

Blinking LED using User Button with STM32CubeMX and HAL

This tutorial will demonstrate how to utilize STM32CubeMX tool to initialize peripherals, build and generate C code using HAL libraries.

After this tutorial you will be able to:

- Create and configure STM32CubeMX project and generate initialization code
- Program and use HAL functions to control LED on Nucleo-L476RG board with User Button

Hardware:

- Nucleo-L476RG board(64-pin),available at: www.st.com/en/evaluation-tools/nucleo-l476rg.html
- Standard-A -to- Mini USB cable

Literature:

- [STM32L476xx Datasheet](#)
- [UM1724](#) User manual STM32 Nucleo-64 boards
- [UM1884](#) Description of STM32L4/L4+ HAL and low-layer drivers
- [UM1718](#) User manual STM32CubeMX for STM32 configuration and initialization C code generation

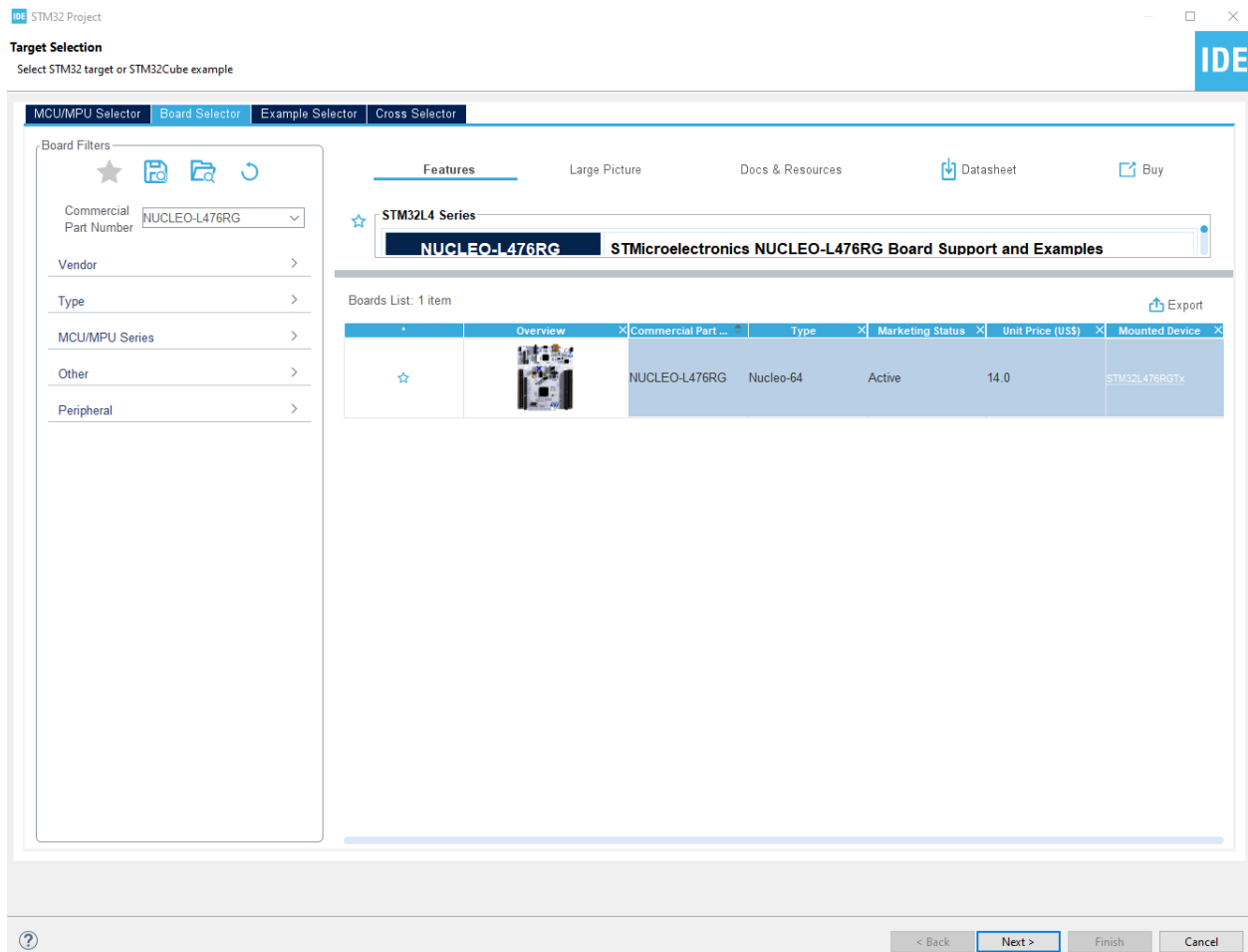
Stages

- 1: Create New Project with STM32CubeMX
- 2: Pinout Configuration
- 3: Clock Configuration
- 4: Configure project and Generate Source Code
- 5: Edit main.c to blink LED with blue button
- 6: Build Project
- 7: Debug the Project

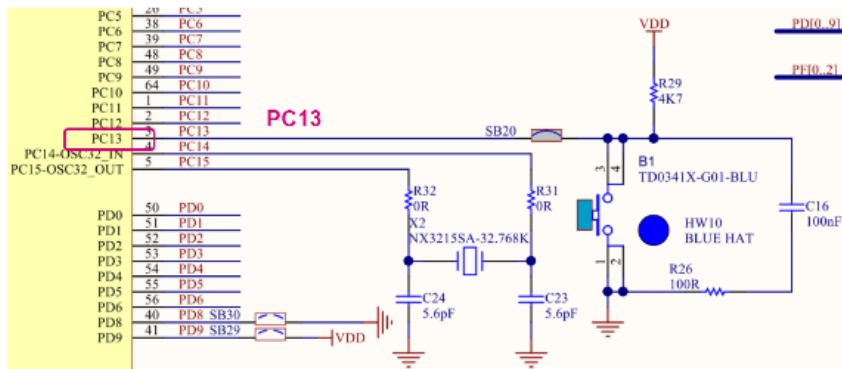


1: CREATE NEW PROJECT USING STM32CUBEMX:

- Open STM32CubeIDE
- Click *File -> New -> STM32 Project*. A target selection window will open.
- From Board Selector type *Nucleo-L476RG*. Select the board and click next.
- Name your project “Nucleo_L476RG_Button” and click Finish.
- Answer “Yes” to “Initialize all peripherals with their default mode?” popup.

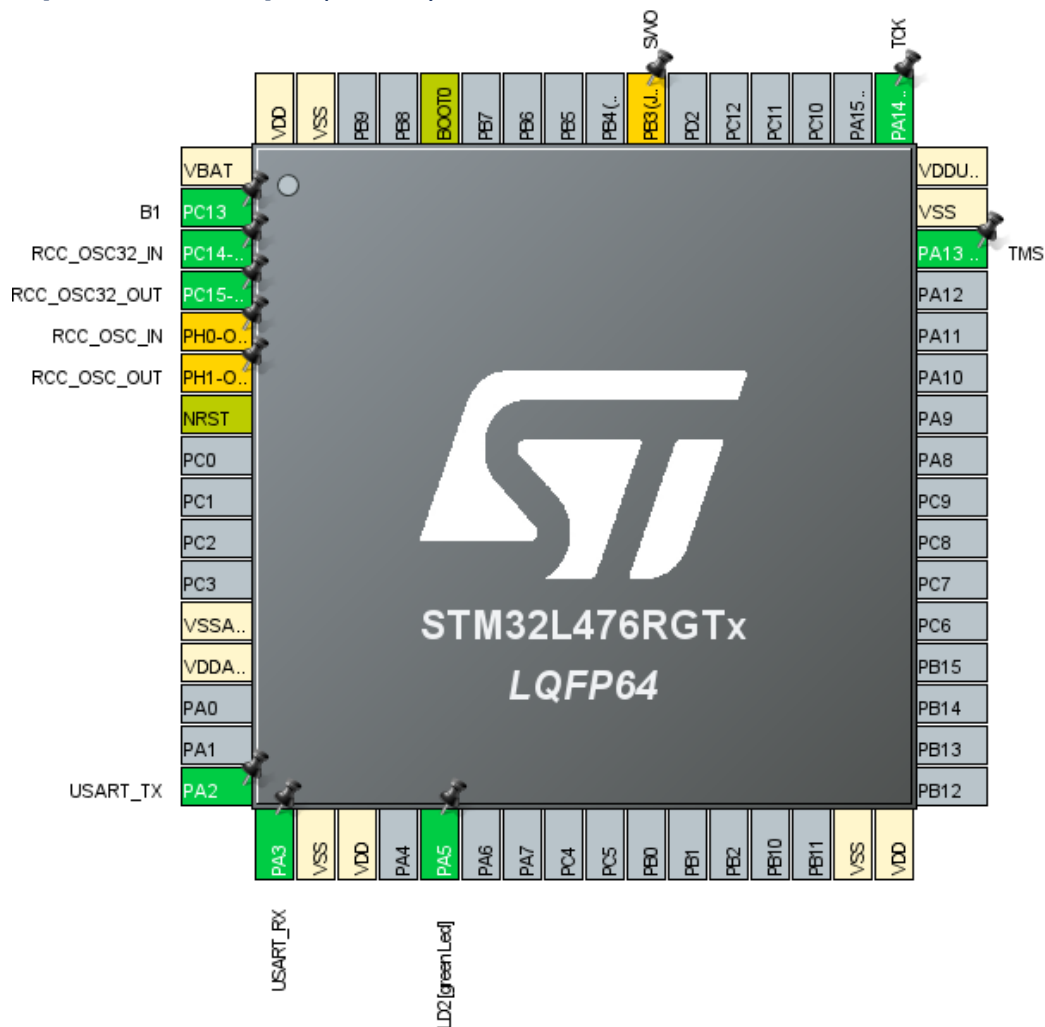


2: PINOUT CONFIGURATION



From the image above we can see user button B1 is connected to PC13. In STM32CubeMX right click PC13 and select *GPIO_Input*. We also want to verify PA5 is set to *GPIO_output*

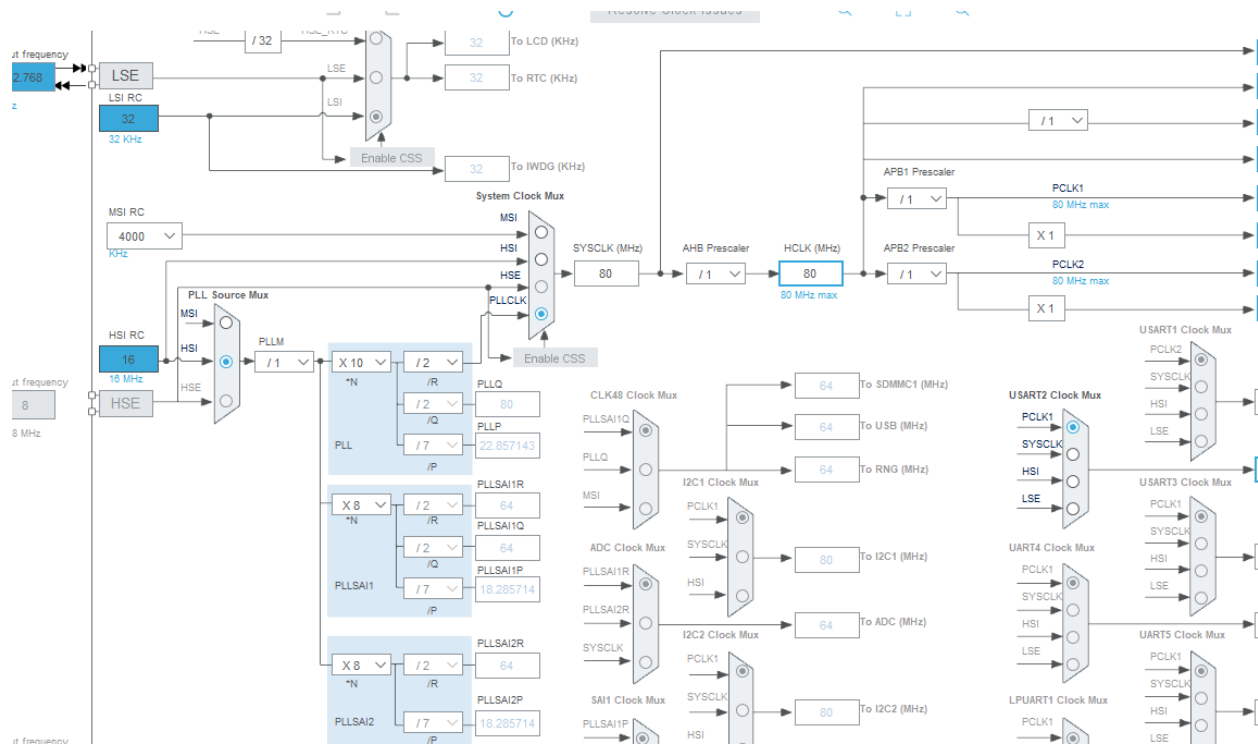
Next, verify under *System Core->GPIO* that PA5 and PC13 user labels are set to LD2 [green Led] and B1 [Blue PushButton] respectively.



3. CLOCK CONFIGURATION

In the clock configuration tab you can see that STM32CubeMX automatically configures the internal oscillator in the clock system with PLL @80MHz. The HSI is selected as the PLL source and the PLLCLK is selected in the system clock mux.

HCLK is set to 80 MHz.



4: GENERATE CODE

We can now generate code. Click File->Save. You will be asked to generate code, press yes.

Under the project explorer navigate to *Core->Src->main.c*.

5: EDIT main.c

Referring to the UM1884 User manual we can use *HAL_GPIO_ReadPin* to read a specified input port pin.

HAL_GPIO_ReadPin

Function name

GPIO_PinState HAL_GPIO_ReadPin (GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin)

Function description

Read the specified input port pin.

Parameters

- **GPIOx**: where x can be (A..H) to select the GPIO peripheral for STM32L4 family
- **GPIO_Pin**: specifies the port bit to read. This parameter can be any combination of GPIO_Pin_x where x can be (0..15).

Return values

- **The**: input port pin value.


If the input pin is equivalent to 0, we blink the LED with a half second delay.

```
1
2
3  /* Infinite loop */
4  /* USER CODE BEGIN WHILE */
5  while (1)
6  {
7      /* USER CODE END WHILE */
8      if(HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin) == GPIO_PIN_RESET) {
9          while(1) {
10             HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
11             HAL_Delay(500);
12         }
13     }
14     /* USER CODE BEGIN 3 */
15 }
16 /* USER CODE END 3 */
17 }
```

6: BUILD THE PROJECT

Connect your USB cable from the computer to your Nucleo Board. Right click the project from the project explorer and click “Build project” to compile the project.

7: DEBUG THE PROJECT

Click on the Debug toolbar icon to start the debug session. Another way to debug is to *Run->Debug* . 

Click the Resume icon to continue the execution.

Press the blue pushbutton and we can now see the green LED(LD2) toggling on the Nucleo-L476RG board.

