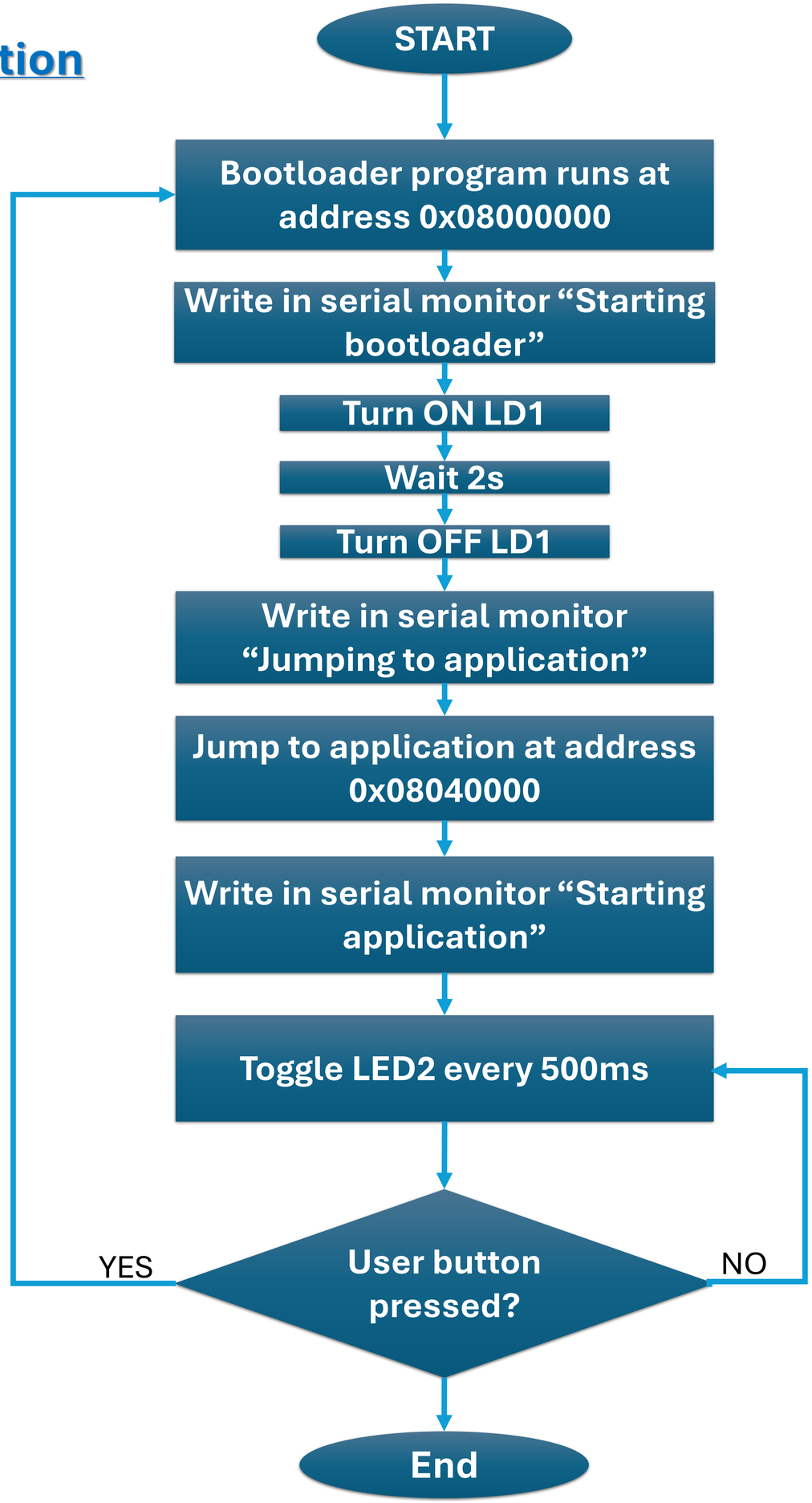
NEOCRUX
SYSTEMS

Create and switch between bootloader and application

Case Study: STM32F767ZI

Diagram of implementation



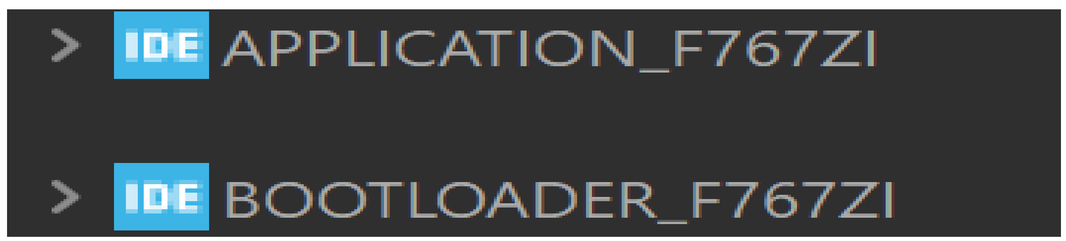
Steps of implementation

STEP1: Understand the FLASH memory architecture from reference manual

Block	Name	Bloc base address on AXIM interface	Block base address on ITCM interface	Sector size
Main memory block	Sector 0	0x0800 0000 - 0x0800 7FFF	0x0020 0000 - 0x0020 7FFF	32 KB
	Sector 1	0x0800 8000 - 0x0800 FFFF	0x0020 8000 - 0x0020 FFFF	32 KB
	Sector 2	0x0801 0000 - 0x0801 7FFF	0x0021 0000 - 0x0021 7FFF	32 KB
	Sector 3	0x0801 8000 - 0x0801 FFFF	0x0021 8000 - 0x0021 FFFF	32 KB
	Sector 4	0x0802 0000 - 0x0803 FFFF	0x0022 0000 - 0x0023 FFFF	128 KB
	Sector 5	0x0804 0000 - 0x0807 FFFF	0x0024 0000 - 0x0027 FFFF	256 KB
	Sector 6	0x0808 0000 - 0x080B FFFF	0x0028 0000 - 0x002B FFFF	256 KB
	Sector 7	0x080C 0000 - 0x080F FFFF	0x002C 0000 - 0x002F FFFF	256 KB
	Sector 8	0x0810 0000 - 0x0813 FFFF	0x0030 0000 - 0x0033 FFFF	256 KB
	Sector 9	0x0814 0000 - 0x0817 FFFF	0x00340000 - 0x0037 FFFF	256 KB
	Sector 10	0x0818 0000 - 0x081B FFFF	0x0038 0000 - 0x003B FFFF	256 KB
	Sector 11	0x081C 0000 - 0x081F FFFF	0x003C 0000 - 0x003F FFFF	256 KB

STEP2: Coding game in stm32cube IDE

STEP2.1: Create two stm32 projects one for bootloader and the other for application



Steps of implementation

STEP2.2: **Modify linker files for both the bootloader and application**

- **For bootloader:**

STM32F767ZITX_FLASH.ld

```
/* Memories definition */
MEMORY
{
  RAM    (xrw)    : ORIGIN = 0x20000000,   LENGTH = 512K
  FLASH    (rx)    : ORIGIN = 0x8000000,   LENGTH = 128K
}
```

- **For application:**

STM32F767ZITX_FLASH.ld

```
/* Memories definition */
MEMORY
{
  RAM    (xrw)    : ORIGIN = 0x20000000,   LENGTH = 512K
  FLASH    (rx)    : ORIGIN = 0x8040000,   LENGTH = 1792K
}
```

STEP2.3: **Modify vector table base for application**

- For application, we changed the base of the vector table, so we should change the vector table offset.

system_stm32f7xx.c

```
#if !defined(VECT_TAB_OFFSET)
#define VECT_TAB_OFFSET         0x00040000U

#endif /* VECT_TAB_OFFSET */
#endif /* USER_VECT_TAB_ADDRESS */
/*****************************************************
```

Steps of implementation

STEP2.4: Create a handler function that allows to jump to bootloader and other to application

- ❑ The «4» added to the address allows to jump directly to the reset handler

```c
static void goto_bootloader(void)
{
    write("\nJumping to bootloader");
    void (*app_reset_handler) (void) =
            (void*) (*(volatile uint32_t *) (0x08000000 + 4));
    app_reset_handler();
}
```

```c
static void goto_application(void)
{
    write("\nJumping to application");
    void (*app_reset_handler) (void) =
            (void*) (*(volatile uint32_t *) (0x08040000 + 4));
    app_reset_handler();
}
```

Steps of implementation

STEP2.5: **Create bootloader and application codes**

- **For bootloader**

```
write("\nStarting bootloader");

HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
HAL_Delay(2000);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);

goto_application();
```

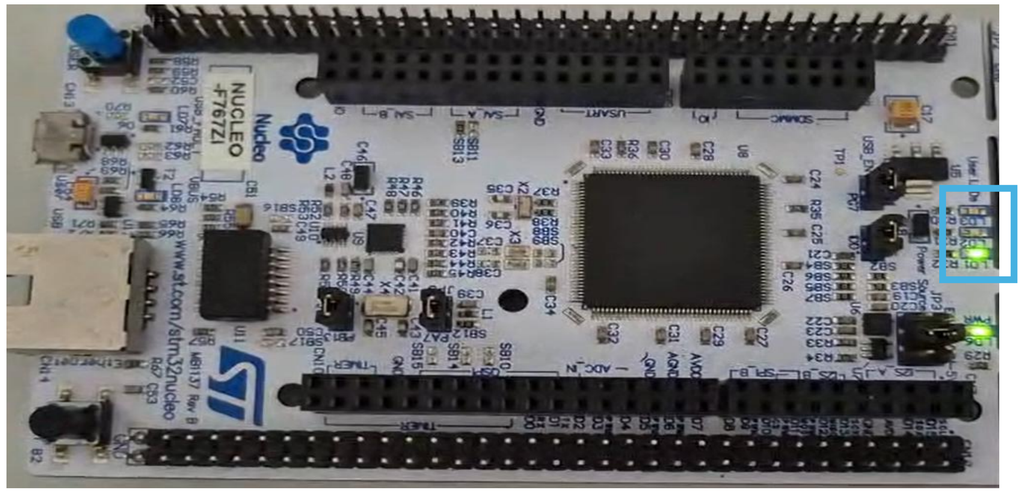
- **For application**

```
write("\nStarting application");
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    currentTick = HAL_GetTick();

    if ((currentTick - tickstart) >= wait)
    {
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_7);
        tickstart = currentTick;
    }

    Bt_state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13);
    if(Bt_state == GPIO_PIN_SET){
        goto_bootloader();
    }

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
}
```

Results

After powering the MCU, the CPU goes to the first address of the Flash memory which is 0X08000000.

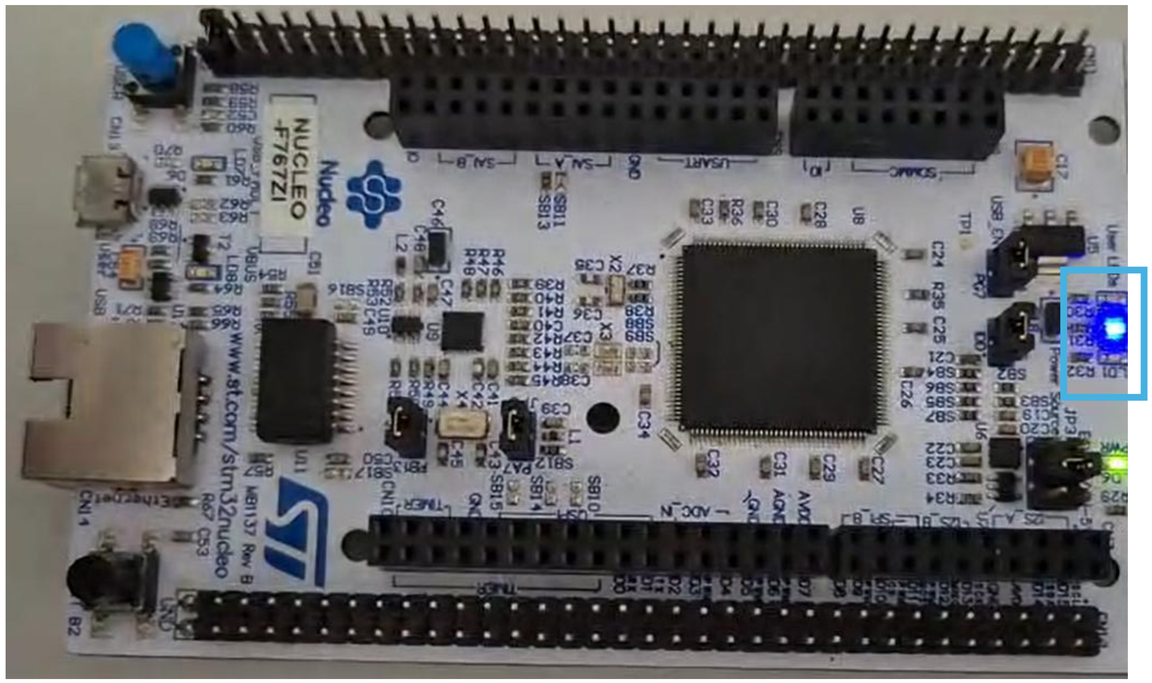
- **It’s the bootloader**




```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindow
s

PS C:\Users\      > python -m serial.tools.miniterm COM6 115200
--- Miniterm on COM6  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

Starting bootloader
Jumping to application
Starting application
```

Results

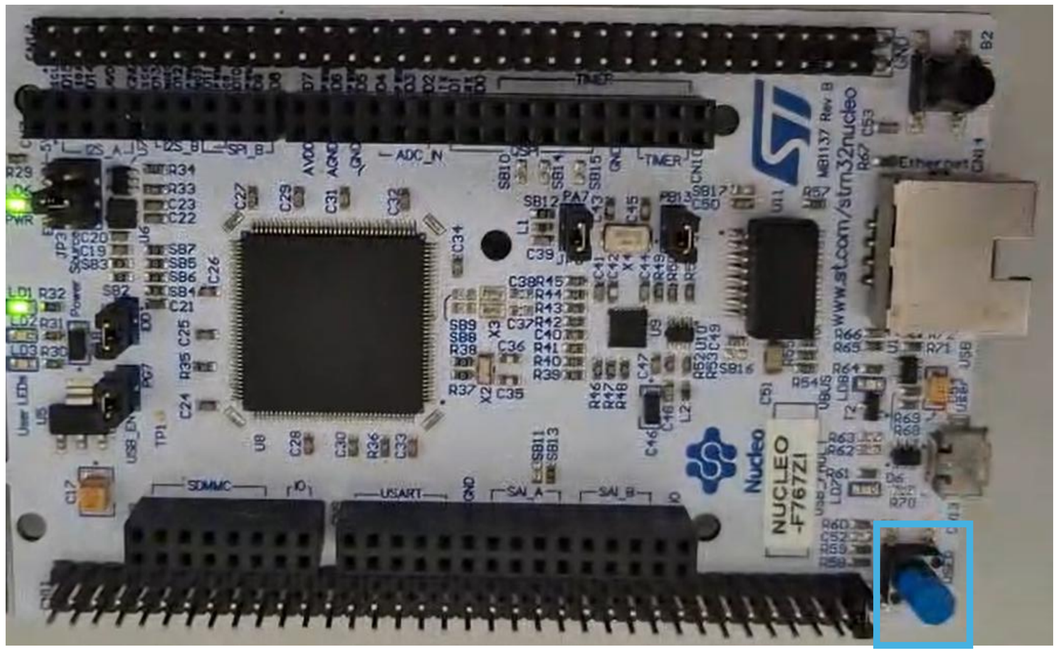
- Then the bootloader takes care of jumping to the address 0x08040000
  - **It’s the application**



- Our application is a very simple blinking led, and checks if the user button is pressed, if it’s the case it will jump to the bootloader

Results

- Our application is a very simple blinking led, and checks if the user button is pressed, if it's the case it will jump to the bootloader




```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindow
s

PS C:\Users\      > python -m serial.tools.miniterm COM6 115200
--- Miniterm on COM6  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

Starting bootloader
Jumping to application
Starting application
Jumping to bootloader
Starting bootloader
Jumping to application
Starting application
```

See you soon

NEOCRUX
SYSTEMS

NEOCRUX_SYSTEMS
NEOCRUX_SYSTEMS