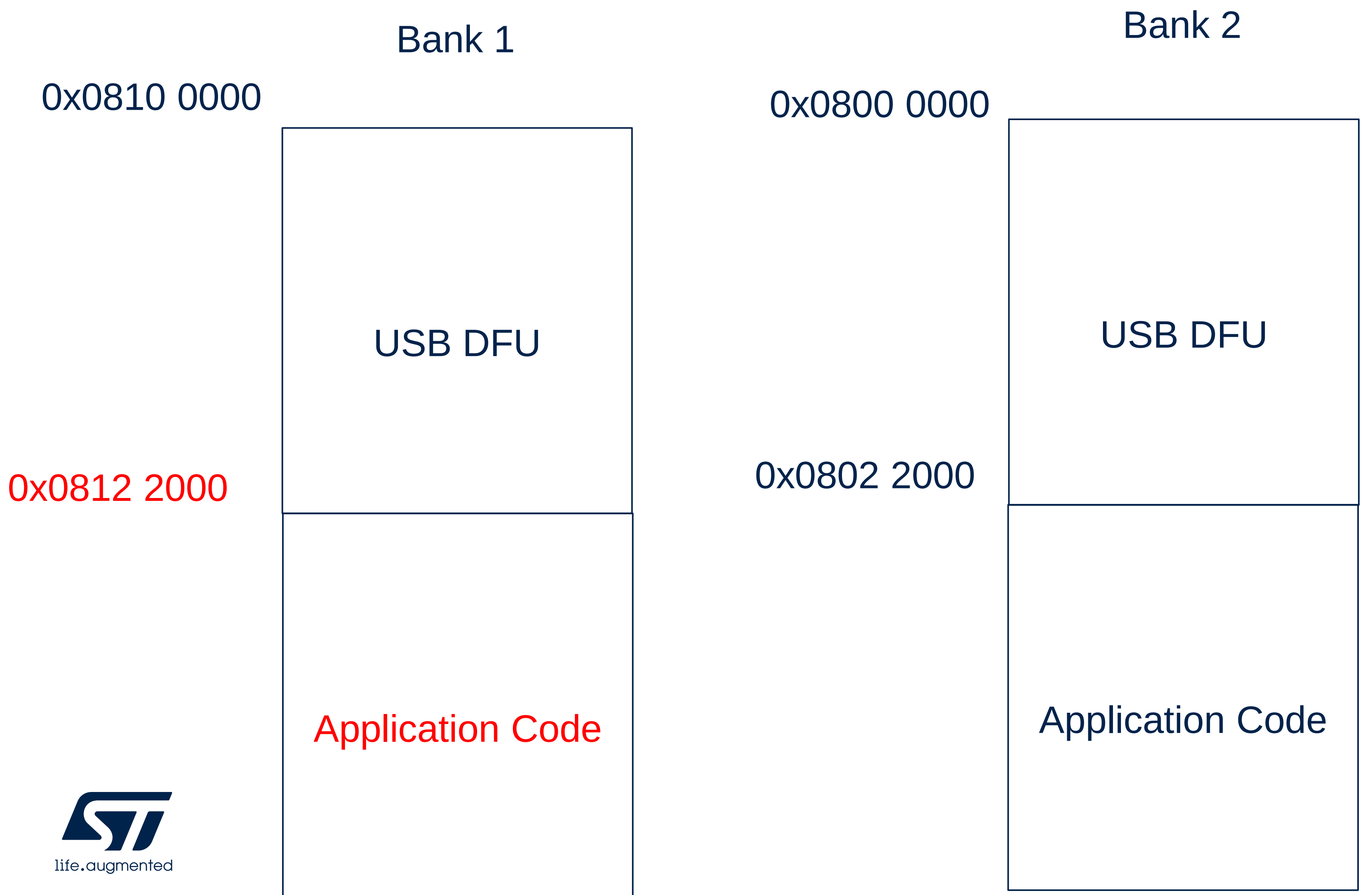
- USB DFU is running from Bank 1 and we are now doing a firmware upgrade of the other bank application code at address 0x0812 2000

Memory Map – Swap bank after updated code – bank 2 active



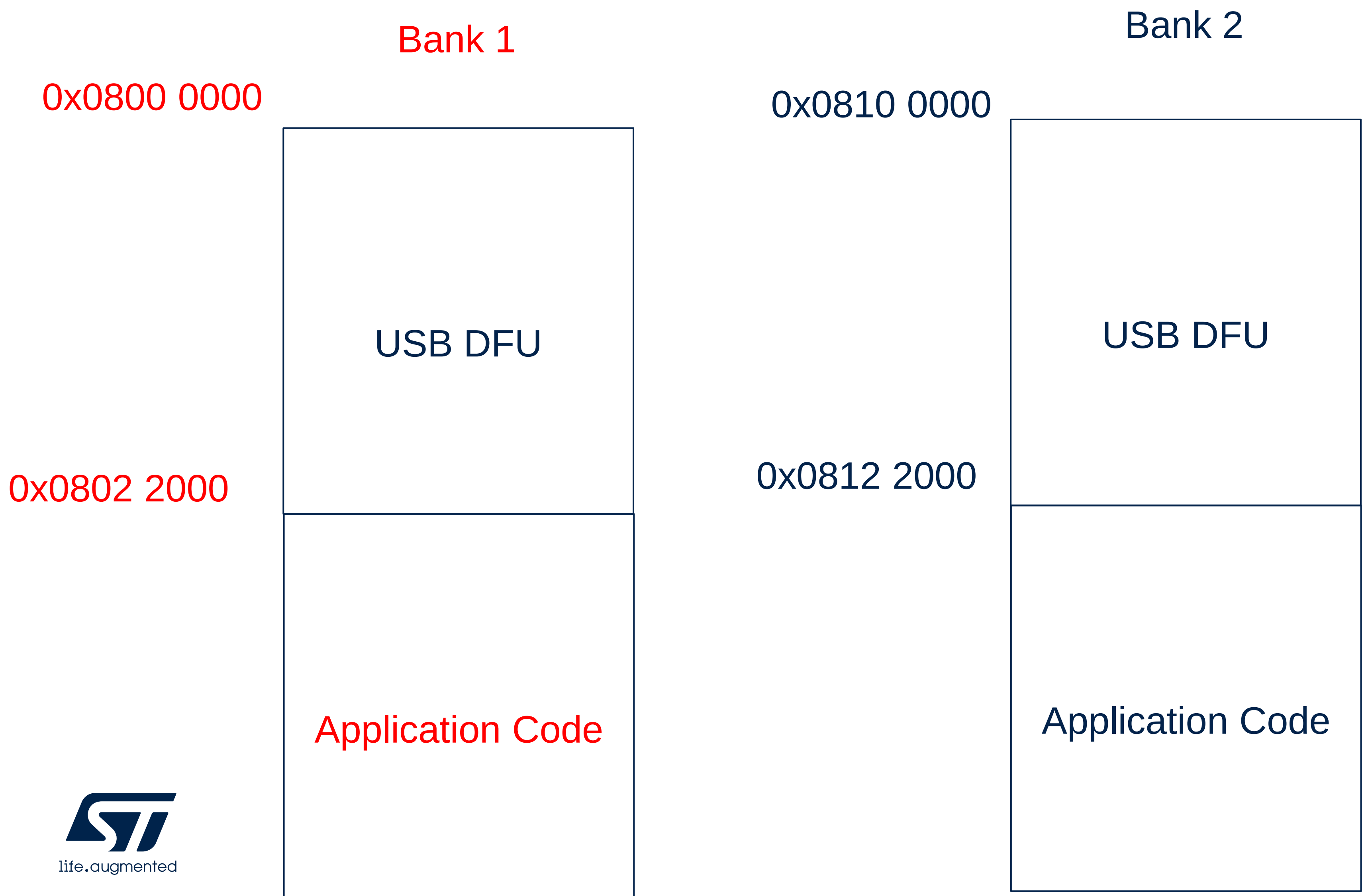
- We are swapping banks from bank 1 to bank 2, so now, bank 2 is the active one.

Memory Map – Upgrading code second time



- When forcing USB DFU the USB DFU is now running from Bank 2
- When performing a firmware upgrade, we are now writing to the bank 1 in application code section (address 0x0812 2000)

Memory Map – Swapping bank – Bank 1 active



- We are swapping banks from bank 2 to bank 1, so now the bank 1 is the active one.

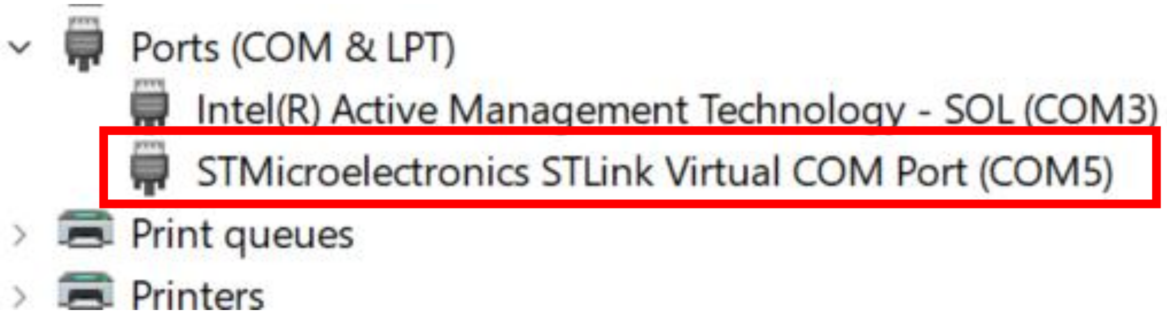
life.augmented

Demo

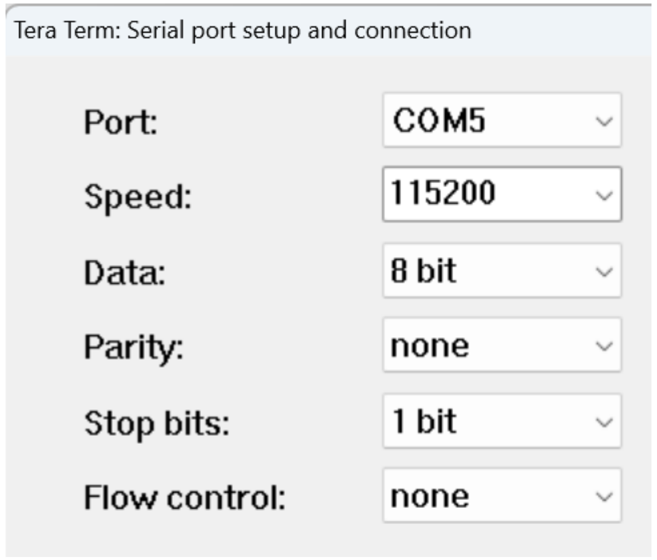
ST Microcontrollers

Running demo – Original State (1/5)

1- With Teraterm (or similar Terminal Software), connect to STLINK Virtual COM port of the Nucleo board (baud rate@115K & no parity) to view printf messages from code.



Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
STMicroelectronics STLink Virtual COM Port (COM5)
Print queues
Printers

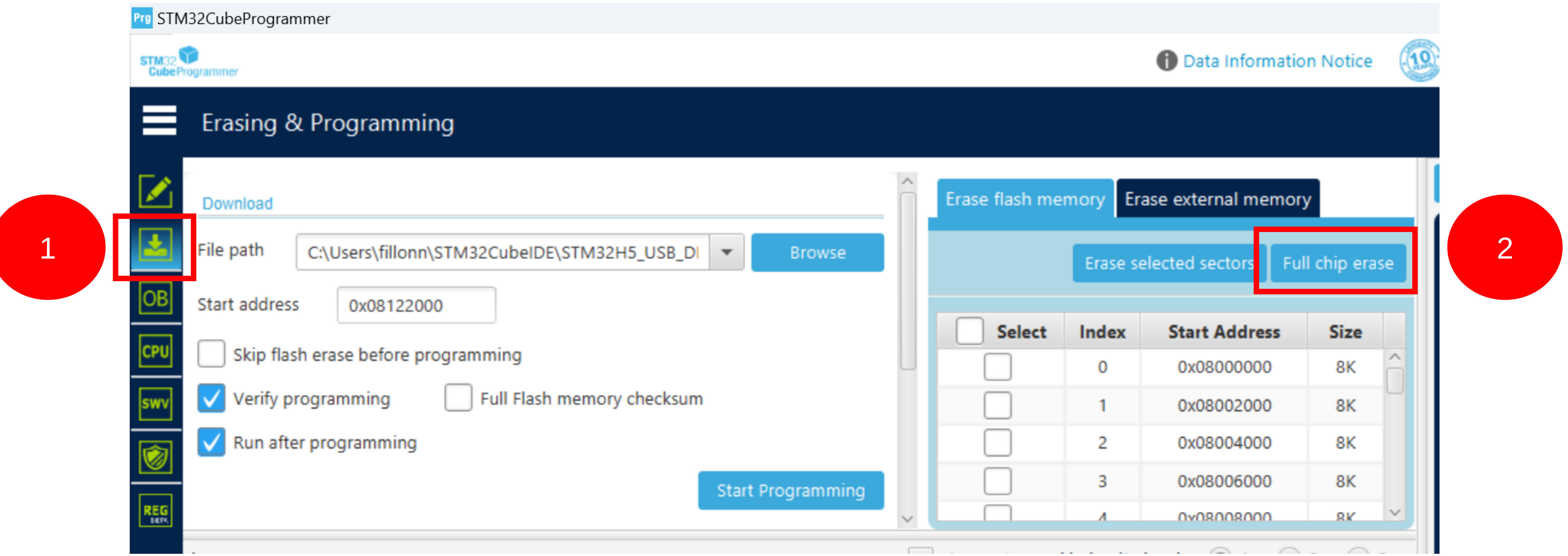


Tera Term: Serial port setup and connection

Port:	COM5
Speed:	115200
Data:	8 bit
Parity:	none
Stop bits:	1 bit
Flow control:	none

Running demo – Original State (2/5)

2- Using STM32CubeProg (with STLINK connection mode) do a full erase of the chip



File path: C:\Users\fillonn\STM32CubeIDE\STM32H5_USB_DI

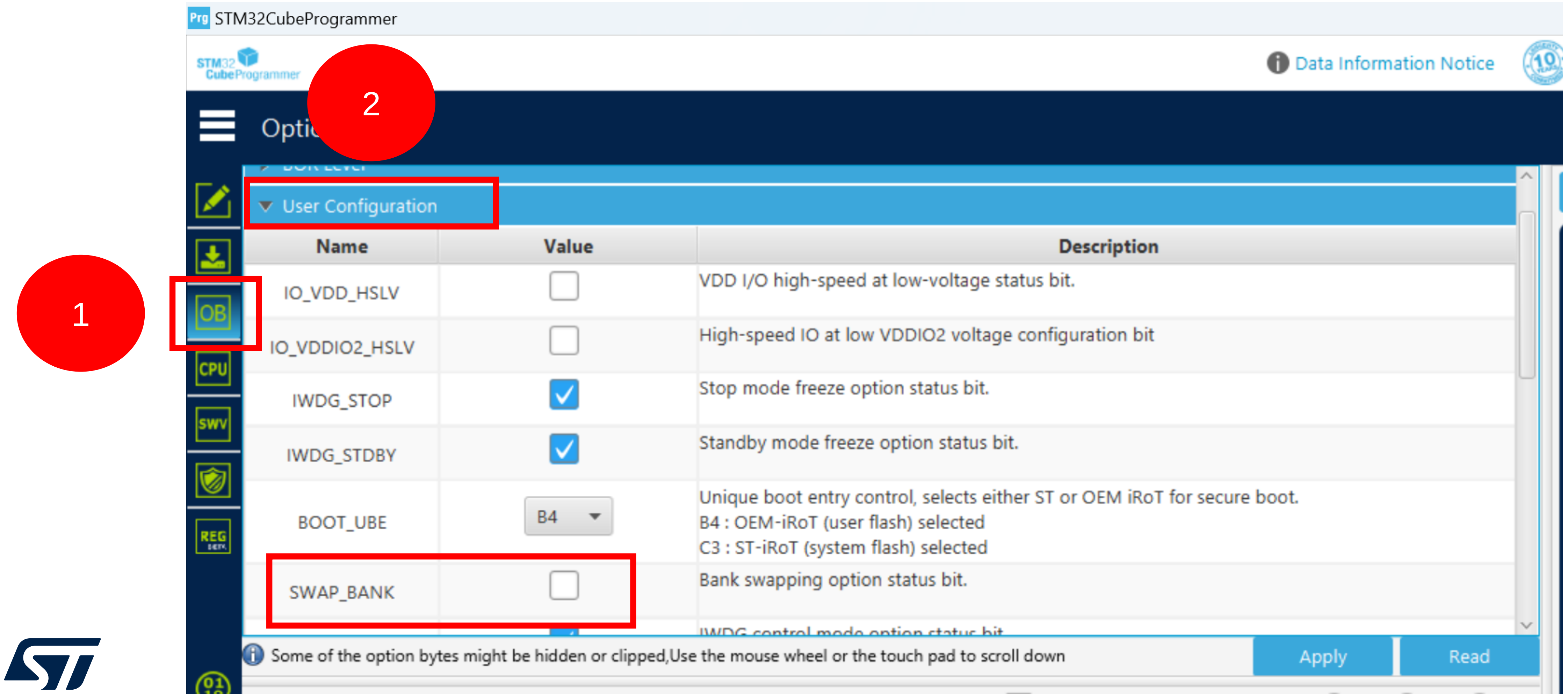
Start address: 0x08122000

- [ ] Skip flash erase before programming
- [x] Verify programming
- [ ] Full Flash memory checksum
- [x] Run after programming

Select	Index	Start Address	Size
☐	0	0x08000000	8K
☐	1	0x08002000	8K
☐	2	0x08004000	8K
☐	3	0x08006000	8K
☐	4	0x08008000	8K

Running demo – Original State (3/5)

3- Using STM32CubeProg (with STLINK connection mode) check that SWAP_BANK bit is reset (not checked) in the User Configuration option Bytes:



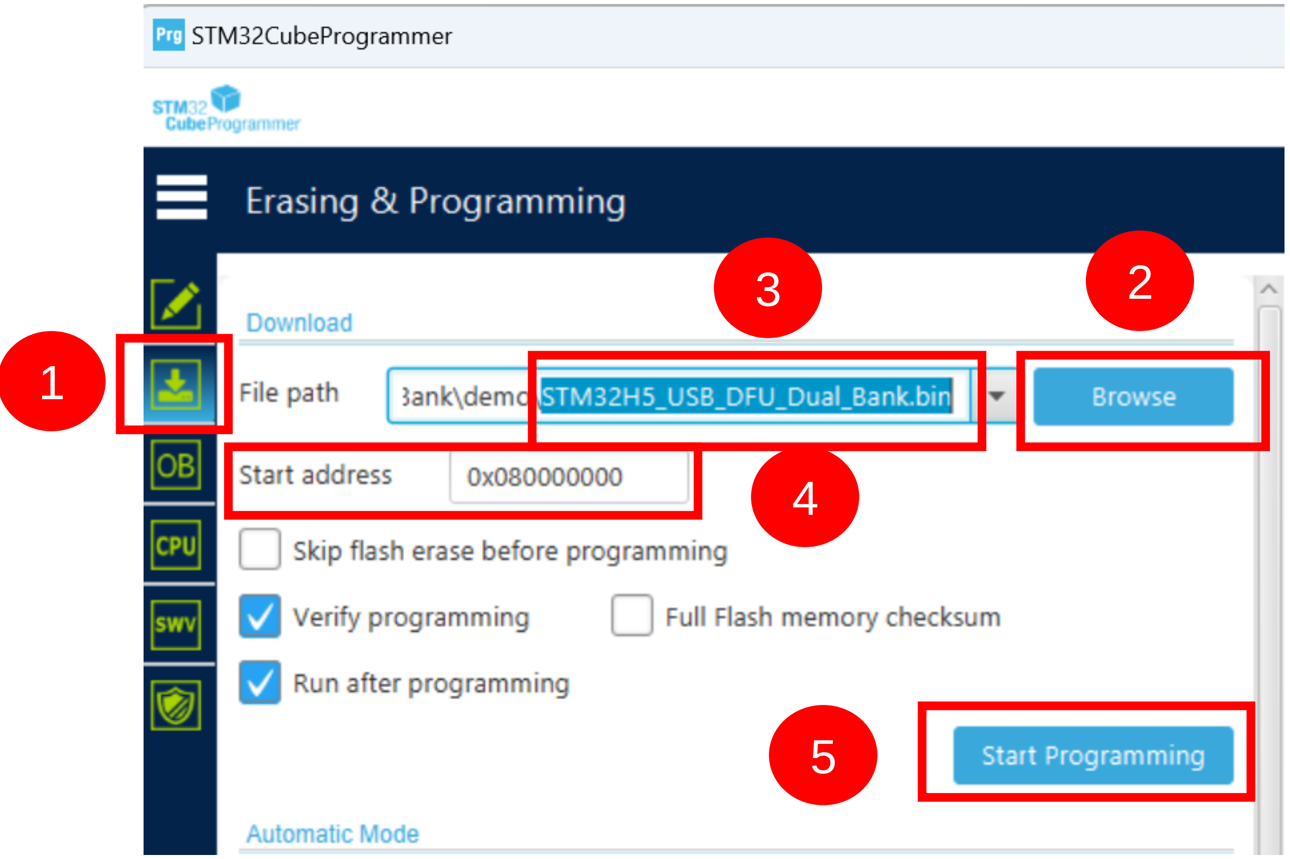
User Configuration

Name	Value	Description
IO_VDD_HSLV	☐	VDD I/O high-speed at low-voltage status bit.
IO_VDDIO2_HSLV	☐	High-speed IO at low VDDIO2 voltage configuration bit
IWDG_STOP	✓	Stop mode freeze option status bit.
IWDG_STDBY	✓	Standby mode freeze option status bit.
BOOT_UBE	B4	Unique boot entry control, selects either ST or OEM iRoT for secure boot. B4 : OEM-iRoT (user flash) selected C3 : ST-iRoT (system flash) selected
SWAP_BANK	☐	Bank swapping option status bit.

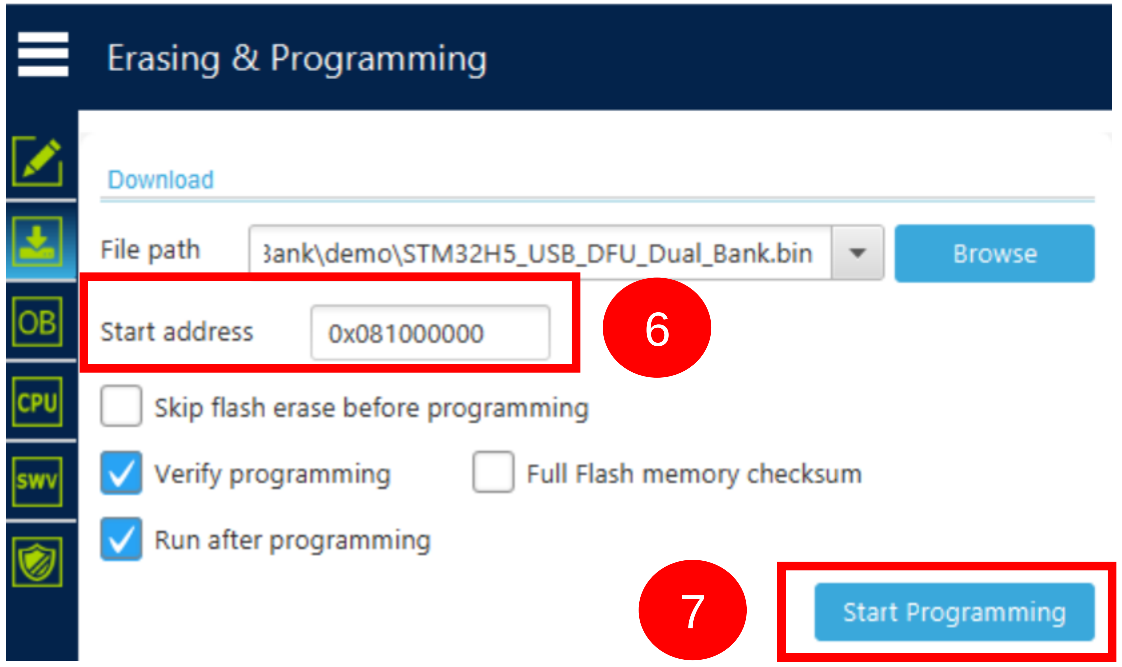
Some of the option bytes might be hidden or clipped,Use the mouse wheel or the touch pad to scroll down

Running demo – Original State (4/5)

4- Program STM32H5_USB_DFU_Dual_Bank.bin at address 0x08000000 and 0x08100000 (at beginning of both banks 1 and 2)



File path: 3ank\demo\STM32H5_USB_DFU_Dual_Bank.bin
Start address: 0x080000000
☐ Skip flash erase before programming
☑ Verify programming
☐ Full Flash memory checksum
☑ Run after programming
Start Programming



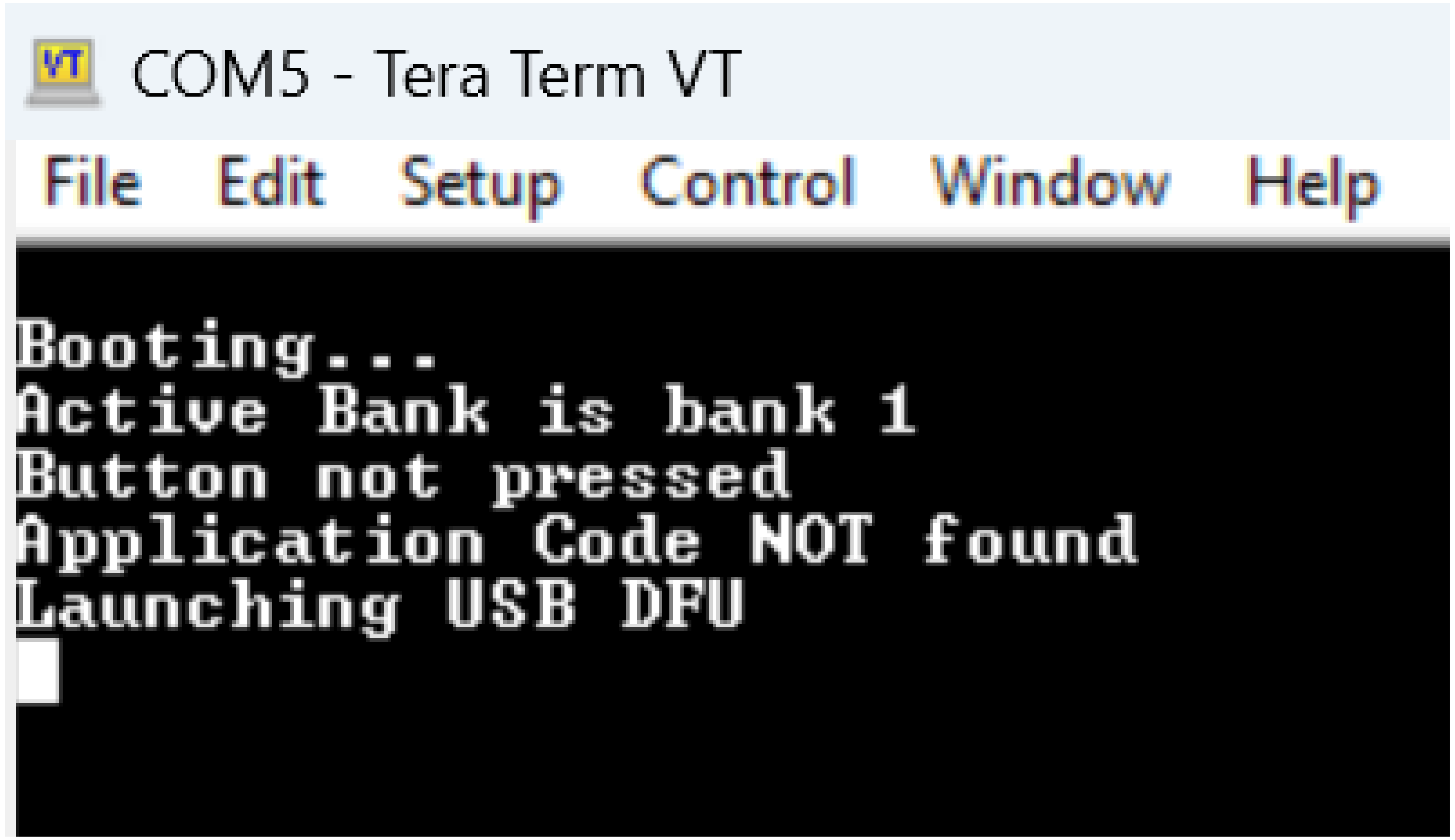
File path: 3ank\demo\STM32H5_USB_DFU_Dual_Bank.bin
Start address: 0x081000000
☐ Skip flash erase before programming
☑ Verify programming
☐ Full Flash memory checksum
☑ Run after programming
Start Programming

life.augmented

Running demo – Original State (5/5)

5- Reset the STM32H5 on the Nucleo board by pressing reset Button (black button)

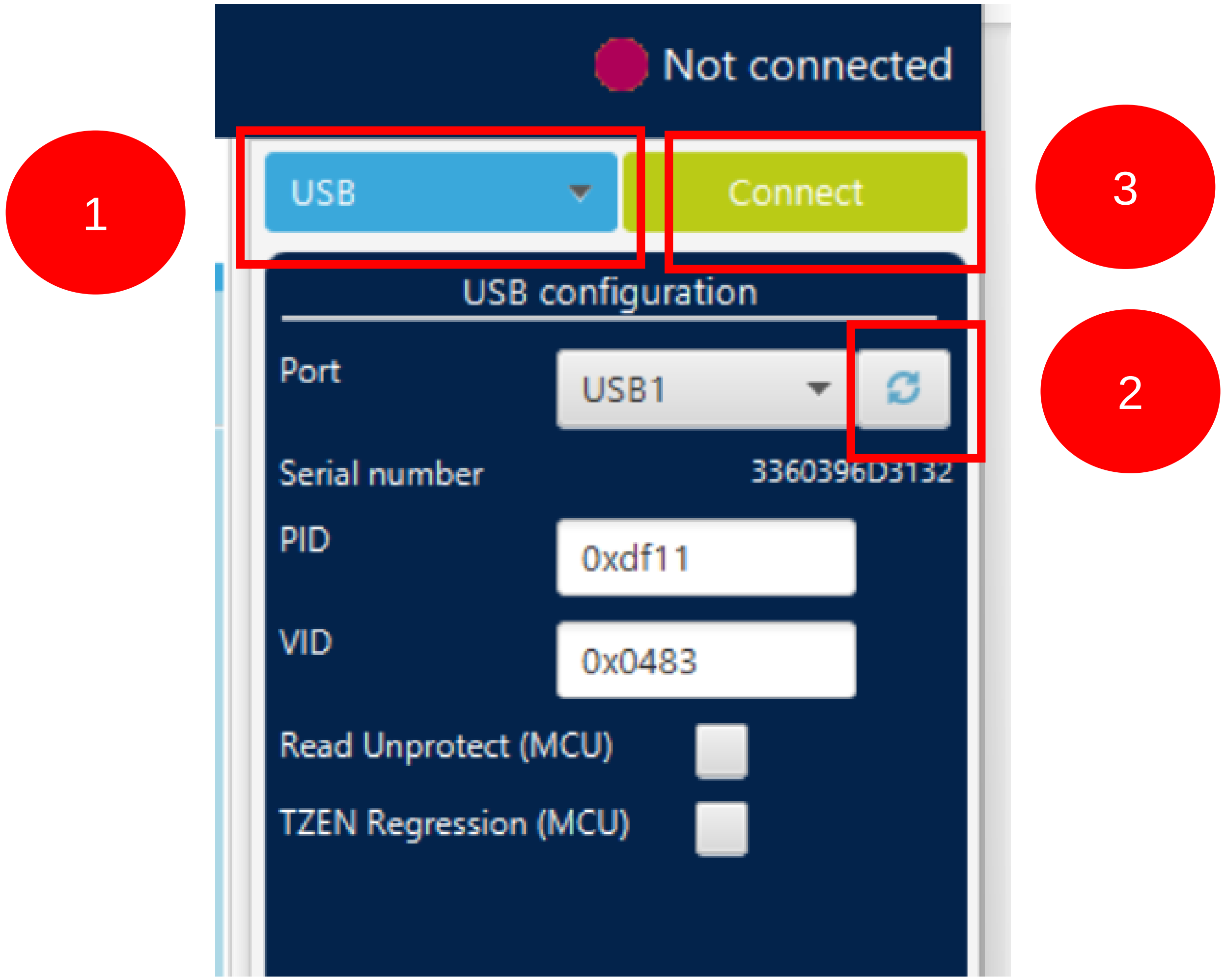
Running demo – Status message after reset



```
Booting...
Active Bank is bank 1
Button not pressed
Application Code NOT found
Launching USB DFU
```

- After Resetting the board will boot from Bank 1.
- Button is not pressed so it will check if there is an application code in the active bank.
- But here it is blank so it will automatically execute USB DFU.

Running demo – Connecting to USB DFU device



Port USB1

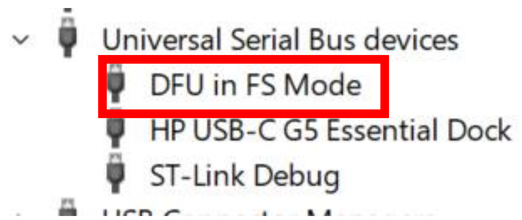
Serial number 3360396D3132

PID 0xdf11

VID 0x0483

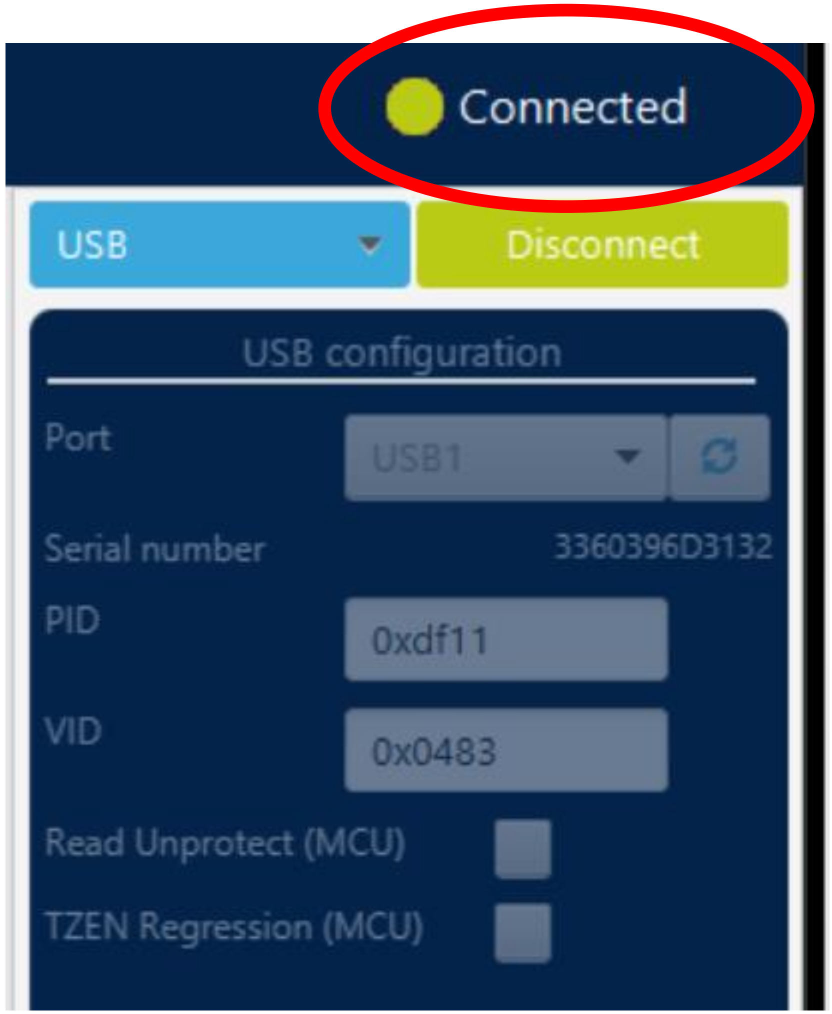
Read Unprotect (MCU)

TZEN Regression (MCU)



- The device is enumerated as a USB DFU device.
- Using STM32CubeProgrammer connect to the device using USB as seen here following these steps

Running demo – Connected to the USB DFU device



USB configuration

Port: USB1

Serial number: 3360396D3132

PID: 0xdf11

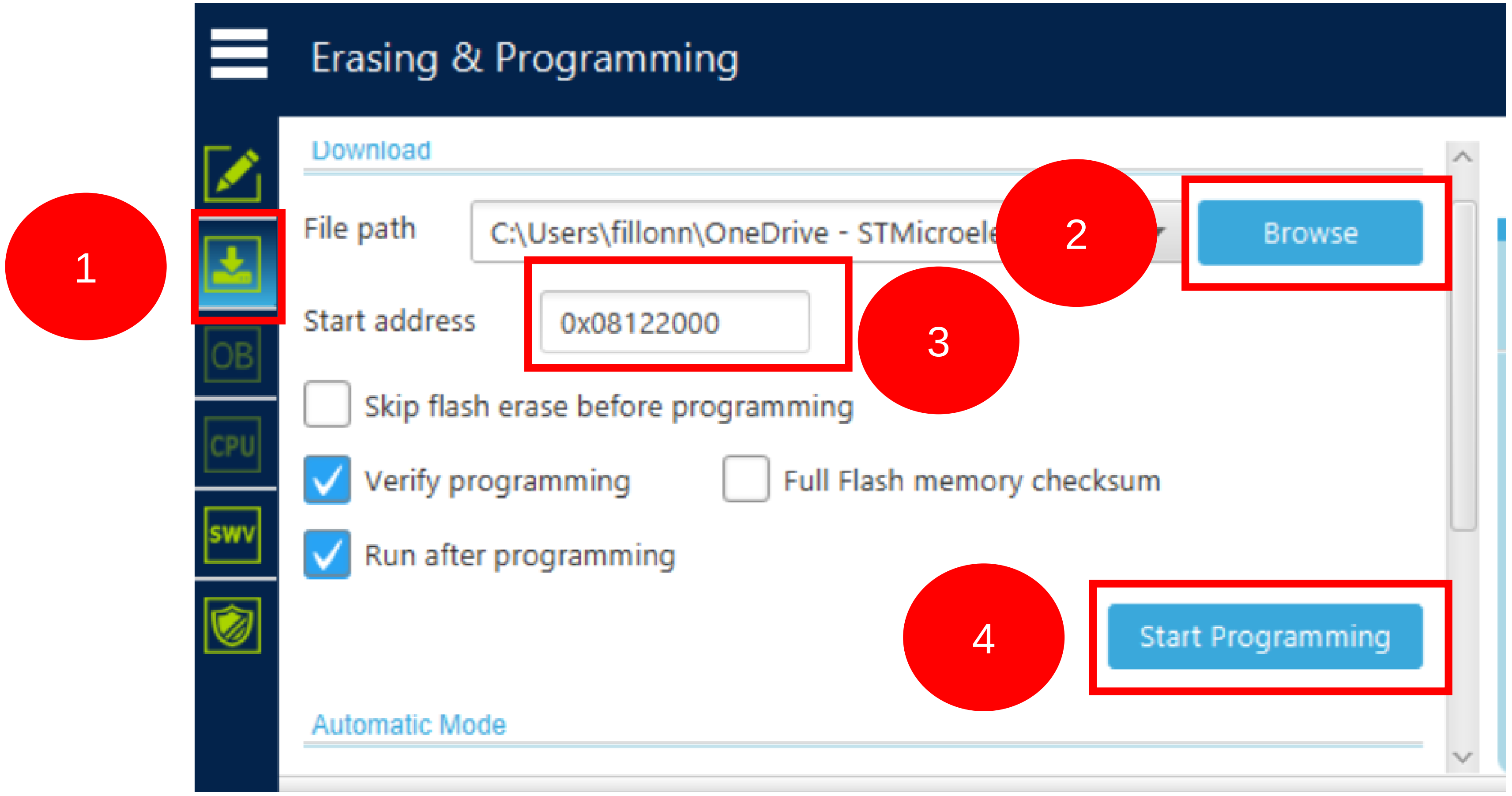
VID: 0x0483

Read Unprotect (MCU)

TZEN Regression (MCU)

- After clicking connect, the STM32CubeProgrammer is now in connected state and we can proceed to do a firmware upgrade,

Running demo – performing a firmware upgrade on the other bank (bank 2)



File path C:\Users\fillonn\OneDrive - STMicroele

Start address 0x08122000

☐ Skip flash erase before programming

☑ Verify programming ☐ Full Flash memory checksum

☑ Run after programming

- Follow these steps to do a firmware upgrade on the other bank.

2 Browse to point to where the file "Application_Example_for_STM32H5_USB_DFU_Dual_Bank.bin" is located

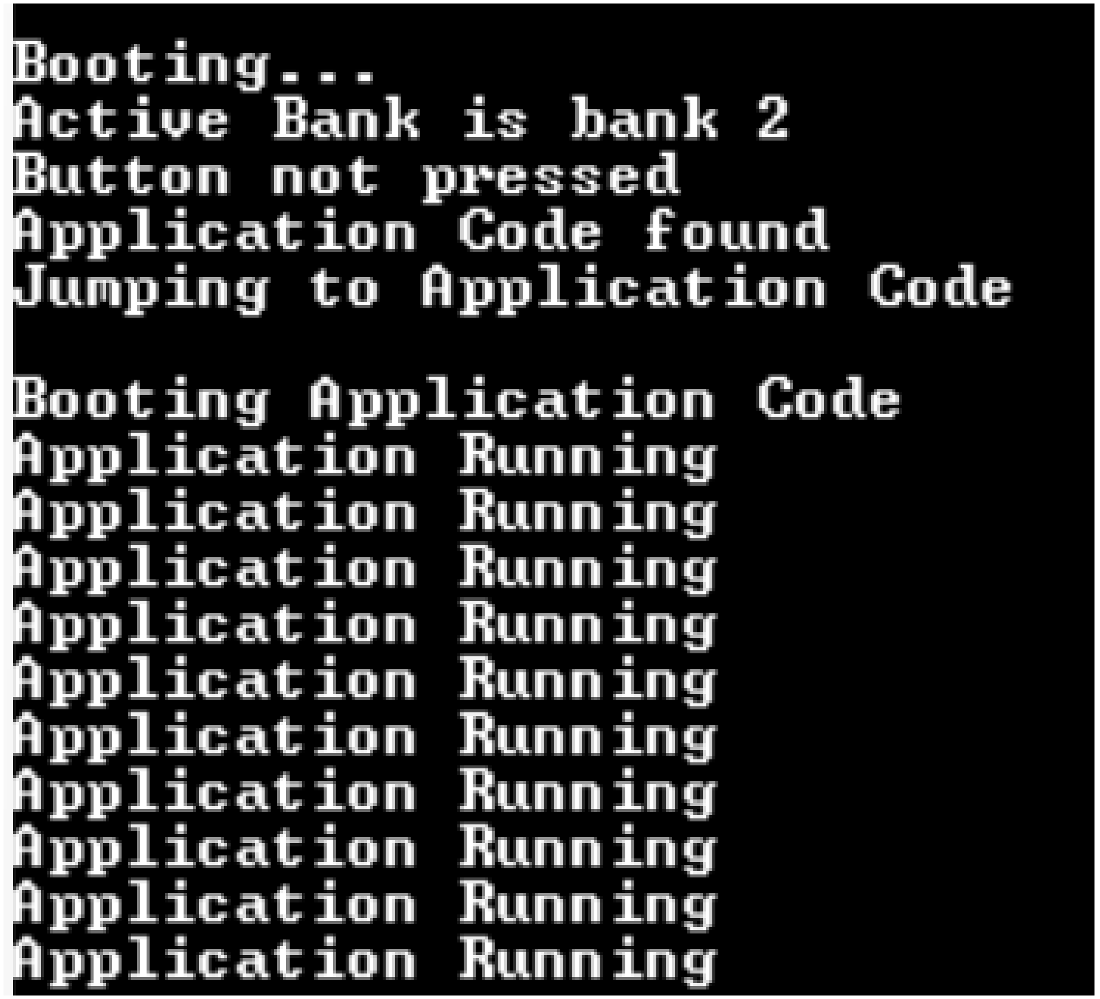
Running demo – Log from STM32CubeProgrammer after firmware upgrade

```
16:24:34 : Memory Programming ...
  16:24:34 : Opening and parsing file:
Application_Example_for_STM32H5_USB_DFU_Dual_Bank.bin
  16:24:34 :   File          :
Application_Example_for_STM32H5_USB_DFU_Dual_Bank.bin
  16:24:34 :   Size          : 34.42 KB
  16:24:34 :   Address       : 0x08122000
  16:24:34 : Erasing memory corresponding to segment 0:
  16:24:34 : erasing sector 0017 @: 0x08122000 done
  16:24:34 : erasing sector 0018 @: 0x08124000 done
  16:24:35 : erasing sector 0019 @: 0x08126000 done
  16:24:35 : erasing sector 0020 @: 0x08128000 done
  16:24:35 : erasing sector 0021 @: 0x0812a000 done
  16:24:35 : Download in Progress:
  16:24:44 : File download complete
  16:24:44 : Time elapsed during download operation:
00:00:10.521
  16:24:44 : Verifying ...
  16:24:44 : Read progress:
  16:24:45 : Download verified successfully
  16:24:45 : RUNNING Program ...
  16:24:45 :   Address:      : 0x08122000
  16:24:45 : Start operation achieved successfully
  16:24:45 : Warning: Connection to USB device is lost
  16:24:45 : Disconnected from device.
```

Connection is lost because we are now running the application code

life.augmented

Running demo – After the firmware upgrade is done on bank 2



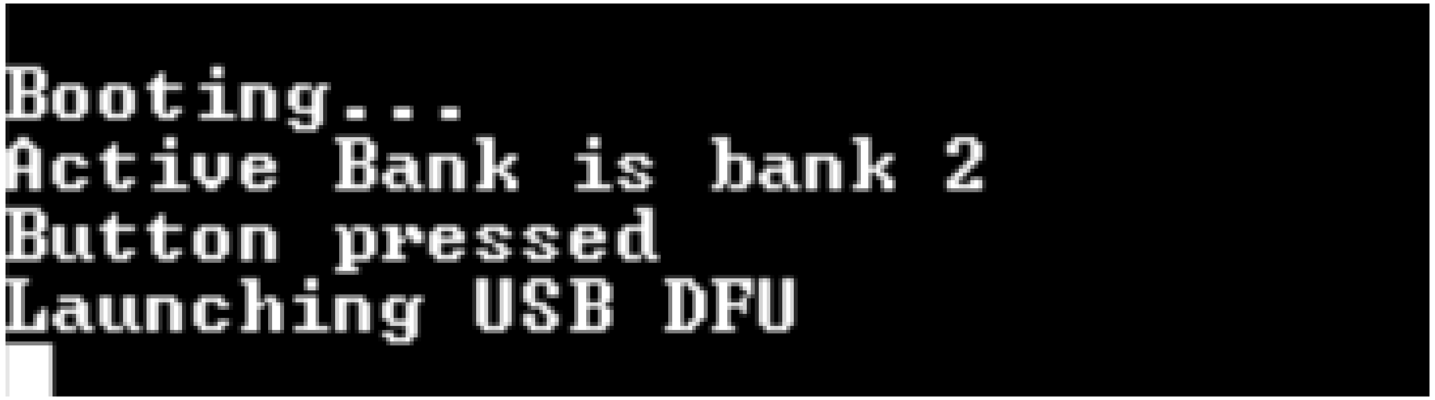
```
Booting...
Active Bank is bank 2
Button not pressed
Application Code found
Jumping to Application Code

Booting Application Code
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
```

- This shows the terminal messages after the firmware upgrade has been performed.
- This shows that we are now running from bank 2 because after the firmware upgrade is done, we perform a bank swap.
- This shows that in this case the user button is not pressed.
- This shows that because an application code was found in the Application code section of the bank 2 that we are jumping to application.
- This shows the application code running.

life.augmented

Running demo – performing a new firmware upgrade

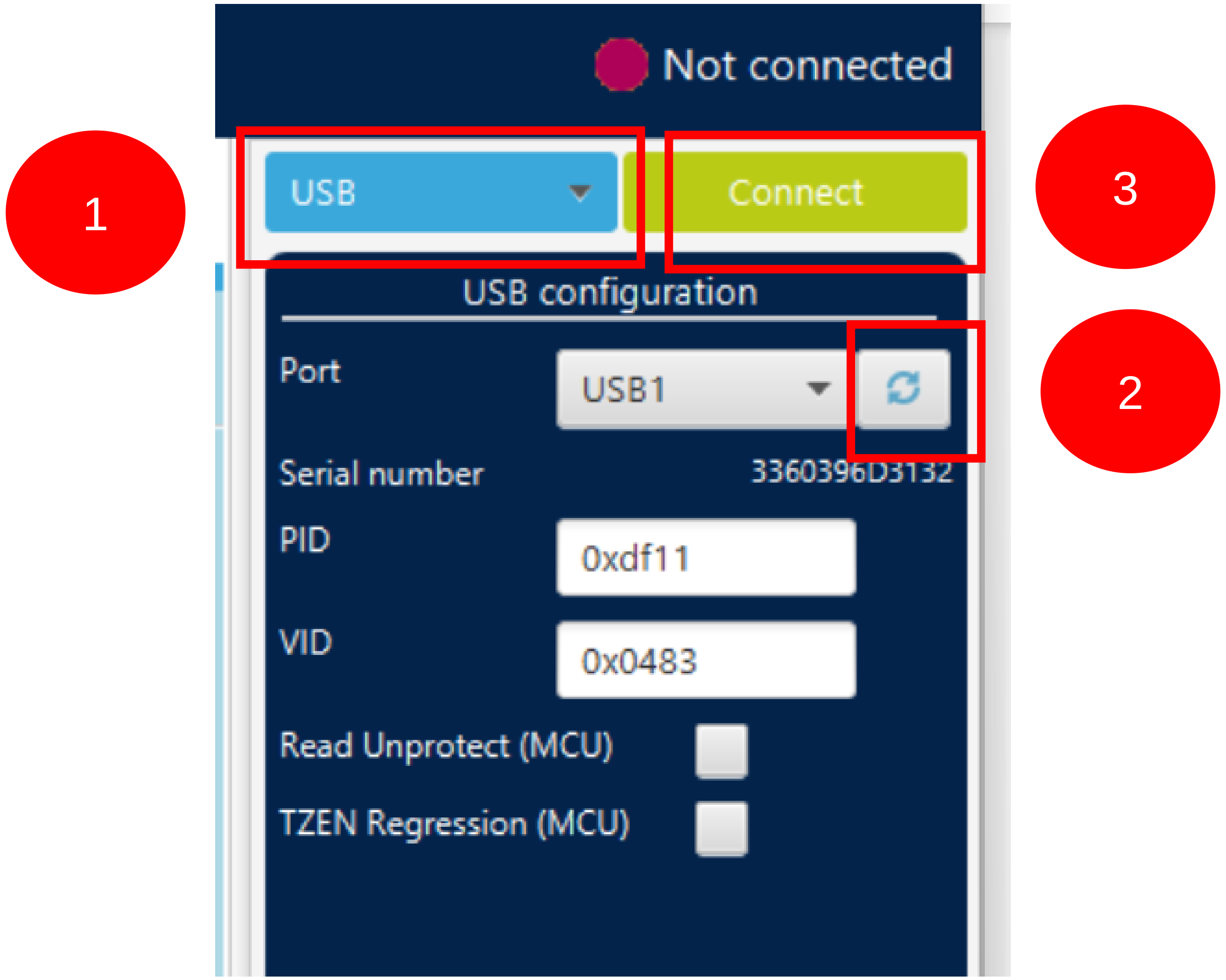
To do a new firmware upgrade we will hold the user button prior to and after releasing the reset button, this will force the execution of the USB DFU.



```
Booting...
Active Bank is bank 2
Button pressed
Launching USB DFU
```

- This still shows that the active bank is bank 2.
- This shows that in this case the user button is pressed.
- This shows that the USB DFU is running

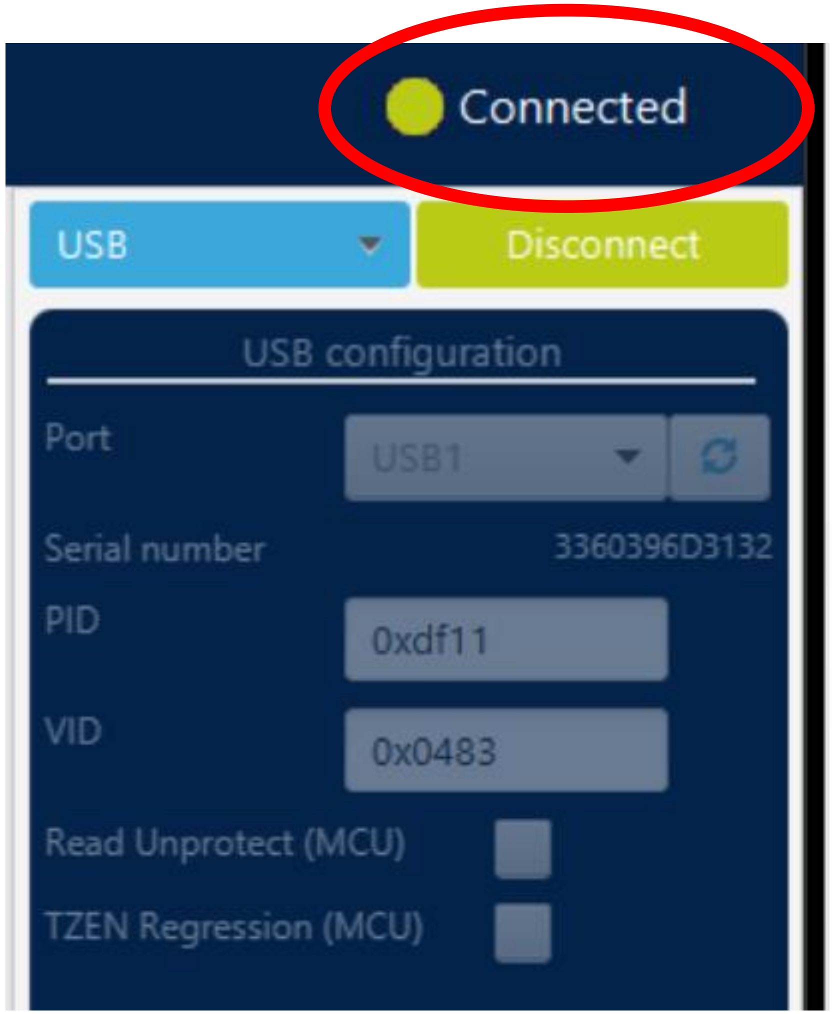
Running demo – Connecting to USB DFU device



Port: USB1
Serial number: 3360396D3132
PID: 0xdf11
VID: 0x0483
Read Unprotect (MCU): ☐
TZEN Regression (MCU): ☐

- The device is enumerated as a USB DFU device.
- Using STM32CubeProgrammer connect to the device using USB as seen here following these steps.

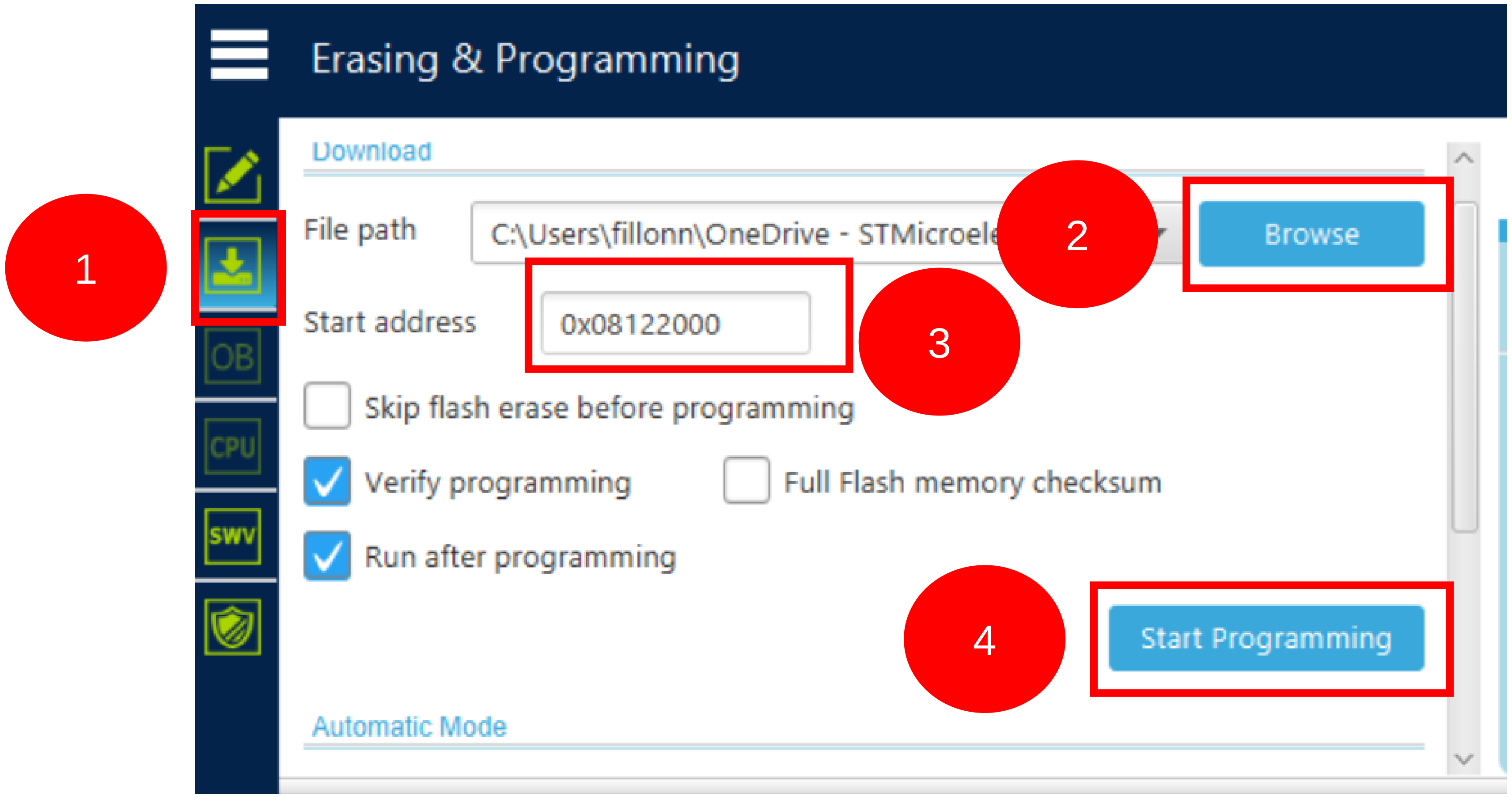
Running demo – Connected to the USB DFU device



Port: USB1
Serial number: 3360396D3132
PID: 0xdf11
VID: 0x0483
Read Unprotect (MCU)
TZEN Regression (MCU)

- After clicking connect, the STM32CubeProgrammer is now in “Connected” state and we can proceed to do a firmware upgrade.

Running demo – performing a firmware upgrade on the other bank (bank 1)



File path C:\Users\fillonn\OneDrive - STMicroele

Start address 0x08122000

☐ Skip flash erase before programming

✓ Verify programming ☐ Full Flash memory checksum

✓ Run after programming

- Follow these steps to do a firmware upgrade on the other bank.

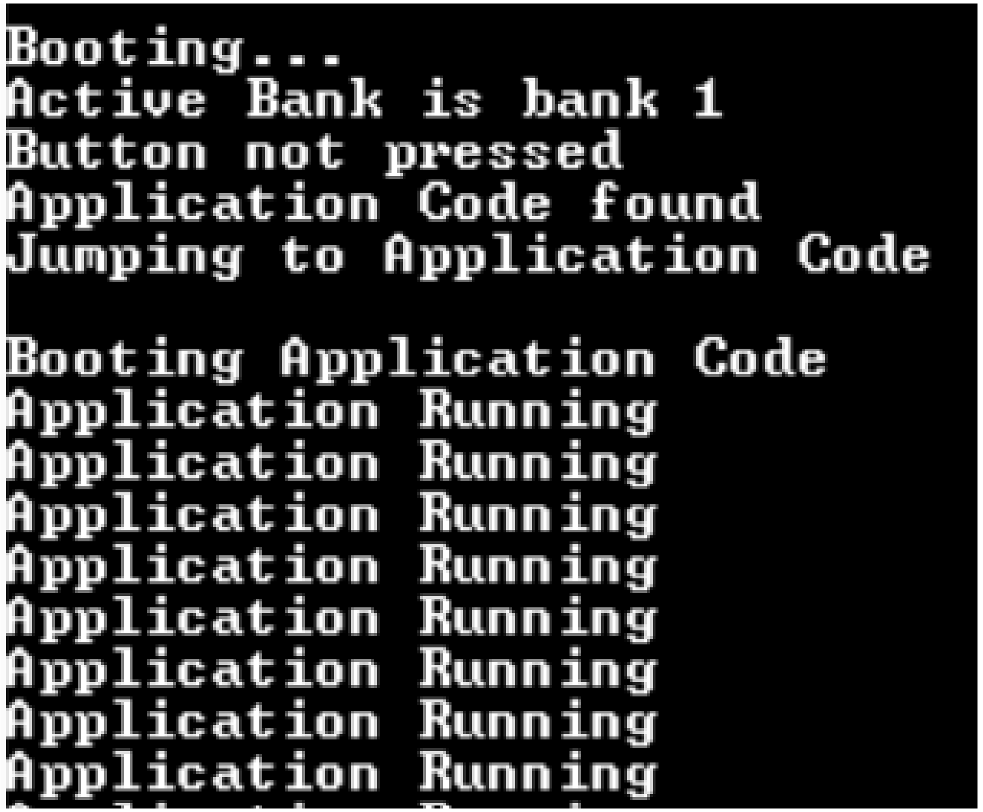
2 Browse to point to where the file "Application_Example_for_STM32H5_USB_DFU_Dual_Bank.bin" is located

Running demo - Log from STM32CubeProgrammer after firmware upgrade

```
 16:31:00 : Memory Programming ...
  16:31:00 : Opening and parsing file:
Application_Example_for_STM32H5_USB_DFU_Dual_Bank.bin
  16:31:00 :   File          :
Application_Example_for_STM32H5_USB_DFU_Dual_Bank.bin
  16:31:00 :   Size          : 34.42 KB
  16:31:00 :   Address       : 0x08122000
  16:31:00 : Erasing memory corresponding to segment 0:
  16:31:00 : erasing sector 0017 @: 0x08122000 done
  16:31:00 : erasing sector 0018 @: 0x08124000 done
  16:31:00 : erasing sector 0019 @: 0x08126000 done
  16:31:01 : erasing sector 0020 @: 0x08128000 done
  16:31:01 : erasing sector 0021 @: 0x0812a000 done
  16:31:01 : Download in Progress:
  16:31:10 : File download complete
  16:31:10 : Time elapsed during download operation:
00:00:10.345
  16:31:10 : Verifying ...
  16:31:10 : Read progress:
  16:31:10 : Download verified successfully
  16:31:10 : RUNNING Program ...
  16:31:10 :   Address:      : 0x08122000
  16:31:11 : Start operation achieved successfully
  16:31:11 : Warning: Connection to USB device is lost
  16:31:11 : Disconnected from device.
```

Connection is lost because we are now running the application code

Running demo after firmware upgrade on bank 1



```
Booting...
Active Bank is bank 1
Button not pressed
Application Code found
Jumping to Application Code

Booting Application Code
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
Application Running
```

- This shows the terminal messages after the firmware upgrade has been performed.
- This shows that we are now running from bank 1 because after this new firmware upgrade was completed, we performed a bank swap
- This shows that in this case the user button is not pressed.
- This shows that because an application code was found in the Application code section of the bank 1 that we are jumping to application.
- This shows the application code running.