

RGB searchlight (GPIO)

RGB searchlight (GPIO)

1. Software-Hardware
2. Brief principle
 1. Hardware schematic diagram
 2. Physical connection diagram
 3. Control principle
3. Engineering experience
 - Open project
4. Project configuration
 1. New project
 2. chip selection
 3. Project settings
 4. Pin configuration
 5. Clock configuration
 6. Project configuration
 7. Generate code
5. Main functions
 1. User function
 2. HAL library function parsing
6. Program download
 1. Serial port download
 2. ST-Link Download
7. Experimental Phenomenon

This tutorial demonstrates: controlling the onboard RGB searchlight on the development board through the output function of **GPIO**.

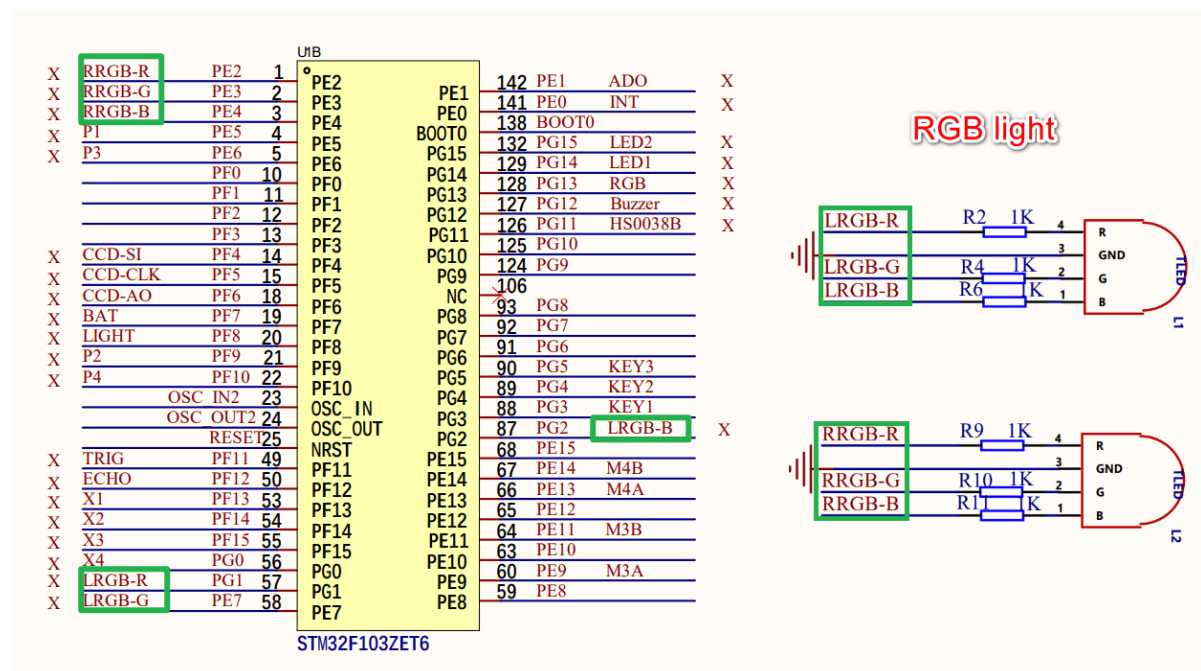
The first tutorial in this chapter will be more detailed than the following tutorials. The purpose is to demonstrate from the new project to the complete effect, and guide users how to use STM32CubeIDE to develop

1. Software-Hardware

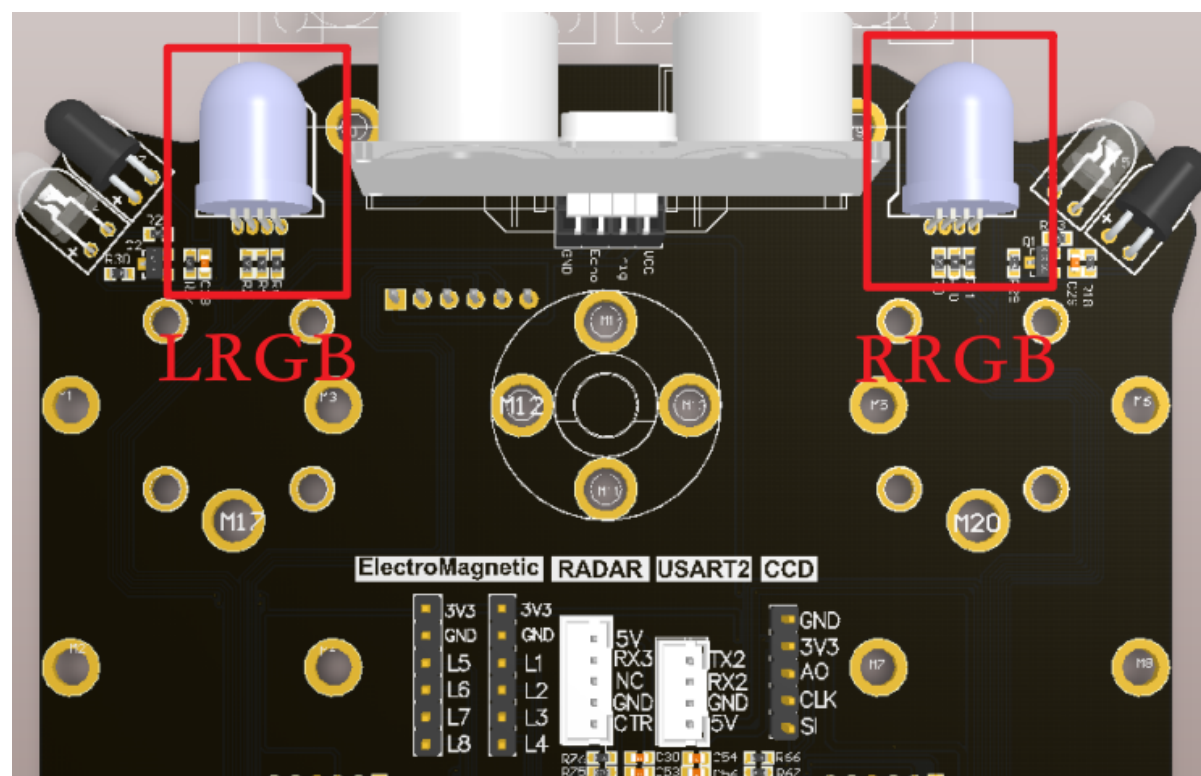
- **STM32F103CubeIDE**
- **STM32 Robot Development Board**
 - GPIO: chip internal peripherals
 - RGB searchlight: onboard
- **Type-C data cable or ST-Link**
 - Download programs or simulate the development board

2. Brief principle

1. Hardware schematic diagram



2. Physical connection diagram



3. Control principle

By controlling the high and low levels of the RGB light pins, the color displayed by the RGB light is controlled.

RGB: High level lights up, low level turns off

RGB（原理图名称）	控制引脚	功能
LRGB-R	PG1	Control the red light display of the left RGB light
LRGB-G	PE7	Control the green light display of the left RGB light
LRGB-B	PG2	Control the blue light display of the left RGB light
RRGB-R	PE2	Control the red light display of the RGB light on the right
RRGB-G	PE3	Control the green light display of the right RGB light
RRGB-B	PE4	Control the blue light display of the right RGB light

RGB lights can display other colors by combining different brightnesses of the three basic colors of red, green, and blue.

3. Engineering experience

You can use the project files we provide to directly experience the corresponding functions of the development board.

Later tutorials do not provide this content to avoid duplication of content. You can go to [2. Development environment construction and use: engineering experience and transplantation] to view the operation

Open project

- **Project file location**

Project file path: Under the [Project Source Code] folder of the Chapter 3 tutorial

名称	类型	大小
 1.RGB	文件夹	

- **Open project file**

Copy the project file to the directory of **English path**, use STM32CubeIDE to open the project file, open the project file and select the **.project** file

名称	类型
.settings	文件夹
Bsp	文件夹
Core	文件夹
Debug	文件夹
Drivers	文件夹
.cproject	CPROJECT 文件
.mxproject	MXPROJECT 文件
.project	PROJECT 文件
RGB Debug.cfg	CFG 文件
RGB Debug.launch	LAUNCH 文件
RGB.ioc	STM32CubeMX
STM32F103ZETX_FLASH.ld	LD 文件

Tip: You can create a new STM32 folder on the desktop and select the STM32 folder as the workspace when opening the project file.

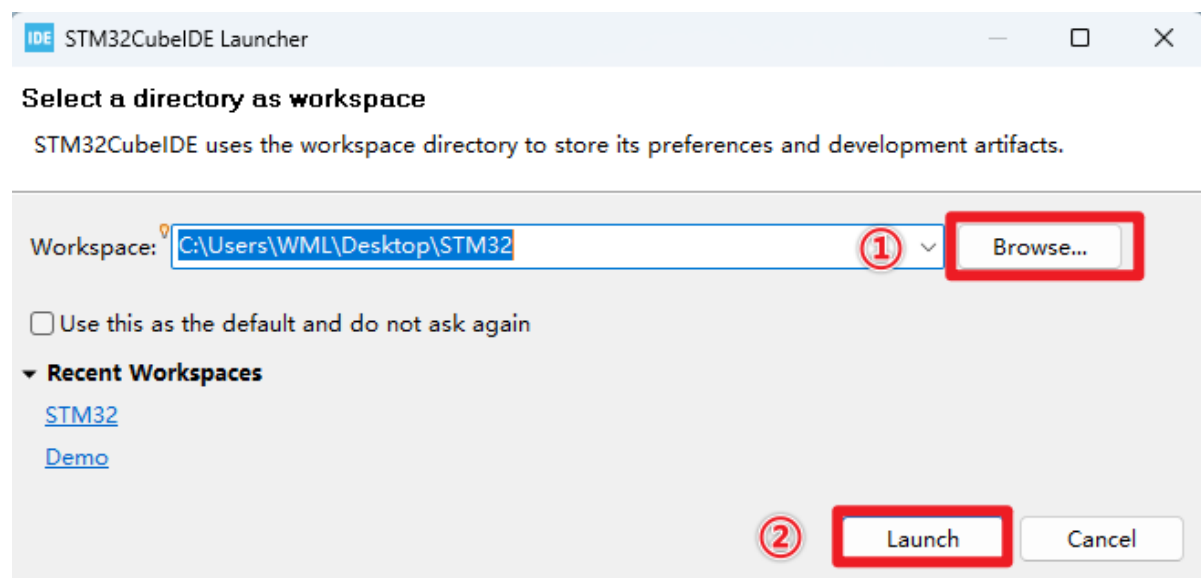
4. Project configuration

This tutorial will completely demonstrate the configuration process. Later, the content of **new project, chip selection, project settings, pin settings of SYS, RCC configuration, clock configuration and project configuration** will be omitted. Any changes will be stated in the tutorial. .

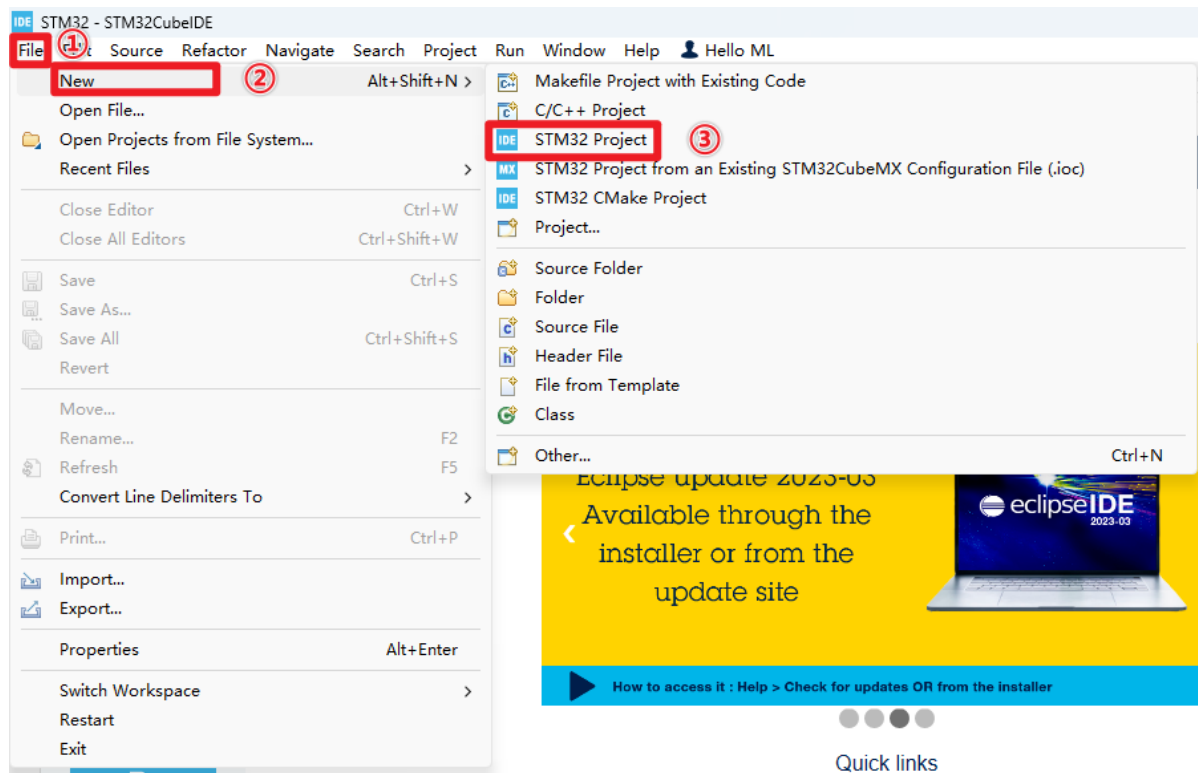
1. New project

- **Select workspace**

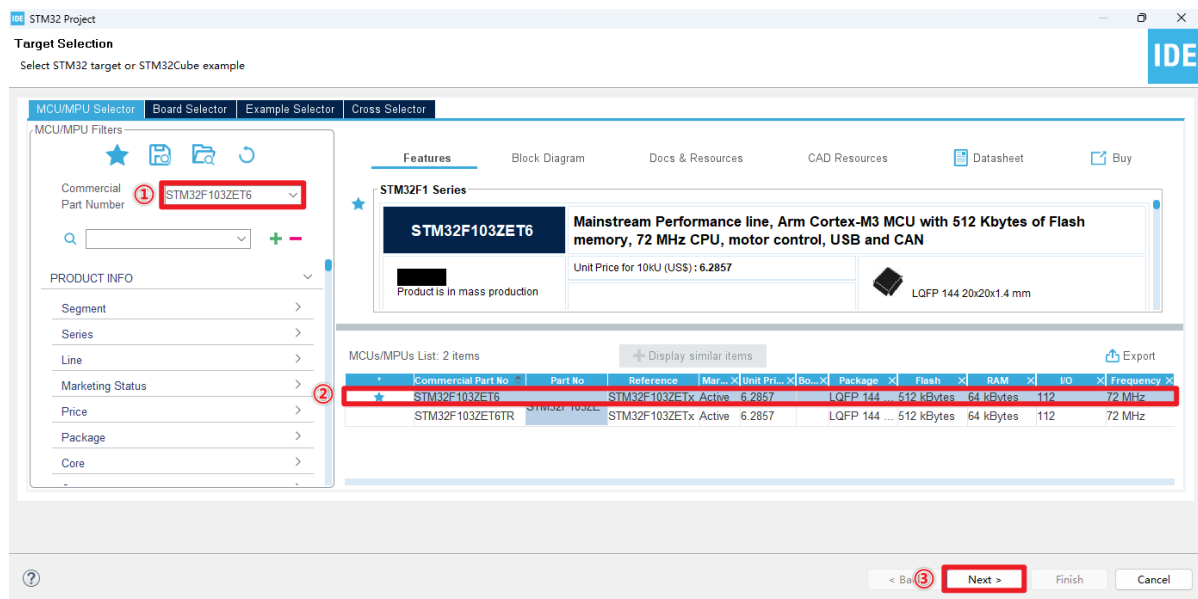
The path of a new project or the path of an existing project file: the path cannot contain Chinese characters



- **New Project**



2. chip selection



3. Project settings

- project name

IDE STM32 Project

Setup STM32 project

IDE

Project

Project Name: RGB ①

☒ Use default location

Location: C:/Users/WML/Desktop/STM32 Browse...

Options

Targeted Language

☒ C ☐ C++

Targeted Binary Type

☒ Executable ☐ Static Library

Targeted Project Type

☒ STM32Cube ☐ Empty

②

< Back Next > ② Finish Cancel

- Firmware version

IDE STM32 Project

Firmware Library Package Setup

Setup STM32 target's firmware

IDE

Target and Firmware Package

Target Reference: STM32F103ZETx

Firmware Package Name and Version: STM32Cube FW_F1 V1.8.5 ①

Firmware and Software Package Repository

Location:
C:\Users\WML\STM32Cube\Repository

See '[Firmware Updater](#)' for settings related to package installation

Code Generator Options

☐ Add necessary library files as reference in the toolchain project configuration file

☐ Copy all used libraries into the project folder

☒ Copy only the necessary library files ②

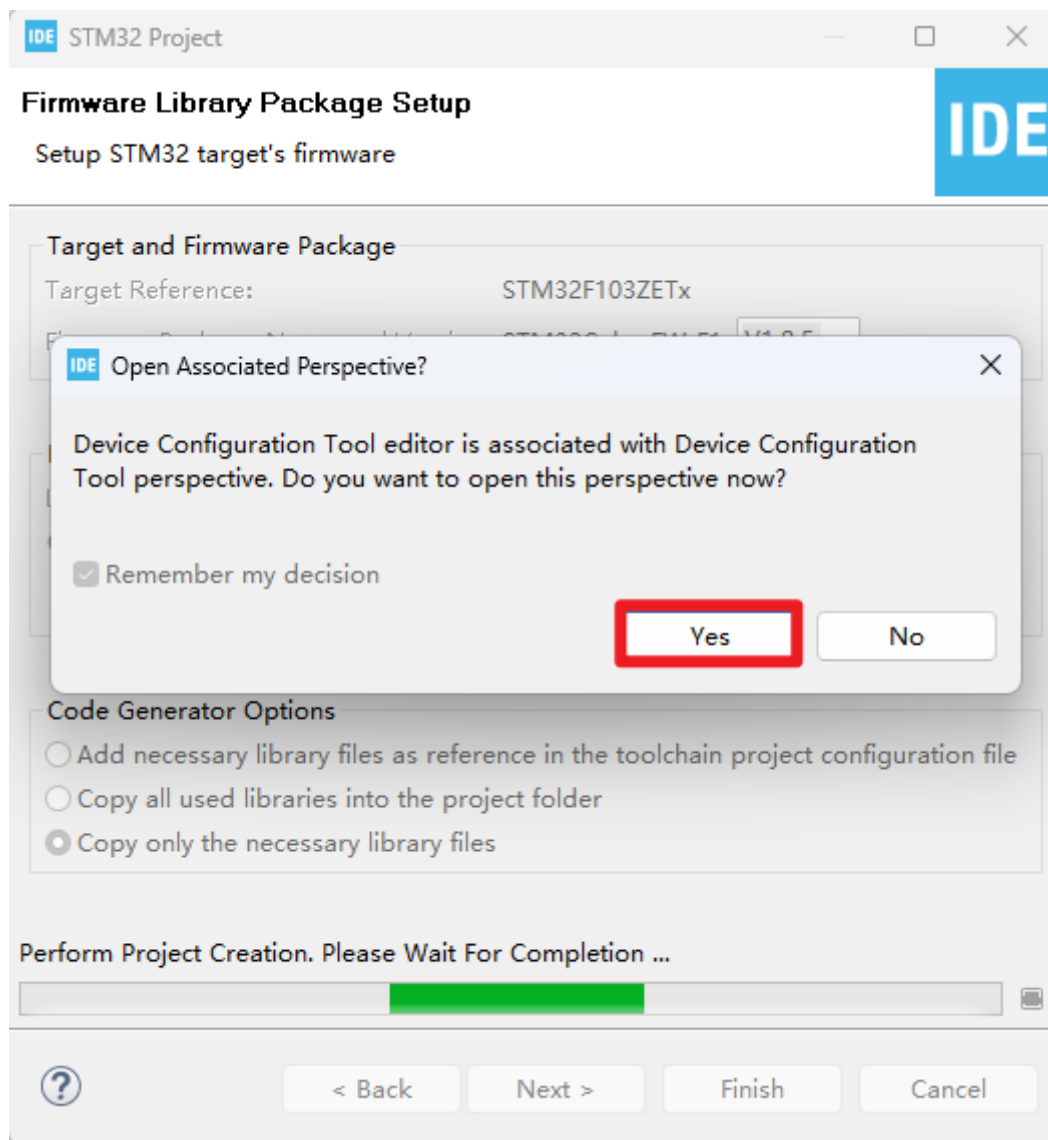
?

< Back

Next > ③

Finish

Cancel



4. Pin configuration

- SYS

RGB.ioc - Pinout & Configuration

Pinout & Configuration

Clock Configuration

Project Manager

Software Packs

Pinout

Search

Categories A-Z

System Core

- DMA
- GPIO
- IWDG
- NVIC
- ✓ RCC
- ⚠ SYS
- WWDG

Analog>

Timers>

Connectivity>

Multimedia>

Computing>

Middleware and Software Packs>

SYS Mode and Configuration

Mode

DebugSerial Wire

☐ System Wake-Up

Timebase SourceSysTick

Configuration

⚠ Warning: This peripheral has no parameters to be configured.

- RCC

RGB.ioc - Pinout & Configuration

Pinout & Configuration

Clock Configuration

Project Manager

Software Packs

Pinout

Search

Categories A-Z

System Core

- DMA
- GPIO
- IWDG
- NVIC
- ✓ RCC
- ⚠ SYS
- WWDG

Analog>

Timers>

Connectivity>

Multimedia>

Computing>

Middleware and Software Packs>

RCC Mode and Configuration

Mode

High Speed Clock (HSE)Crystal/Ceramic Resonator

Low Speed Clock (LSE)Disable

☐ Master Clock Output

Configuration

Reset Configuration

Parameter SettingsUser ConstantsNVIC SettingsGPIO Settings

Configure the below parameters :

Search (Ctrl+F)

<>

i

System Parameters

VDD voltage (V)3.3 V

Prefetch BufferEnabled

Flash Latency(WS)0 WS (1 CPU cycle)

RCC Parameters

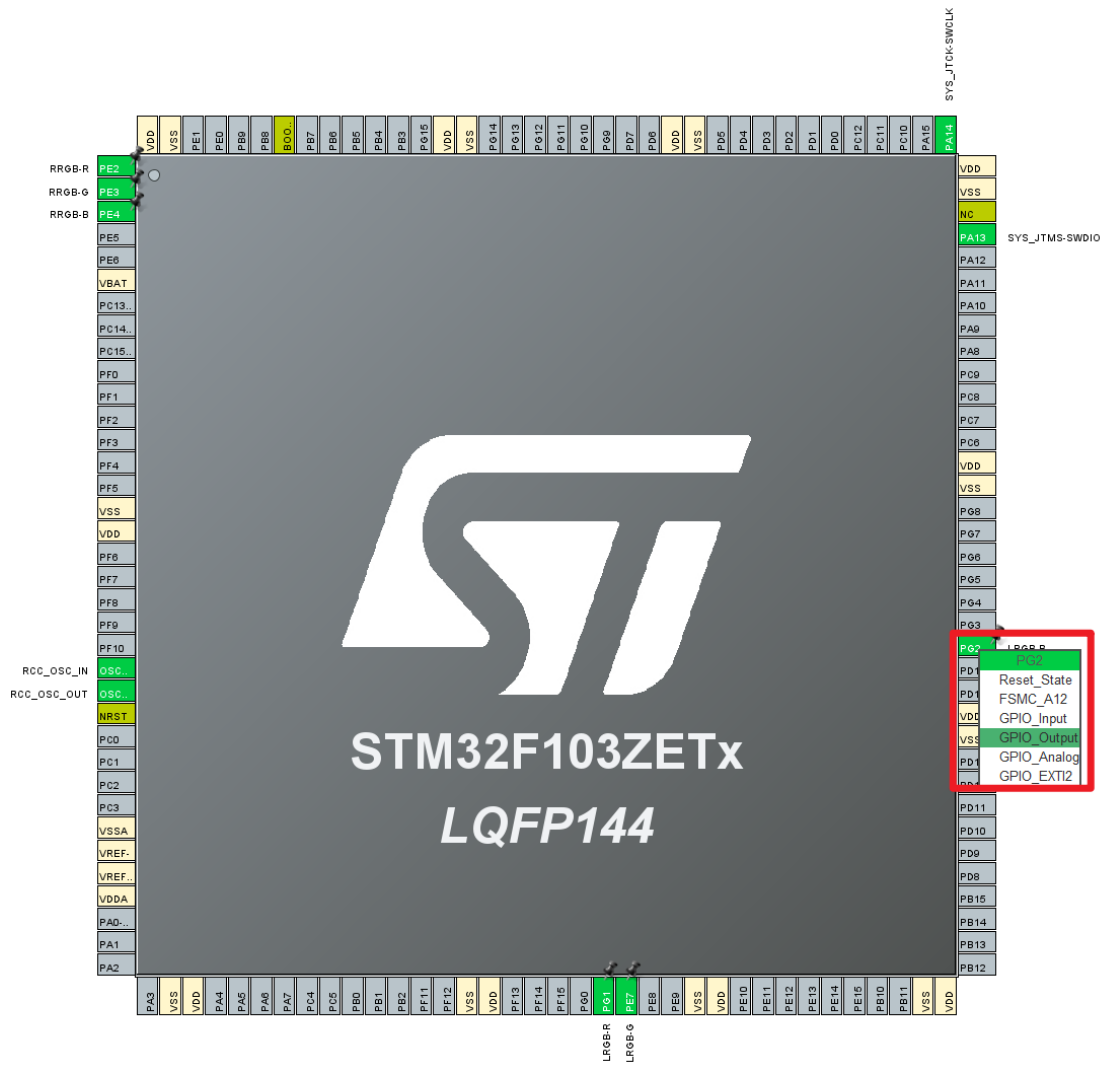
HSI Calibration Value16

HSE Startup Timeout Value (ms)100

LSE Startup Timeout Value (ms)5000

- **Configure specified pin function**

You can directly select the corresponding pin number in the pin view, and the corresponding options will appear when you left-click the mouse.



- **GPIO**

RGB.ioc - Pinout & Configuration

Pinout & Configuration

Clock Configuration

Project Manager

Software Packs

Pinout

GPIO Mode and Configuration

Configuration

Search

Categories

A->Z

System Core

DMA

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Analog

Timers

Connectivity

Multimedia

Computing

Middleware and Software Packs

Group By Peripherals

GPIO

RCC

SYS

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin Name	Signal on Pin	GPIO out...	GPIO mode	GPIO Pull-up/...	Maximum...	Us...	Modified
PE2	/a	Low	Output P...	No pull-up an...	Low	RRGB-R	✓
PE3	/a	Low	Output P...	No pull-up an...	Low	RRGB-G	✓
PE4	/a	Low	Output P...	No pull-up an...	Low	RRGB-B	✓
PG1	/a	Low	Output P...	No pull-up an...	Low	LRGB-R	✓
PE7	/a	Low	Output P...	No pull-up an...	Low	LRGB-G	✓
PG2	/a	Low	Output P...	No pull-up an...	Low	LRGB-B	✓

GPIO output level

Low

GPIO mode

Output Push Pull

GPIO Pull-up/Pull-down

No pull-up and no pull-down

Maximum output speed

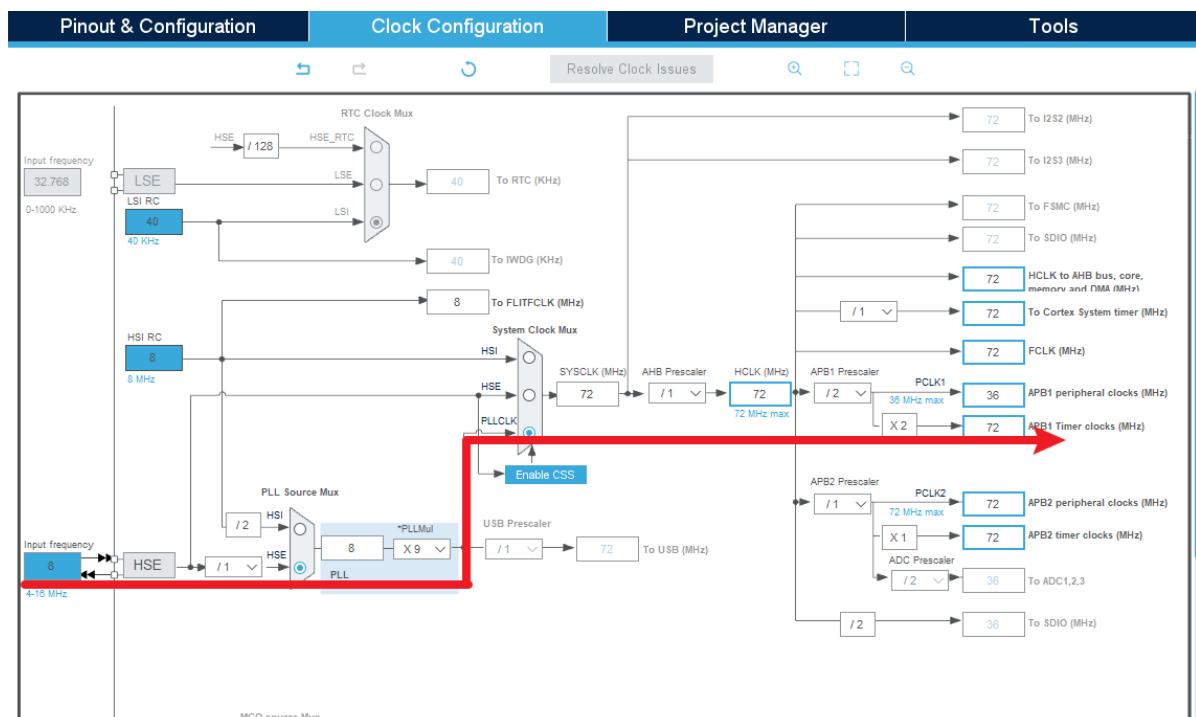
Low

User Label

RRGB-R

Please refer to the figure above for specific configuration options. It is recommended to add tags. STM32CubeIDE will generate corresponding macro definitions.

5. Clock configuration



Refer to the options covered by the red arrows

6. Project configuration

- **Project**

do not need to change

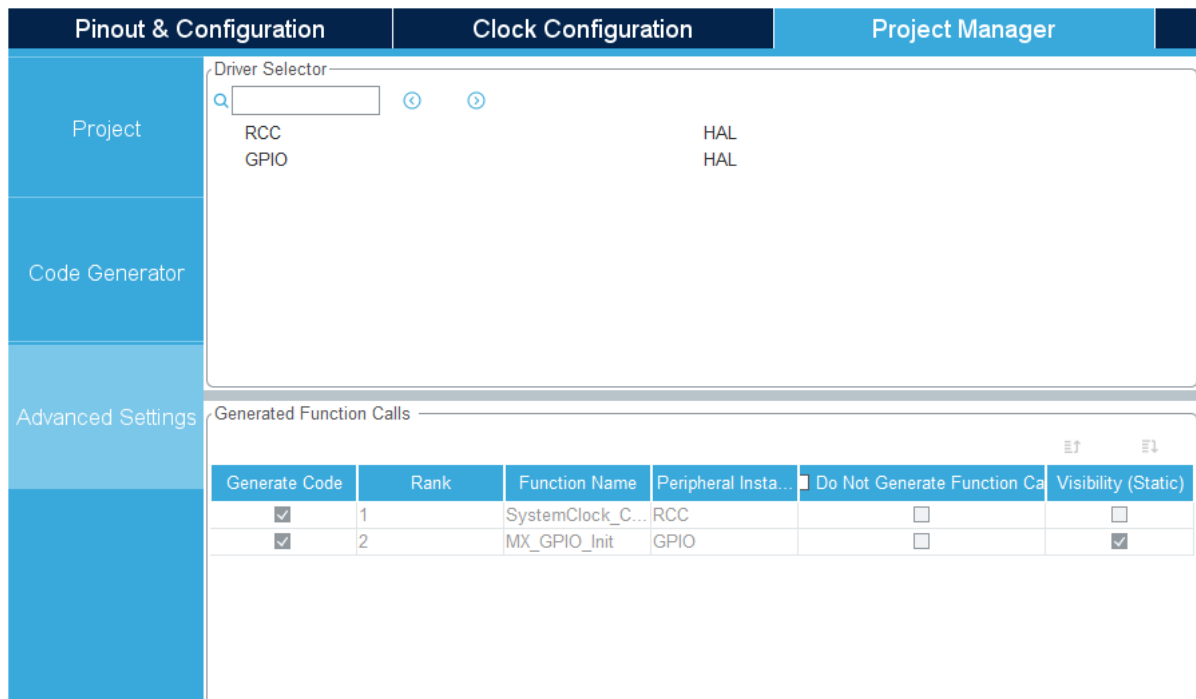
Pinout & Configuration	Clock Configuration	Project Manager	Tools
Project	<div>Project Settings</div> <div>Project Name</div> <div>RGB</div>		
Code Generator	<div>Project Location</div> <div>C:\Users\WML\Desktop\STM32</div> <div>Browse</div>		
	<div>Application Structure</div> <div>Advanced</div> <div><input type="checkbox"/> Do not generate the main()</div>		
	<div>Toolchain Folder Location</div> <div>C:\Users\WML\Desktop\STM32\RGB\</div>		
Advanced Settings	<div>Toolchain / IDE</div> <div>STM32CubeIDE</div> <div><input checked="" type="checkbox"/> Generate Under Root</div>		
	<div>Linker Settings</div>		
	<div>Minimum Heap Size</div> <div>0x200</div>		
	<div>Minimum Stack Size</div> <div>0x400</div>		
	<div>Thread-safe Settings</div>		
	<div>Cortex-M3NS</div>		
	<div><input type="checkbox"/> Enable multi-threaded support</div>		
	<div>Thread-safe Locking Strategy</div> <div>Default - Mapping suitable strategy depending on RTOS selection.</div>		
	<div>Mcu and Firmware Package</div>		
	<div>Mcu Reference</div> <div>STM32F103ZETx</div>		
	<div>Firmware Package Name and Version</div> <div>STM32Cube FW_F1 V1.8.5</div> <div><input checked="" type="checkbox"/> Use latest available version</div>		

- **Code Generator**

Pinout & Configuration	Clock Configuration	Project Manager
Project	<div>STM32Cube MCU packages and embedded software packs</div> <div><input type="radio"/> Copy all used libraries into the project folder</div> <div><input checked="" type="radio"/> Copy only the necessary library files</div> <div><input type="radio"/> Add necessary library files as reference in the toolchain project configuration file</div>	
Code Generator	<div>Generated files</div> <div><input checked="" type="checkbox"/> Generate peripheral initialization as a pair of '.c/.h' files per peripheral</div> <div><input type="checkbox"/> Backup previously generated files when re-generating</div> <div><input checked="" type="checkbox"/> Keep User Code when re-generating</div> <div><input checked="" type="checkbox"/> Delete previously generated files when not re-generated</div>	
Advanced Settings	<div>HAL Settings</div> <div><input type="checkbox"/> Set all free pins as analog (to optimize the power consumption)</div> <div><input type="checkbox"/> Enable Full Assert</div>	
	<div>User Actions</div>	
	<div>Before Code Generation</div>	

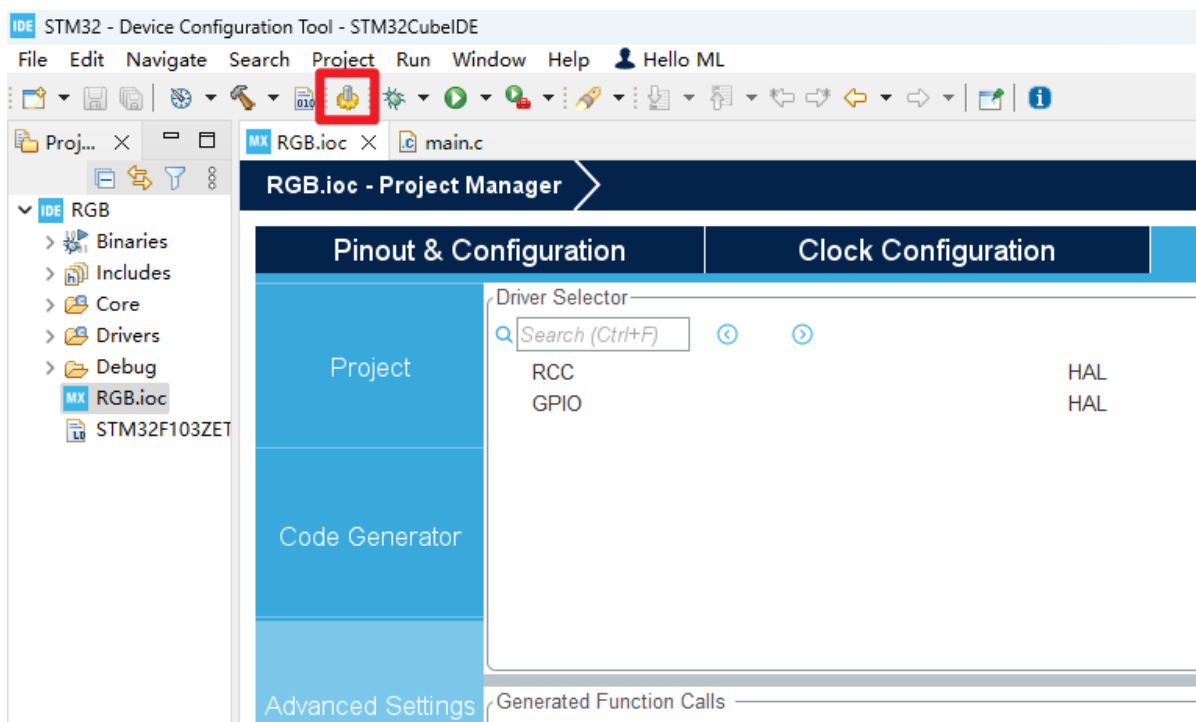
- **Advanced Settings**

do not need to change



7. Generate code

- Click on the "pinion" icon



Click here to save or the Ctrl+C shortcut key to generate code.

- Edit code

The user code must be located between USER CODE BEGIN and USER CODE END.

```

71
72  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
73  HAL_Init();
74
75  /* USER CODE BEGIN Init */
76
77  /* USER CODE END Init */
78
79  /* Configure the system clock */
80  SystemClock_Config();
81
82  /* USER CODE BEGIN SysInit */
83
84  /* USER CODE END SysInit */
85
86  /* Initialize all configured peripherals */
87  MX_GPIO_Init();
88  /* USER CODE BEGIN 2 */
89
90  /* USER CODE END 2 */
91
92  /* Infinite loop */
93  /* USER CODE BEGIN WHILE */
94  while (1)
95  {
96      /* USER CODE END WHILE */
97
98      /* USER CODE BEGIN 3 */
99
100 }

```

The above is the peripheral configuration and initialization code generation.

5. Main functions

It mainly introduces the functional code written by the user. For detailed code, you can open the project file provided by us yourself and enter the Bsp folder to view the source code. **

1. User function

function: **BSP_Init**

function prototype	void BSP_Init(void)
Function description	Initialize the underlying/peripheral driver
Input parameters	none
return value	none

function: **BSP_Loop**

function prototype	void BSP_Loop(void)
Function description	Low-level/peripheral loop functions
Input parameters	none
return value	none

function: **Set_color_R**

function prototype	void Set_color_R(RGB_Color color)
Function description	Demonstration of setting up the RGB light display on the right
Input parameters	color : Set RGB light color
return value	none

function: Set_color_L

function prototype	void Set_color_L(RGB_Color color)
Function description	Demonstration of setting up the RGB light display on the left
Input parameters	color: Set RGB light color
return value	none

function: Set_RGB

function prototype	void Set_RGB(car_RGB light,RGB_Color color)
Function description	Set the color displayed by the left and right RGB lights
Input parameters 1	light: Set the displayed RGB light
Input parameters 2	color: Set RGB light color
return value	none

2. HAL library function parsing

Since using STM32CubeIDE will automatically generate initialization code, only the HAL library functions involved in this tutorial are introduced here.

If you want to find the HAL library and LL library function analysis involved in the entire tutorial, you can view the documents in the folder [8. STM32 Manual: STM32F1_HAL Library and LL Library_User Manual]

function: HAL_GPIO_Init

function prototype	void HAL_GPIO_Init(GPIO_TypeDef *GPIOx, GPIO_InitTypeDef *GPIO_Init)
Function description	Initialize GPIO pin parameters
Input parameters 1	GPIOx: Set the GPIO port, x takes the value A, B, C, D, E, F, G
Input parameters 2	GPIO_Init: GPIO initialization structure
return value	none

function: HAL_GPIO_WritePin

function prototype	void HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)
Function description	Set/Clear the specified data port bit

function prototype	void HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)
Input parameters 1	GPIOx: Set the GPIO port, x takes the value A, B, C, D, E, F, G
Input parameters 2	GPIO_Pin: Set GPIO pin, x value is 0-15
Input parameters 3	PinState: Bit_RESET: clear the data port bit (low level); Bit_SET: set the data port bit (high level)
return value	none

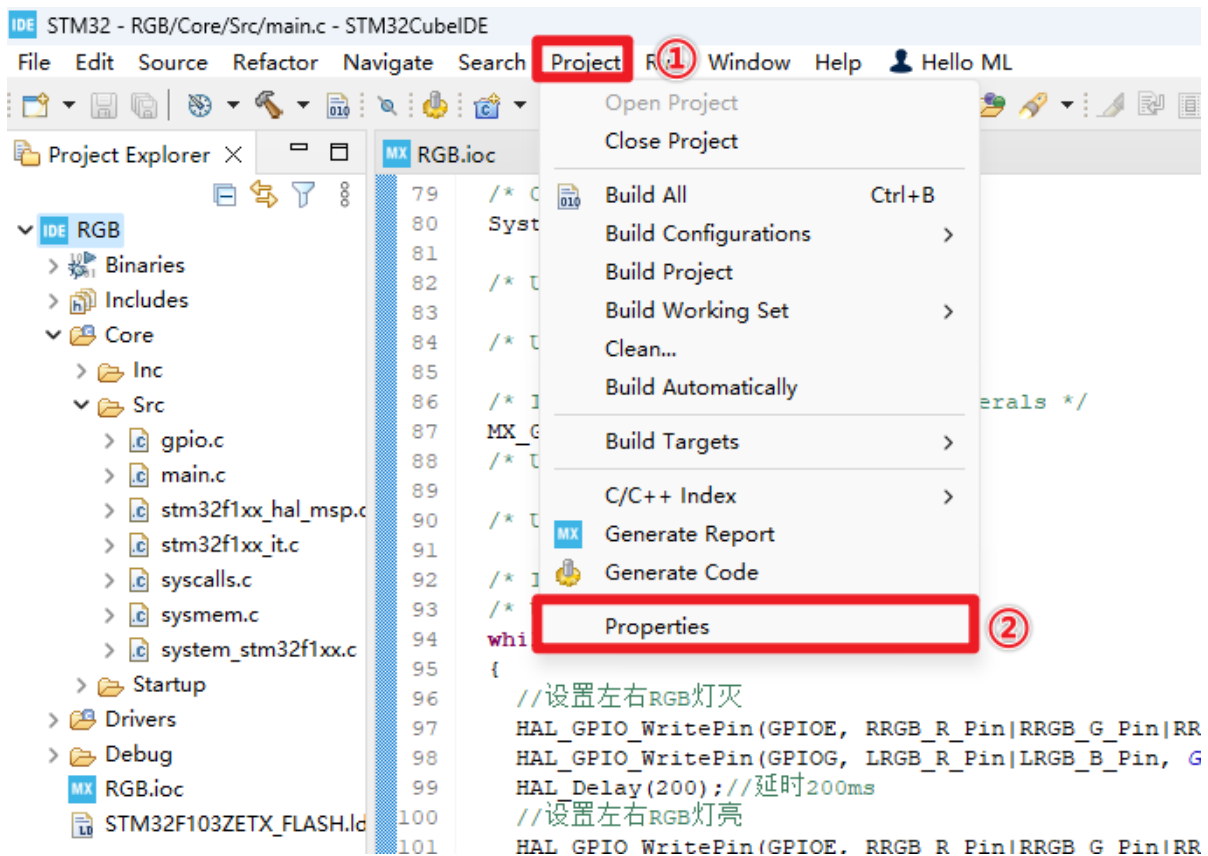
function: HAL_Delay

function prototype	void HAL_Delay(uint32_t Delay)
Function description	Delay for a certain period of time
Input parameters	Delay: Set the delay in milliseconds
return value	none

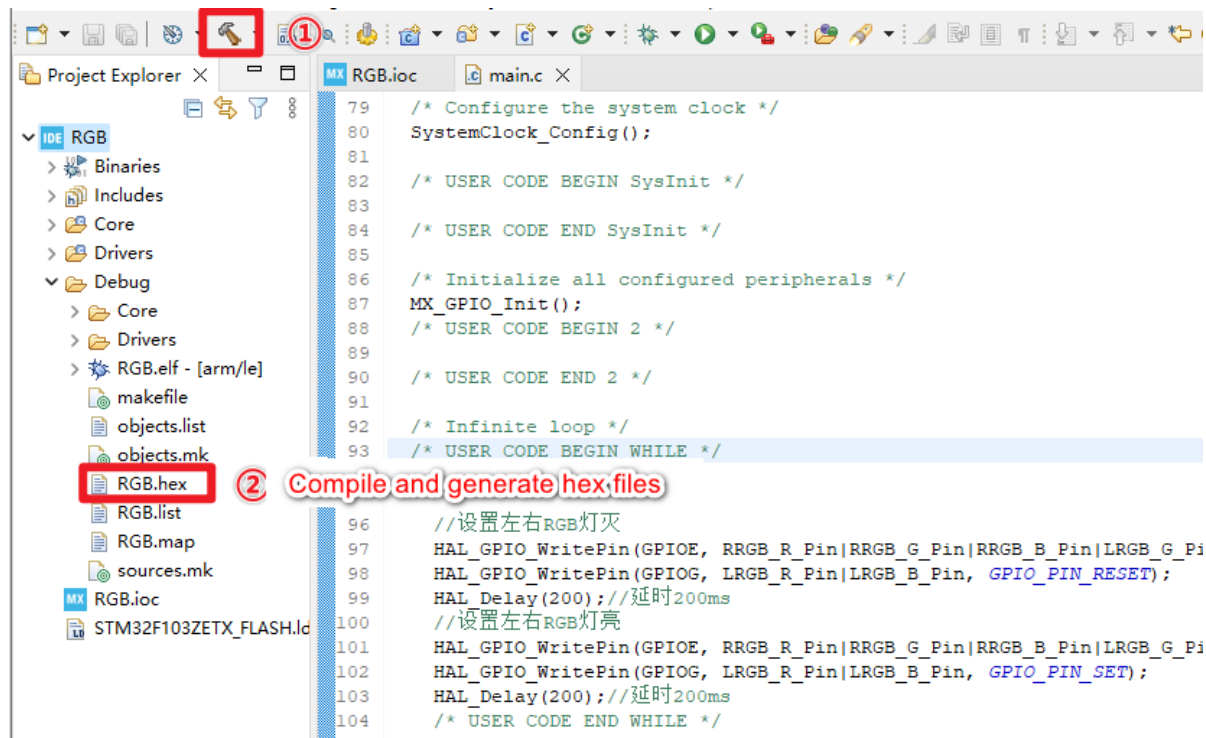
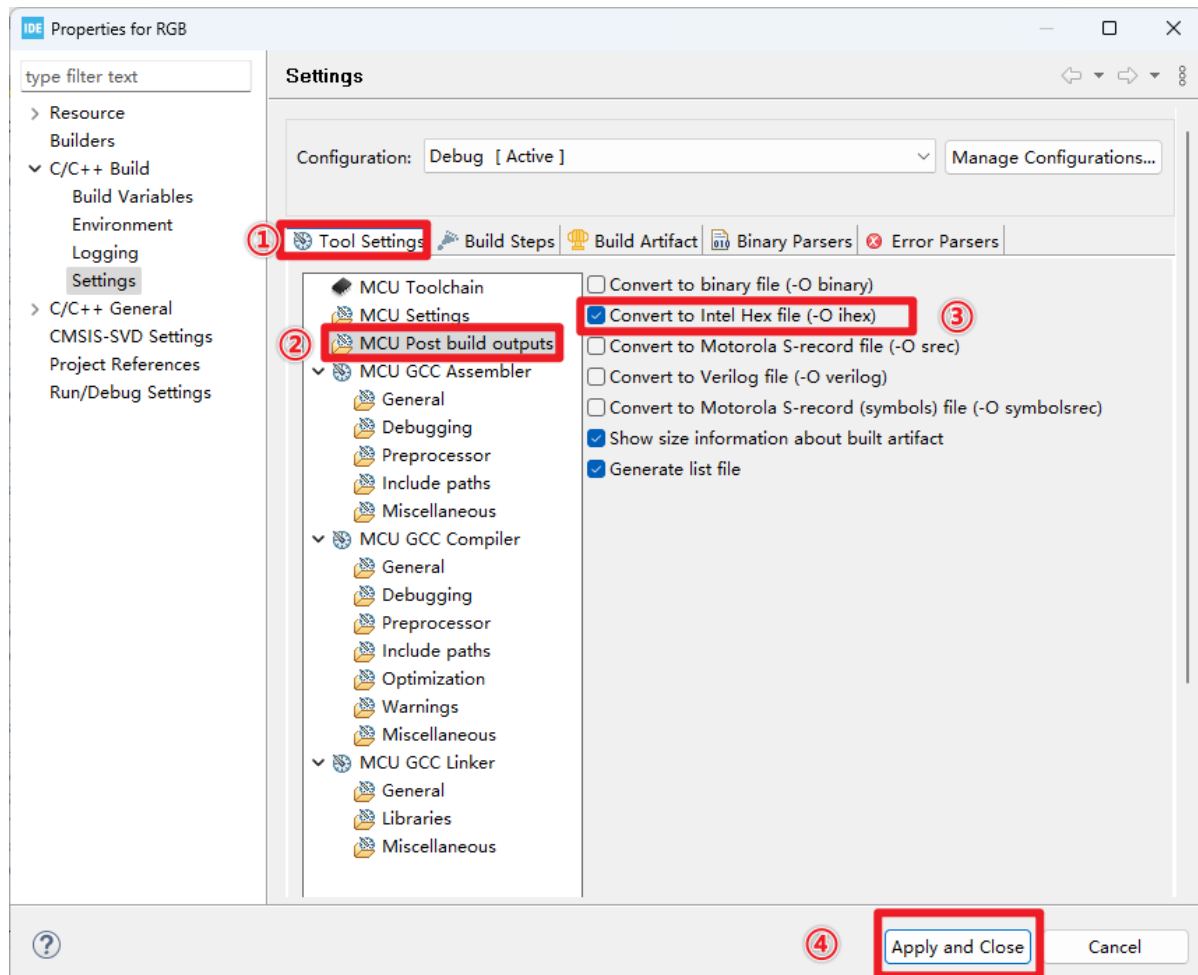
6. Program download

1. Serial port download

- **Generate .hex file:** Left-click the project → select "Properties"



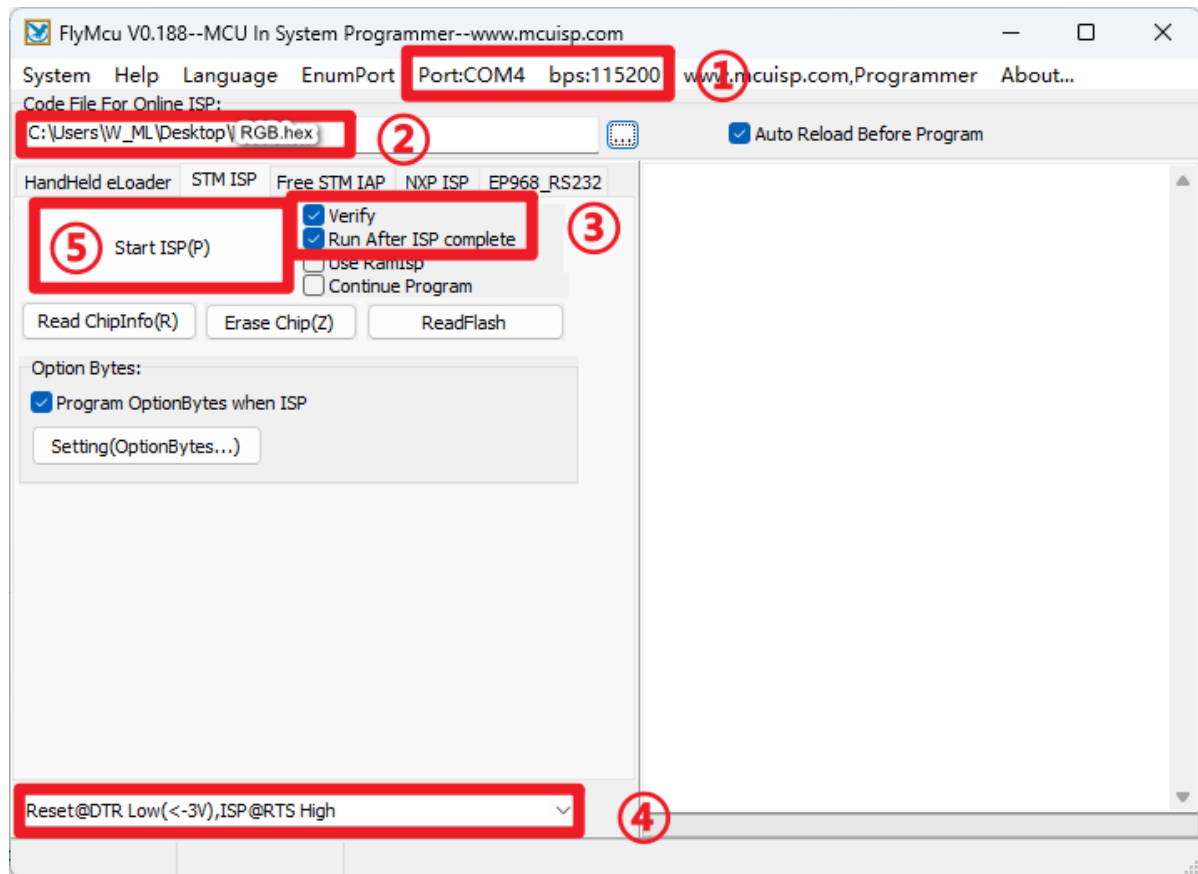
- **Check the corresponding option**



hex file location: under the Debug folder of the project file

Program Download

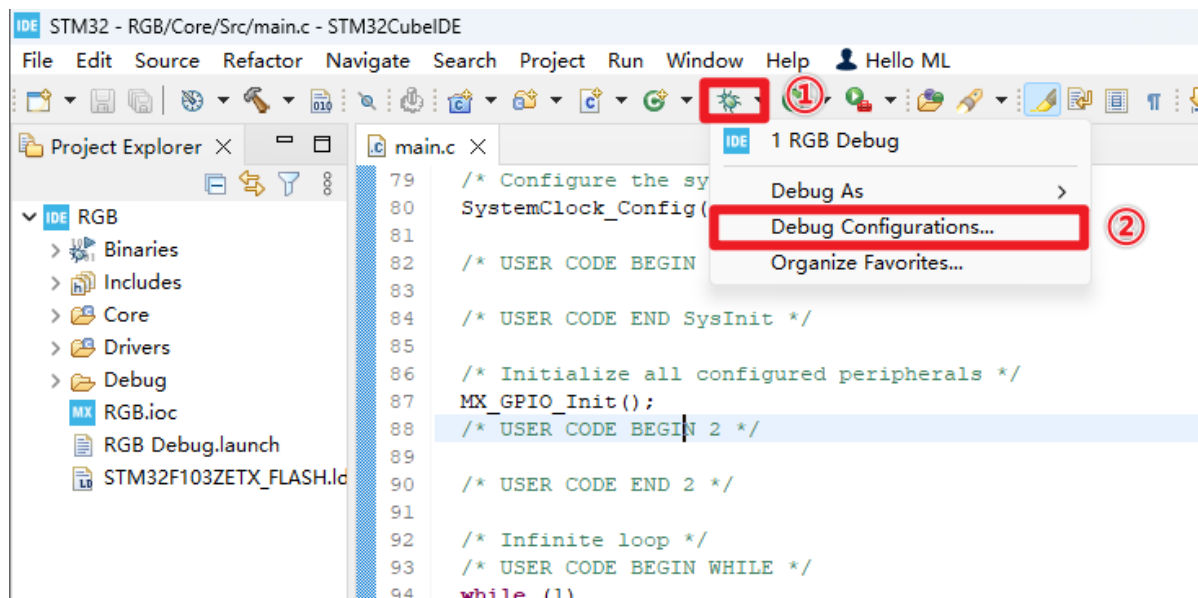
Use the Type-C data cable to connect the development board and computer. For more detailed programming procedures, please refer to [2. Development environment construction and use: program download and simulation]

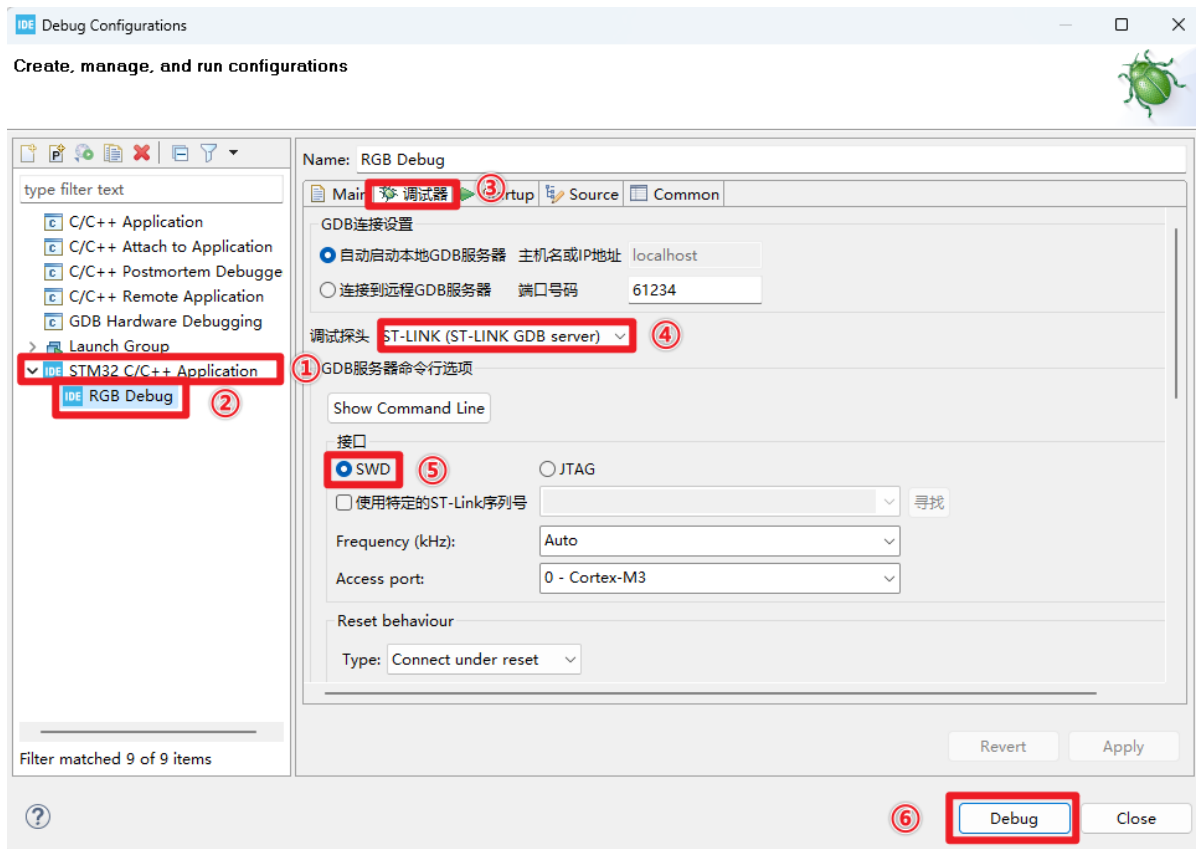


Pay attention to the contents selected in the red box, which must be consistent

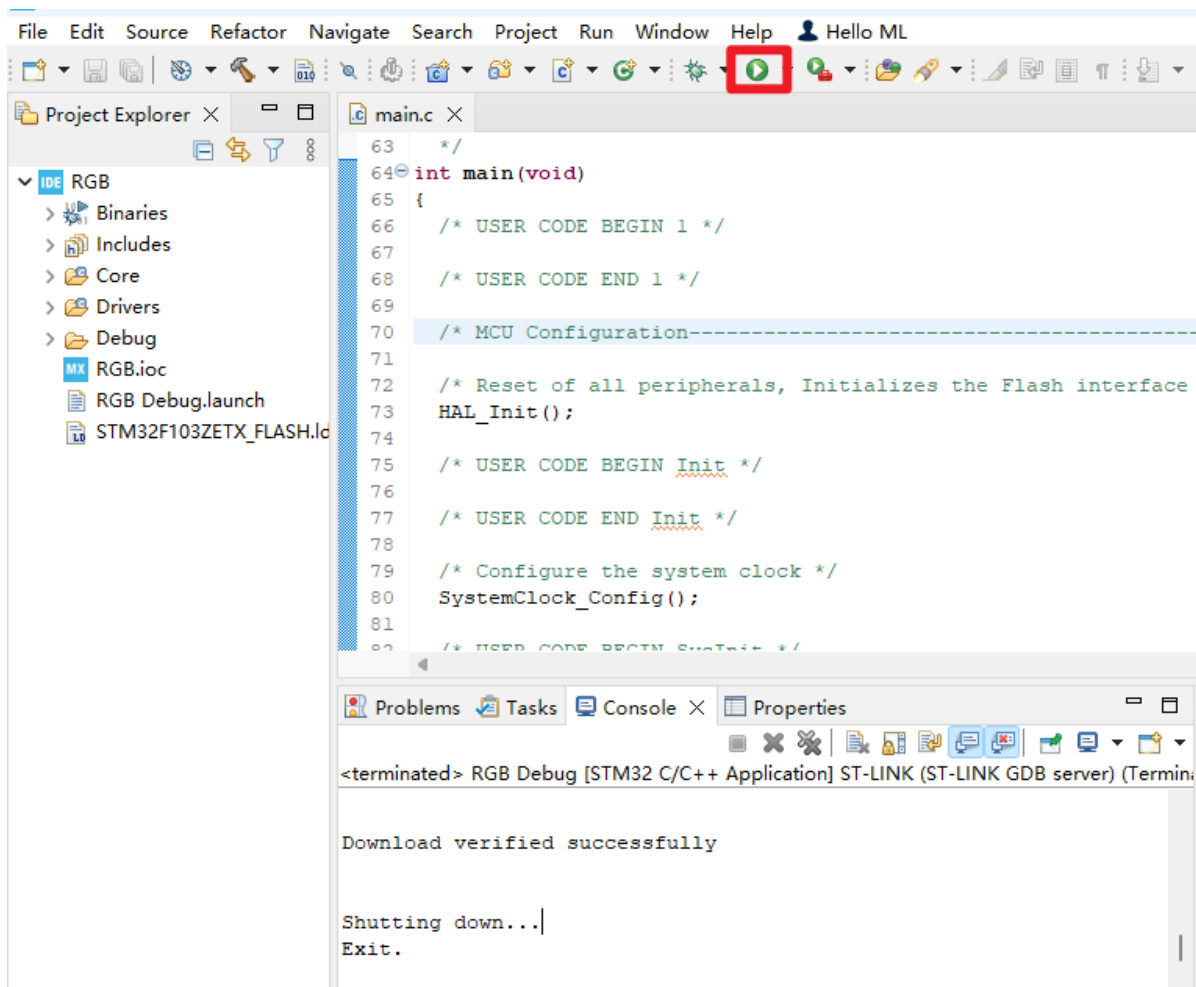
2. ST-Link Download

- **Debug Settings:** ST-Link → SWD





- **Program Download:** Click the Debug icon



7. Experimental Phenomenon

After downloading the program successfully, press the RESET button of the development board and observe the development board phenomenon!

For program download, please refer to [2. Development environment construction and use: program download and simulation]

Phenomenon:

The RGB light on the right turns red for 0.2 seconds and turns off for 0.2 seconds;

The RGB light on the left turns green for 0.2 seconds and turns off for 0.2 seconds;

The RGB lights on the left and right sides turn purple for 0.2 seconds and turn off for 0.2 seconds;

You can see the experimental phenomenon [RGB Searchlight_Experimental Phenomenon.mp4]