

Universal timers

Universal timers

- 1、software-hardware
- 2、Brief principle
 - 2.1、Hardware schematic diagram
 - 2.2、Physical connection diagram
 - 2.3、Principle of control
- 3、Engineering configuration
 - 3.1、Notes
 - 3.2、Pin configuration
- 4、Main function
 - Function analysis
- 5、Experimental phenomenon

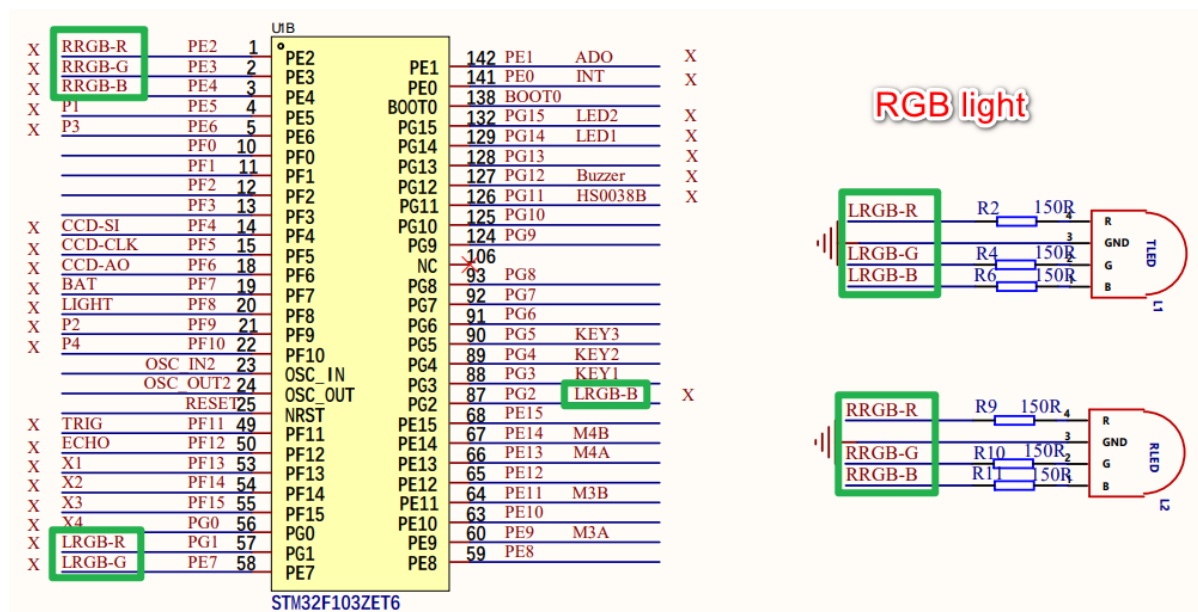
This tutorial demonstrates how to control an onboard RGB breathing light on a development board using a **Universal Timer (TIM3)**.

1、software-hardware

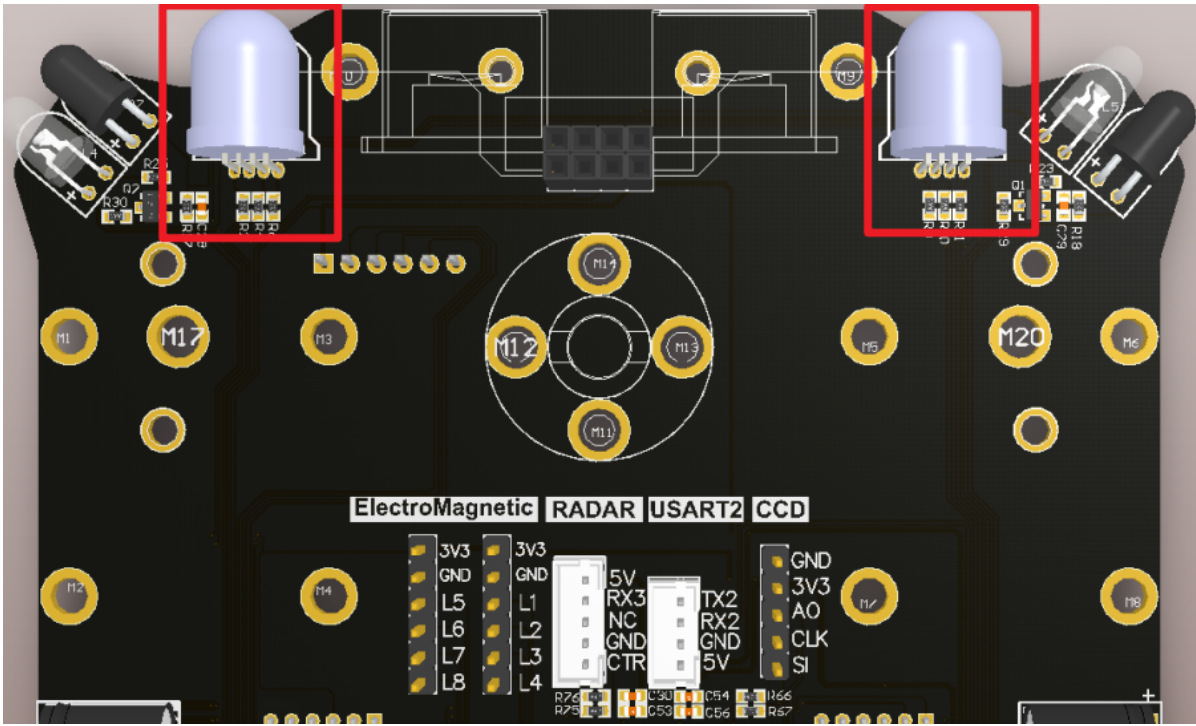
- **STM32F103CubeIDE**
- **STM32 robot extension board**
 - TIM: Chip internal peripherals
 - RGB: Onboard
- **Type-C cable or ST-Link**
 - Download or simulate the program of the development board

2、Brief principle

2.1、Hardware schematic diagram



2.2、 Physical connection diagram



2.3、 Principle of control

By changing the duty cycle of the PWM signal, the Angle of rotation of the servo is controlled

- **PWM (Pulse Width Modulation)**

PWM, short for Pulse width modulation, is a technique that controls the level by adjusting the pulse width of a signal.

Period: the duration of a complete PWM waveform;

Duty cycle: The ratio of the duration of the high level to the cycle time;

Frequency: The inverse of the period is called frequency, which is the number of PWM cycles generated per second;

- **Universal timers**

The timing counting function of TIM3 on the STM32F103ZET6 development board is used

- **The Basics**

Timer types	Universal timers
The timer name	TIM2、TIM3、TIM4、TIM5
Number of counter bits	16
Counting mode	Increase/decrease/center alignment
Predivision coefficient	1-65536
Generating DMA requests	Yes
Capture/compare channels	4

Timer types	Universal timers
Complementary output	No
Clock frequency	72MHz (max)
Mount bus	APB1

Time base cell

register	Function
The counter register (TIMx_CNT)	The current value of the counter
Prescaler register (TIMx_PSC)	Set frequency division coefficient (1-65536)
Automatically reload registers (TIMx_ARR)	The counter counts the boundary and the overloaded value

Timing formula

$$T(s) = \frac{(ARR + 1) * (PSC + 1)}{TIM_CLK(Hz)}$$

parameter	meaning
T (s)	Timing time in seconds
ARR	Automatically reload the value
PSC	Predivision coefficient
TIM_CLK	The timer ticks in Hz

Timing time for the project: 10us

$$T(s) = \frac{(ARR + 1) * (PSC + 1)}{TIM_CLK(Hz)} = \frac{(9 + 1) * (71 + 1)}{72000000(Hz)} = 0.00001s = 10us$$

3、Engineering configuration

Project Configuration: Prompt for configuration options during STM32CubeIDE project configuration

3.1、Notes

Omitted project configuration: **New project, chip selection, project configuration, SYS for pin configuration, RCC configuration, clock configuration, and project configuration** content

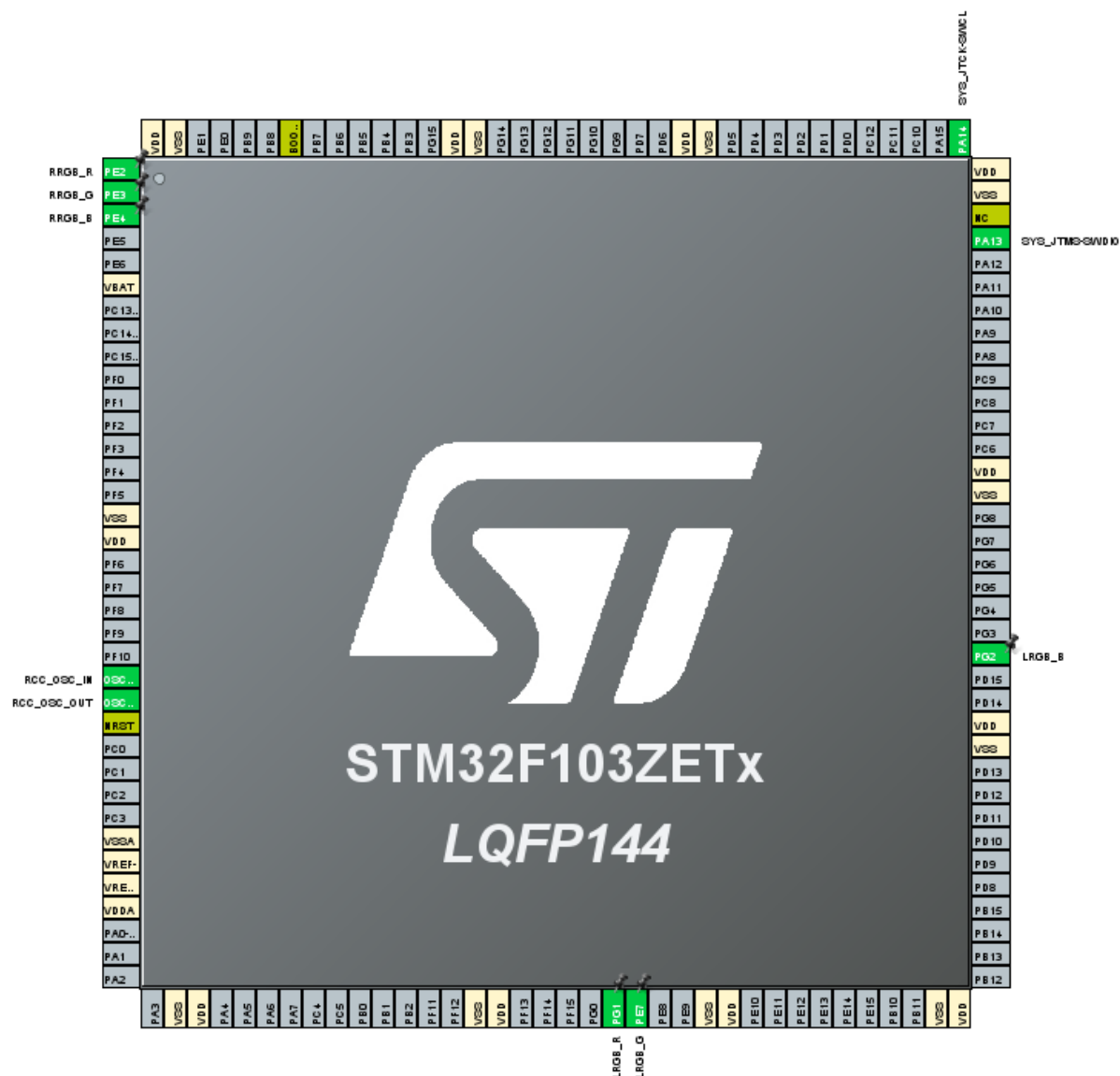
The project configuration part, which is not omitted, is the key point to configure in this tutorial.

Please refer to [2. Development environment construction and use: STM32CubeIDE installation and use] to understand how to configure the omitted parts of the project.

3.2、 Pin configuration

- **Configure the specified pin functionality**

You can directly select the corresponding pin number in the pin view, and the corresponding option will appear when the mouse is left clicked



- **GPIO**

Pinout & Configuration

Clock Configuration

Project Manager

Software Packs

Pinout

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO

RCC

SYS

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin No...	Signal o...	GPIO out...	GPIO mode	GPIO Pull-up/Pull-down	Maximum...	User Label	Modified
PE2	n/a	Low	Output Push Pull	No pull-up and no pull-down	High	RRGB_R	✓
PE3	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	RRGB_G	✓
PE4	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	RRGB_B	✓
PE7	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LRGB_G	✓
PG1	n/a	Low	Output Push Pull	No pull-up and no pull-down	High	LRGB_R	✓
PG2	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LRGB_B	✓

PE2 Configuration :

GPIO output level

Low

GPIO mode

Output Push Pull

GPIO Pull-up/Pull-down

No pull-up and no pull-down

Maximum output speed

High

User Label

RRGB_R

• Timers

Pinout & Configuration

Clock Configuration

Software Packs

Pinout

TIM3 Mode and Configuration

Mode

Slave Mode

Disable

Trigger Source

Disable

Clock Source

Internal Clock

Channel1

Disable

Channel2

Disable

Channel3

Disable

Channel4

Disable

Combined Channels

Disable

Use ETR as Clearing Source

XOR activation

One Pulse Mode

Configuration

Reset Configuration

Parameter Settings

User Constants

NVIC Settings

DMA Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - 16 bits value)

Counter Mode

Counter Period (AutoReload Register - 16 bits value)

Internal Clock Division (CKD)

auto-reload preload

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Trigger Event Selection

72-1

Up

10-1

No Division

Enable

Disable (Trigger input effect not delayed)

Reset (UG bit from TIMx_EGR)

• NVIC

The NVIC option allows you to change the priority

Pinout & Configuration

Clock Configuration

Project Manager

Toolbox

Software Packs

Pinout

Search

Categories

System Core

DMA

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Analog

Timers

Connectivity

Multimedia

Computing

Middleware and Software P...

NVIC Mode and Configuration

Configuration

NVIC

Code generation

Priority Group

Sort by Preemption Priority and Sub Priority

Sort by interrupts names

Search

Force DMA channels Interrupts

NVIC Interrupt Table

Enabled

Preemption Priority

Sub Priority

Non maskable interrupt

Hard fault interrupt

Memory management fault

Prefetch fault, memory access fault

Undefined instruction or illegal state

System service call via SWI instruction

Debug monitor

Pendable request for system service

Time base: System tick timer

PVD interrupt through EXTI line 16

Flash global interrupt

RCC global interrupt

TIM3 global interrupt

• Advanced Settings

Pinout & Configuration

Clock Configuration

Project Manager

Toolbox

Project

Code Generator

Advanced Settings

Driver Selector

RCC

GPIO

TIM

TIM3

HAL

HAL

HAL

HAL

Generated Function Calls

Generate Code

Rank

Function Name

Peripheral Instance Na...

Do Not Generate Function Call

Visibility (Static)

SystemClock_Config

RCC

MX_GPIO_Init

GPIO

MX_TIM3_Init

TIM3

• Generate code

Generate Code

4、 Main function

This section mainly introduces the functional code written by users. **Detailed code can be opened by yourself in the project file we provide, and enter the Bsp folder to view the source code.**

Function analysis

The HAL library functions that were covered in the previous tutorial will not be covered

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]

HAL_TIM_PeriodElapsedCallback: The timer update interrupt callback function

Function prototypes	void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
Functional Description	The timer update interrupt callback function
Input parameters	htim : Timer handle address
Return value	None
Notes	This function is called by HAL_TIM_IRQHandler, which allows you to write specific tasks

Each time the timer interrupt callback is entered, the timer's pwm_val variable is incremented by 1 and compared against the pwm_set variable in the main loop.

If $\text{pwm_val} < \text{pwm_set}$, RGB lights off; If $\text{pwm_val} > \text{pwm_set}$, RGB light is on.

Main loop function

In the main loop function, the comparison value of the interrupt callback function is changed by controlling the increment and decrement of the pwm_set variable to achieve different duty cycles per unit time.

The above two functions can control the PWM duty cycle of RGB to achieve the gradual effect of light.

5、 Experimental phenomenon

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

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

phenomenon:

left and right RGB lights:display the purple color, which goes from bright to dark, and from dark to bright.

