

CPE3500: Embedded Digital Signal Processing

Lab Exercise 1: Setting up the STM32CubeIDE

Objective:

In preparation for future labs, students will set up an integrated development environment on their personal computer.

Introduction:

STM32CubeIDE is a comprehensive and user-friendly platform for developing and debugging STM32 microcontroller applications. This will prepare the student for subsequent labs involving digital signal processing exercises that can run on Nucleo-L476 microcontroller board.

In this exercise students will:

- Install the STM32CubeIDE on their personal computer in preparation for future exercises which will require the use of the IDE to create projects.
- Create and configure the STM32Cube project and generate the initialization code.
- Program and use HAL functions to blink a LED on the NUCLEO-L476RG board.

Required Equipment:

Personal computer (Windows/Mac OS)

Nucleo-L476RG Board (64-pin)

USB Type-A (or Type-C) to Mini-B cable

Procedure:

1. Register to ST website

Using the following link, create a free account (use your student email) on ST.com.

https://www.st.com/content/st_com/en/user-registration.html

2. Download and install STM32CubeIDE (Windows, MAC)

After creating an account, using the following link to the installation page where you can download the installer.

<https://www.st.com/en/development-tools/stm32cubeide.html#get-software>

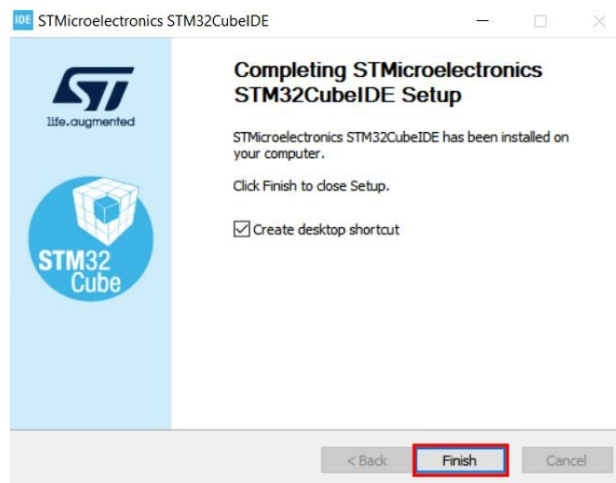
The minimal hardware requirements are:

- 2 GB RAM (minimum)
- 6 GB free space
- Internet access

Get Software					
Part Number	General Description	Latest version	Download	All versions	
+ STM32CubeIDE-DEB	STM32CubeIDE Debian Linux Installer	1.14.0	Get latest	Select version	
+ STM32CubeIDE-Lnx	STM32CubeIDE Generic Linux Installer	1.14.0	Get latest	Select version	
+ STM32CubeIDE-Mac	STM32CubeIDE macOS Installer	1.14.0	Get latest	Select version	
+ STM32CubeIDE-RPM	STM32CubeIDE RPM Linux Installer	1.14.0	Get latest	Select version	
+ STM32CubeIDE-Win	STM32CubeIDE Windows Installer	1.14.1	Get latest	Select version	

From the page after the link, download the installer appropriate to your personal computer system. (Windows or Mac). You may be asked to login with the account you have created.

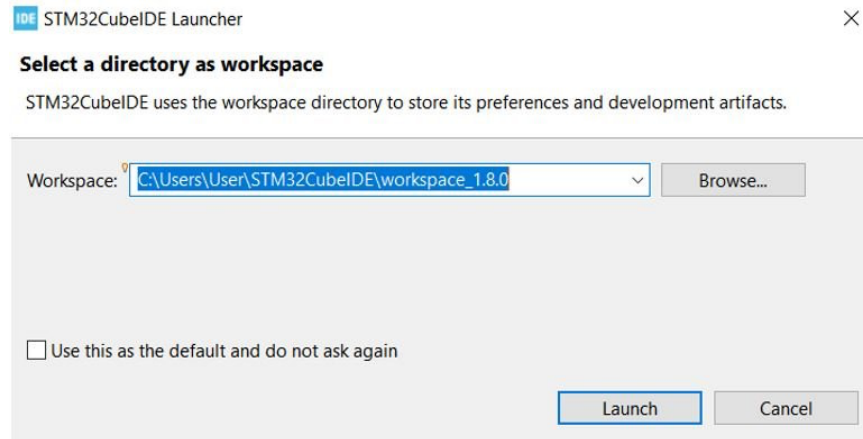
After completing the download, open the .zip file and start installation.



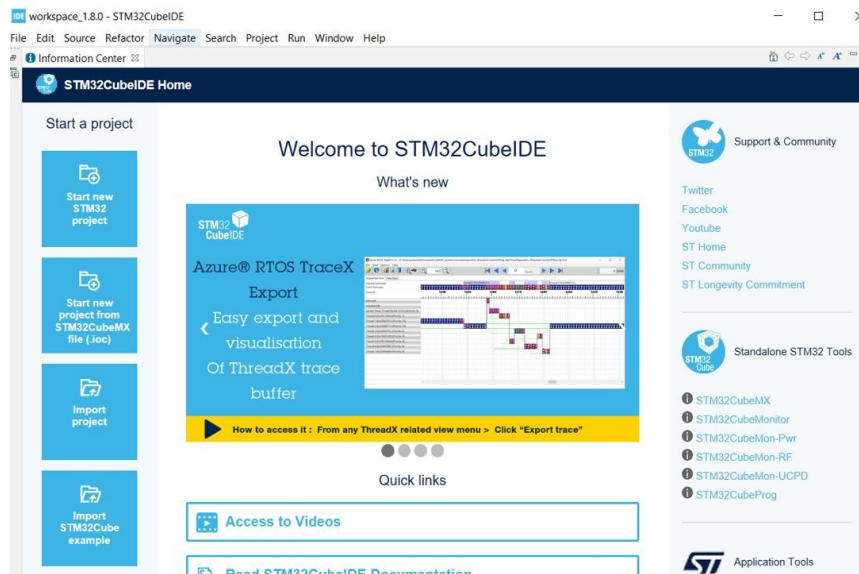
After the installation is complete, click 'Finish' to close the setup. Your desktop will now have a shortcut to STM32CubeIDE created.

3. Create your first Project in STM32CubeIDE

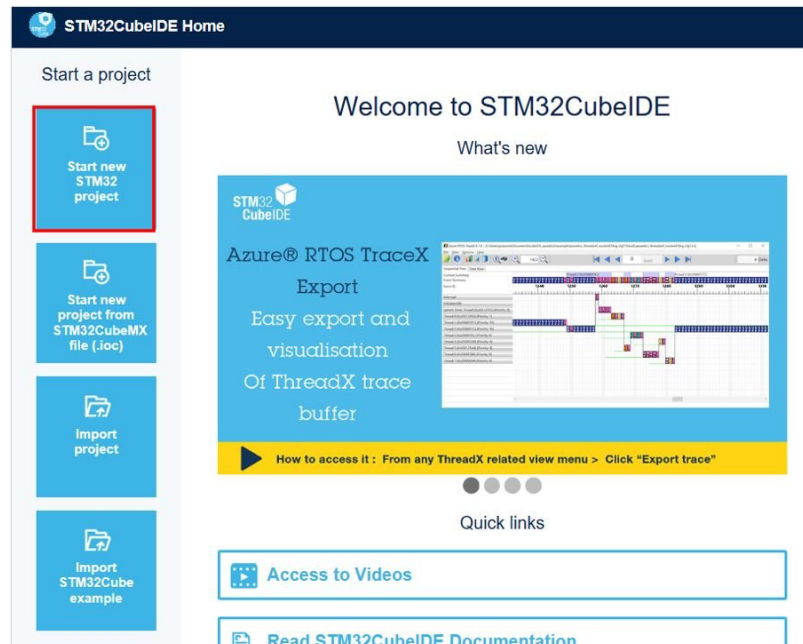
Open the application. You will be asked to specify the directory for the workplace. You can specify the directory and also tick the box below to keep this as the default directory. Next, click 'Launch' to start the application.



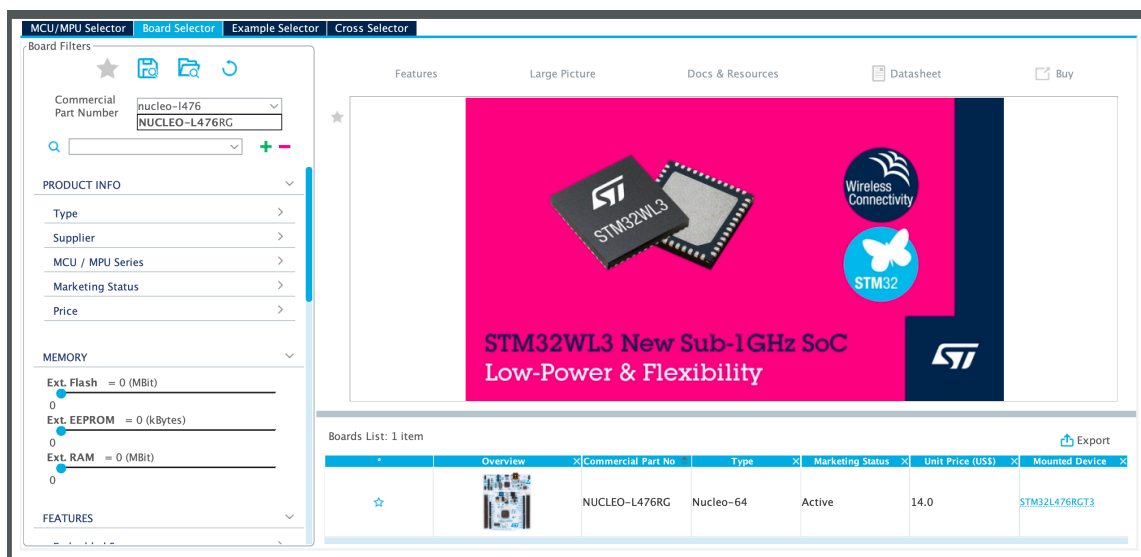
The STM32CubeIDE workspace will open.



Now click 'Start new STM32 project'.

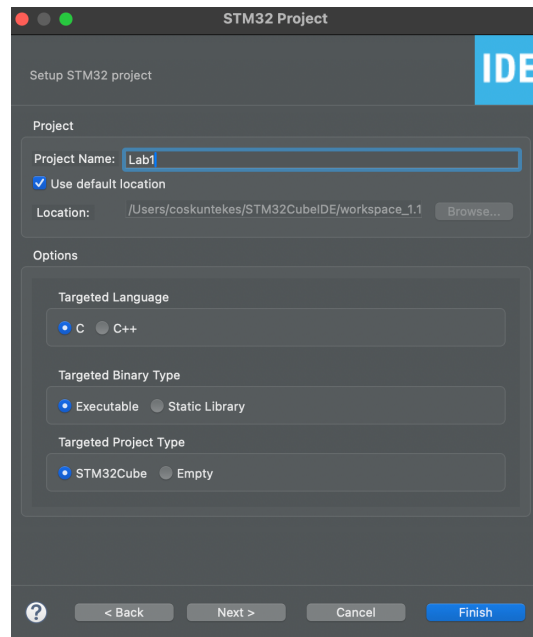


The Target Selection will open.

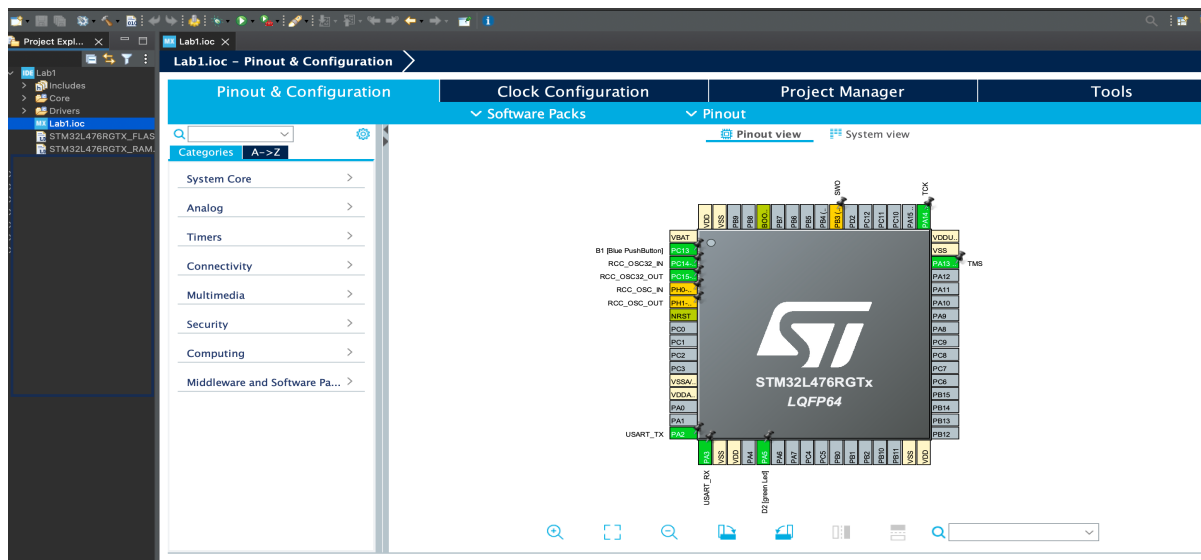


Click on "Board Selector" tab and in part number section type in NUCLEO-L476RG. When you select your development board from the dropdown menu, you will see the board listed at the bottom. Click on the board image and click Next.

In the next window below, enter a name for you project and click Finish.

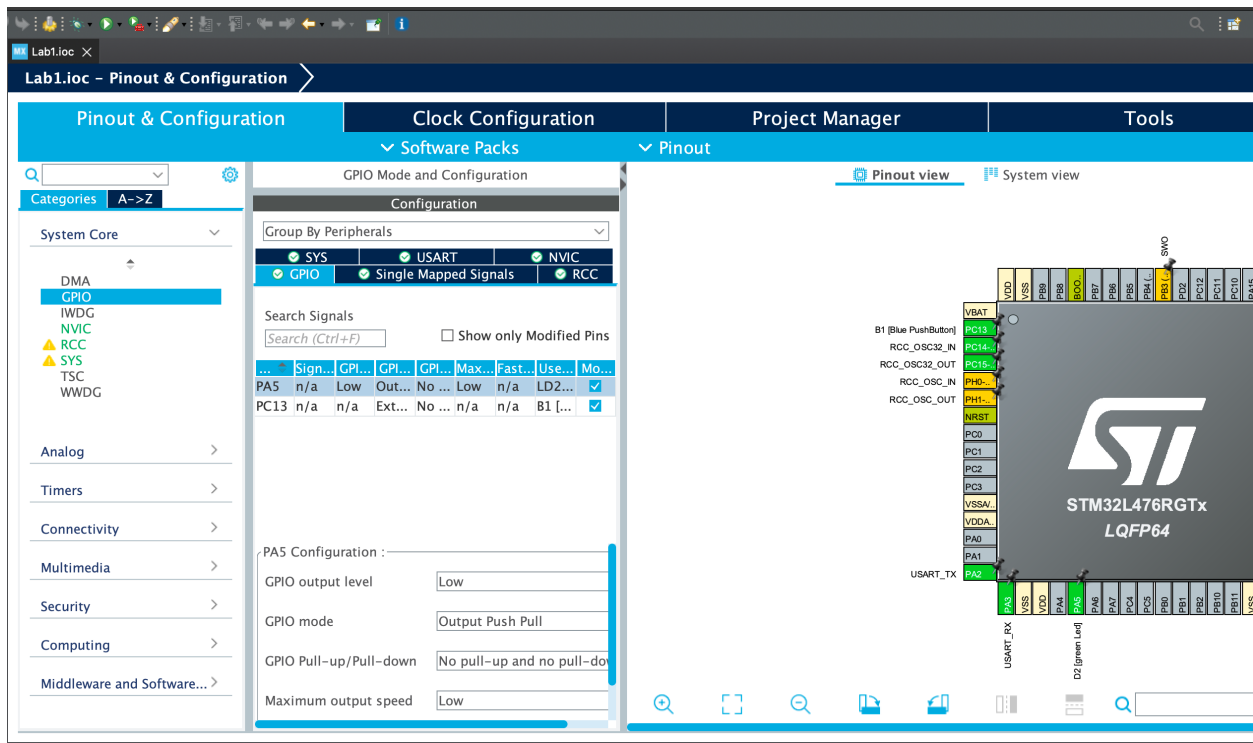


Now the Device Configuration Tool window will open up. We will be able to set pin specific functions here.

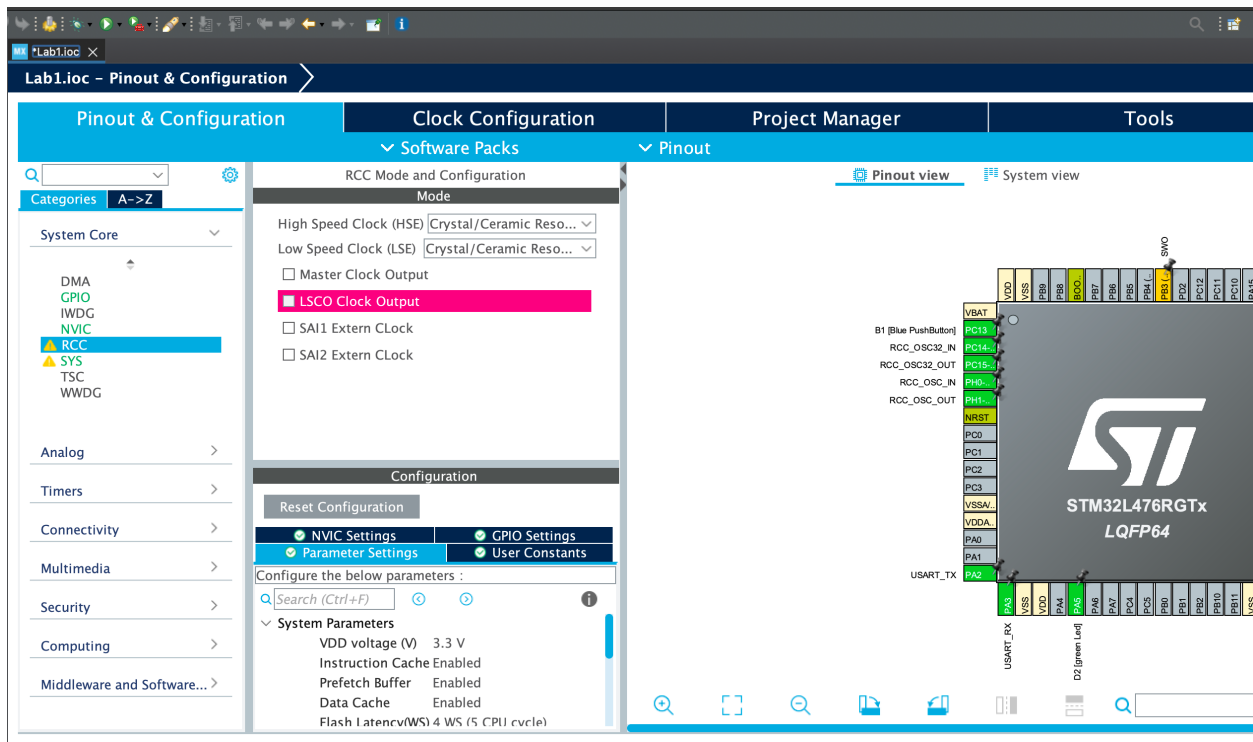


For default configuration of the Nucleo-L476 board, on-board LED is already configured at PA5.

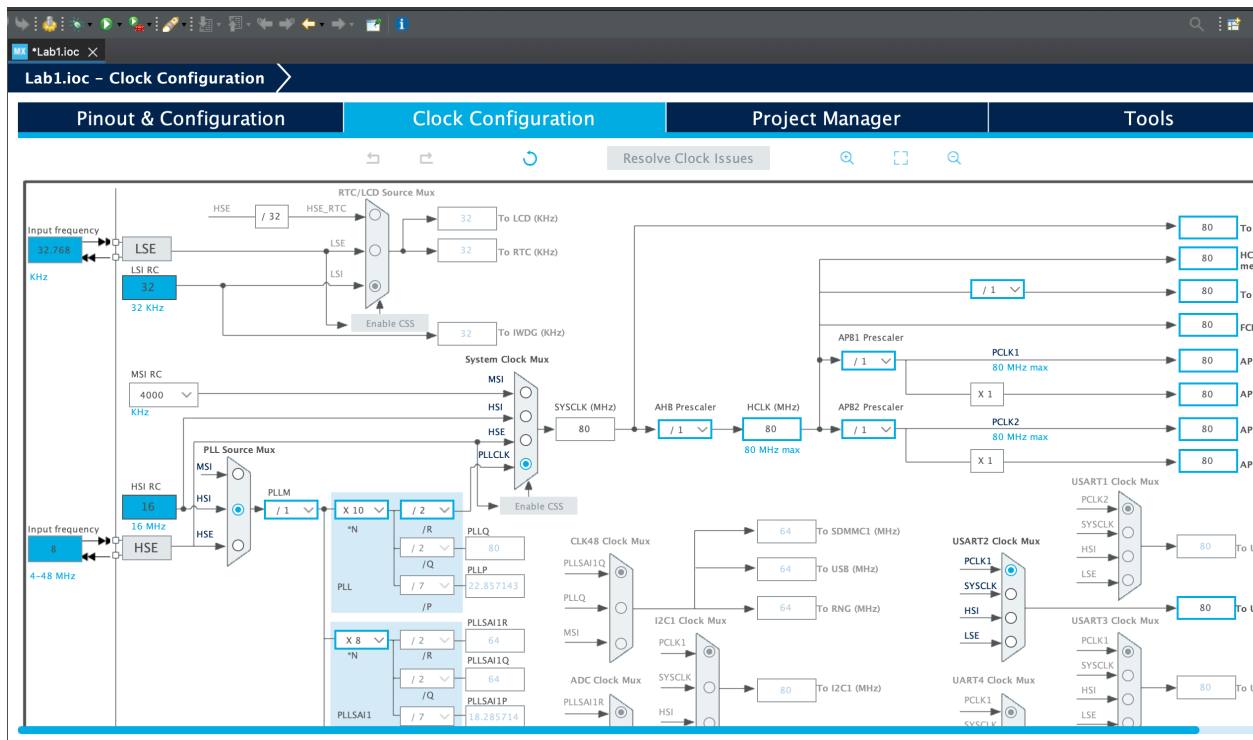
Click on System Core menu on the left and click GPIO to open GPIO Mode and Configuration window.



Click on PA5 (LED port pin) to check configurations are correct for an output port pin.

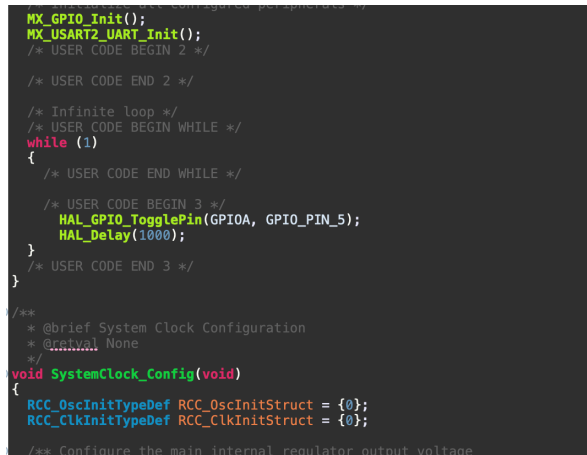


For clock configuration menu, click on RCC and select Crystal/Ceramic Resonator option in HSE (High Speed Clock). Then click on Clock Configuration tab at the top of the project page and check to see system clock is 80 MHz.



When you save the project (by clicking save button or pressing Ctrl+S), the IDE will ask you to generate the code. After clicking “Yes”, the main.c program code will open with all the required configuration lines of codes.

In this exercise, we will control the on-board LED by turning it on/off every 1 sec. To do this, scroll down to main() function and inside the while(1) loop write the below two lines



```
/* USER CODE BEGIN 2 */
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    HAL_Delay(1000);
    /* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Configure the main internal regulator output voltage

```

After this modification, build your project and run!

Lab Exercise 1 Report:

As proof of successfully setting up the IDE and your Blink Led project running, upload pictures of the following to the D2L dropbox.

- A screenshot of your project page on STM32CubeIDE
- A very short video of the Nucleo board with green Led ON/OFF.