

Advanced timers

Advanced timers

- 1、software-hardware
- 2、Brief principle
 - 2.1、Hardware interface
 - 2.2、Physical connection diagram
 - 2.3、Principle of control
- 3、Engineering configuration
 - 3.1、Notes
 - 3.2、Pin configuration
- 4、Main function
 - 4.1、User functions
 - 4.2、HAL library functions
- 5、Experimental phenomenon

This tutorial shows how to generate a PWM signal with a known frequency and duty cycle from the **Universal Timer (TIM3)**, measure the frequency and duty cycle of this known PWM signal using the PWM input mode of the **** Advanced Timer (TIM1)**, and **print the relevant data via the serial port (USART1)****.

1、software-hardware

- **STM32F103CubeIDE**

- **STM32 robot expansion board**

TIM: Chip internal peripherals

Mother to mother dupont wire: 1

- **Type-C cable or ST-Link**

Download or simulate the program of the development board

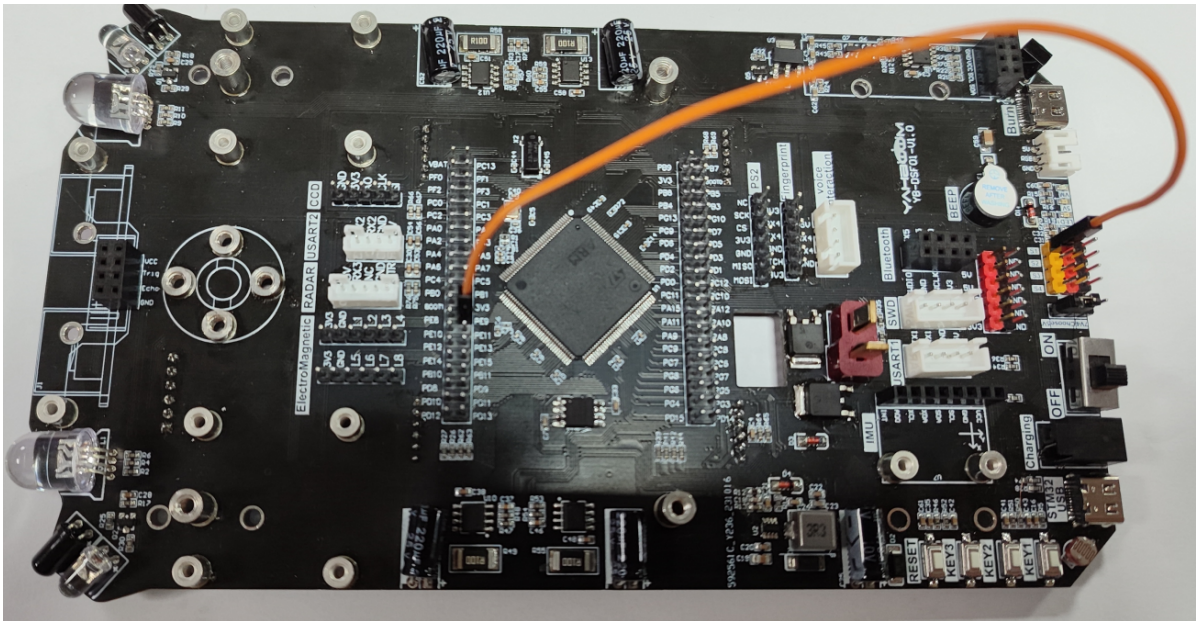
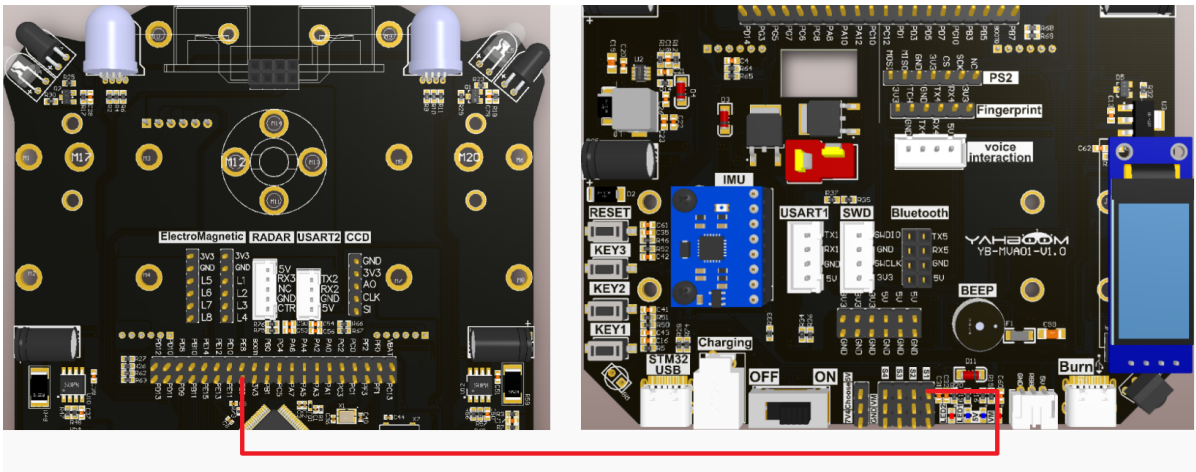
2、Brief principle

2.1、Hardware interface

LIB										LUA									
X	RRGB-R	PE2	1	PE2	PE1	142	PE1	ADO	X	PC13	7	PC13-TAMPER-RTC	PB9	140	PB9				
X	RRGB-G	PE3	2	PE3	PE0	141	PE0	INT	X	OSC_IN1	PC14	8	PC14-OSC32_IN	PB8	139	PB8	RGB	X	
X	RRGB-B	PE4	3	PE4	BOOT0	138	BOOT0			OSC_OUT1	PC15	9	PC15-OSC32_OUT	PB7	137	PB7	I2C1_SDA		
X	P1	PE5	4	PE5	PG15	132	PG15	LED2	X	ADC-L1	PC0	26	PC0	PB6	136	PB6	I2C1_SCL		
X	P3	PE6	5	PE6	PG14	129	PG14	LED1	X	ADC-L2	PC1	27	PC1	PB5	135	PB5	H4B		
		PF0	10	PF0	PG13	128	PG13		X	ADC-L3	PC2	28	PC2	PB4	134	PB4	H4A		
		PF1	11	PF1	PG12	127	PG12	Buzzer	X	ADC-L4	PC3	29	PC3	PB3	133	PB3	H2B		
		PF2	12	PF2	PG11	126	PG11	HS0038B	X	H3A	PA0	35	PA0-WKUP	PB2	132	PB2	TCH		
		PF3	13	PF3	PG10	125	PG10			H3B	PA1	36	PA1	PB1	131	PB1	USART2_RX		
X	CCD-SI	PF4	14	PF4	PG9	124	PG9		X	ADC-L5	PA2	37	PA2	PB0	130	PB0	USART2_TX		
X	CCD-CLK	PF5	15	PF5	PG8	106	PG8			ADC-L6	PA3	38	PA3	PB0	129	PB0			
X	CCD-AO	PF6	18	PF6	PG7	93	PG7			PS2-CS	PA4	40	PA4	PB0	128	PB0			
X	BAT	PF7	19	PF7	PG6	91	PG6			PS2-SCK	PA5	41	PA5	PB0	127	PB0			
X	LIGHT	PF8	20	PF8	PG5	90	PG5	KEY3		PS2-MISO	PA6	42	PA6	PB0	126	PB0			
X	P2	PF9	21	PF9	PG4	89	PG4	KEY2		PS2-MOSI	PA7	43	PA7	PB0	125	PB0			
X	P4	PF10	22	PF10	PG3	88	PG3	KEY1		ADC-L7	PC4	44	PC4	PB0	124	PB0			
		OSC_IN2	23	OSC_IN	PG2	87	PG2	LRGB-B	X	ADC-L8	PC5	45	PC5	PB0	123	PB0			
		OSC_OUT2	24	OSC_OUT	PG1	86	PG1			S1	PB0	46	PB0	PB0	122	PB0			
		RESE	25	NRST	PE15	68	PE15			S2	PB1	47	PB1	PB1	121	PB1			
X	TRIG	PF11	49	PF11	PE14	67	PE14	M4B		BOOT1	PB2	48	PB2	PB2	120	PB2			
X	ECHO	PF12	50	PF12	PE13	66	PE13	M4A		I2C2_SCL	PB10	69	PB10	PB10	119	PB10			
X	X1	PF13	53	PF13	PE12	65	PE12			I2C2_SDA	PB11	70	PB11	PB11	118	PB11			
X	X2	PF14	54	PF14	PE11	64	PE11	M3B		SP12_NSS	PB12	73	PB12	PB12	117	PB12			
X	X3	PF15	55	PF15	PE10	63	PE10			SP12_SCK	PB13	74	PB13	PB13	116	PB13			
X	X4	PG0	56	PG0	PE9	60	PE9	M3A		SP12_MISO	PB14	75	PB14	PB14	115	PB14			
X	LRGB-R	PG1	57	PG1	PE8	59	PE8			SP12_MOSI	PB15	76	PB15	PB15	114	PB15			
X	LRGB-G	PE7	58	PE7						M-CTR	PD8	77	PD8	PB15	113	PB15			
										USART3_RX	PD9	78	PD9	PB15	112	PB15			
											PD10	79	PD10	PB15	111	PB15			
											PD11	80	PD11	PB15	110	PB15			
											PD12	81	PD12	PB15	109	PB15			
											PD13	82	PD13	PB15	108	PB15			
														PB15	107	PB15			
														PB15	106	PB15			
														PB15	105	PB15			
														PB15	104	PB15			
														PB15	103	PB15			
														PB15	102	PB15			
														PB15	101	PB15			
														PB15	100	PB15			
														PB15	99	PB15			
														PB15	98	PB15			
														PB15	97	PB15			
														PB15	96	PB15			
														PB15	95	PB15			
														PB15	94	PB15			
														PB15	93	PB15			
														PB15	92	PB15			
														PB15	91	PB15			
														PB15	90	PB15			
														PB15	89	PB15			
														PB15	88	PB15			
														PB15	87	PB15			
														PB15	86	PB15			
														PB15	85	PB15			

2.2、Physical connection diagram

This tutorial requires connecting the S1 signal wire (yellow terminal) and PE9 pin of the servo



2.3、Principle of control

TIM1: The timer PWM input mode is configured to measure the frequency and duty cycle of the known PWM signal

TIM3: The timer PWM output mode is configured to generate PWM signals with known frequency and duty cycle

- **Advanced timers**

Timer types	Advanced timers
The timer name	TIM1、TIM8
Number of counter bits	16
Counting mode	Increase/decrease/center alignment
Predivision coefficient	1-65536
Generating DMA requests	Yes

Timer types	Advanced timers
Capture/compare channels	4
Complementary output	Yes
Clock frequency	72MHz (Max)
Mount bus	APB2

Time base cell

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

Timing formula

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

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

PWM period: T = 10ms → f = 100Hz

$$T(s) = \frac{(ARR + 1) * (PSC + 1)}{TIM_CLK(Hz)} = \frac{(9999 + 1) * (71 + 1)}{72000000(Hz)} = 0.01s = 10ms$$

- PWM input mode**

When PWM input mode is used, one input channel (TIx) will occupy two capture channels (ICx), and the two ICx signals are valid for edges, but with opposite polarity.

Measure pulse width and frequency

PWM is measured in PWM input mode, PWM signal input can only be input from channel 1 (CH1) or channel 2 (CH2).

Consider input channel TI1 working in PWM input mode:

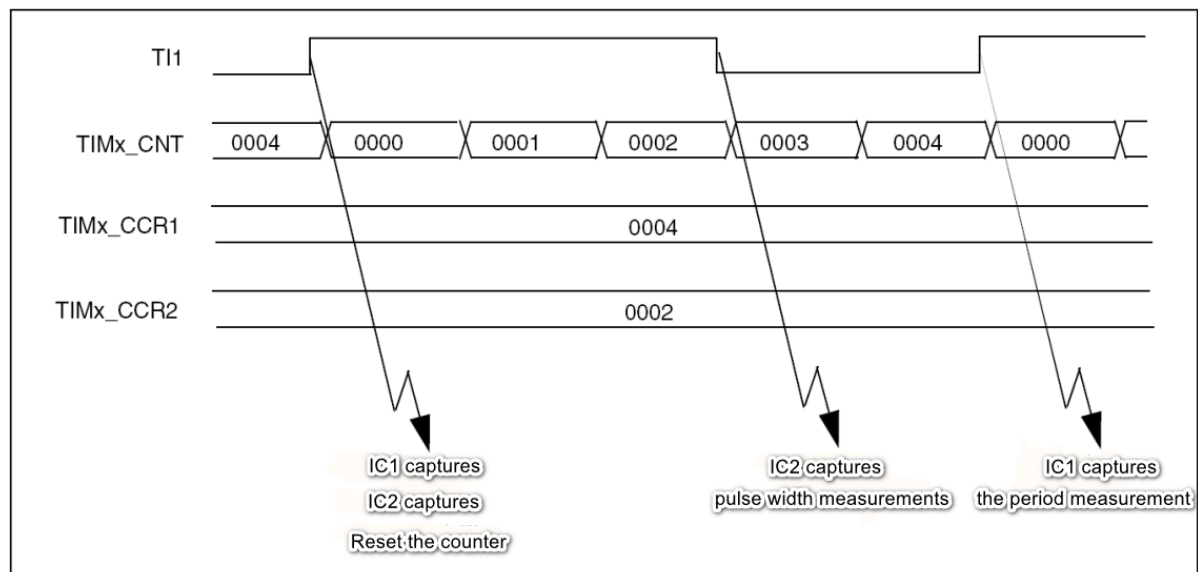
The PWM signal enters through the input channel TI1, and the signal will be divided into two channels, one is TI1FP1, and the other is TI1FP2.

Choose the effective polarity of TI1FP1: effective → period on the rising edge

The effective polarity of TI1FP2 is chosen: effective → duty cycle on the falling edge

When using PWM input mode, the slave mode controller must be configured to reset mode, that is, when we start the trigger signal to start the capture, we also reset the counter CNT to zero.

PWM input mode timing



The PWM signal is entered by the input channel TI1, and TI1FP1 is configured as the trigger signal, and the rising edge is captured. When the rising edge is captured by IC1 and IC2, the counter CNT is reset. When the falling edge is reached, IC2 is captured and the value of counter CNT is latched into the capture register CCR2. When the next rising edge is reached, IC1 is captured and the value of counter CNT is latched into the capture register CCR1. Among them, CCR2 measures pulse width and CCR1 measures period.

3、Engineering 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

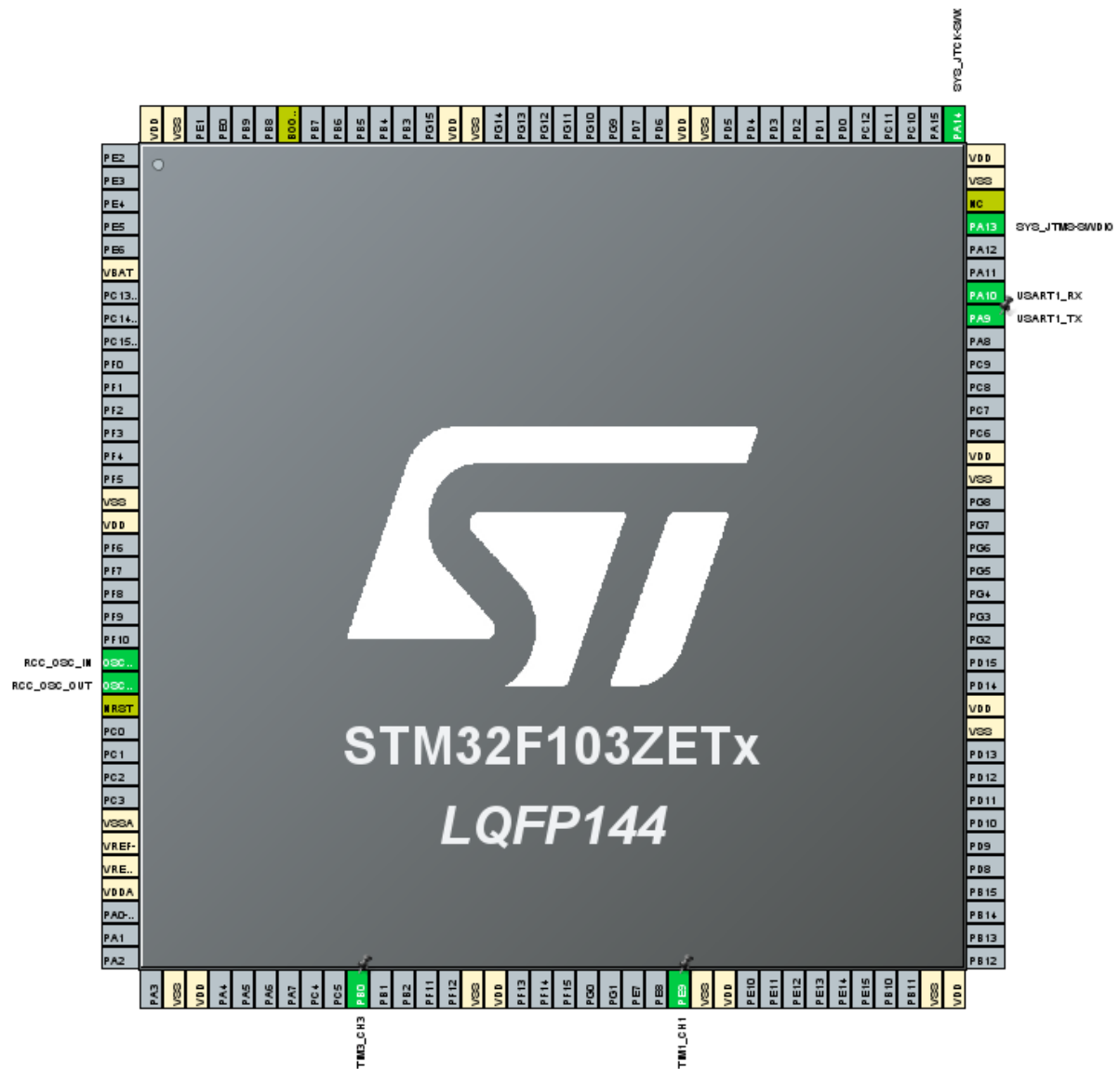
The project configuration part, which is not omitted, is the key point to configure in this tutorial.

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

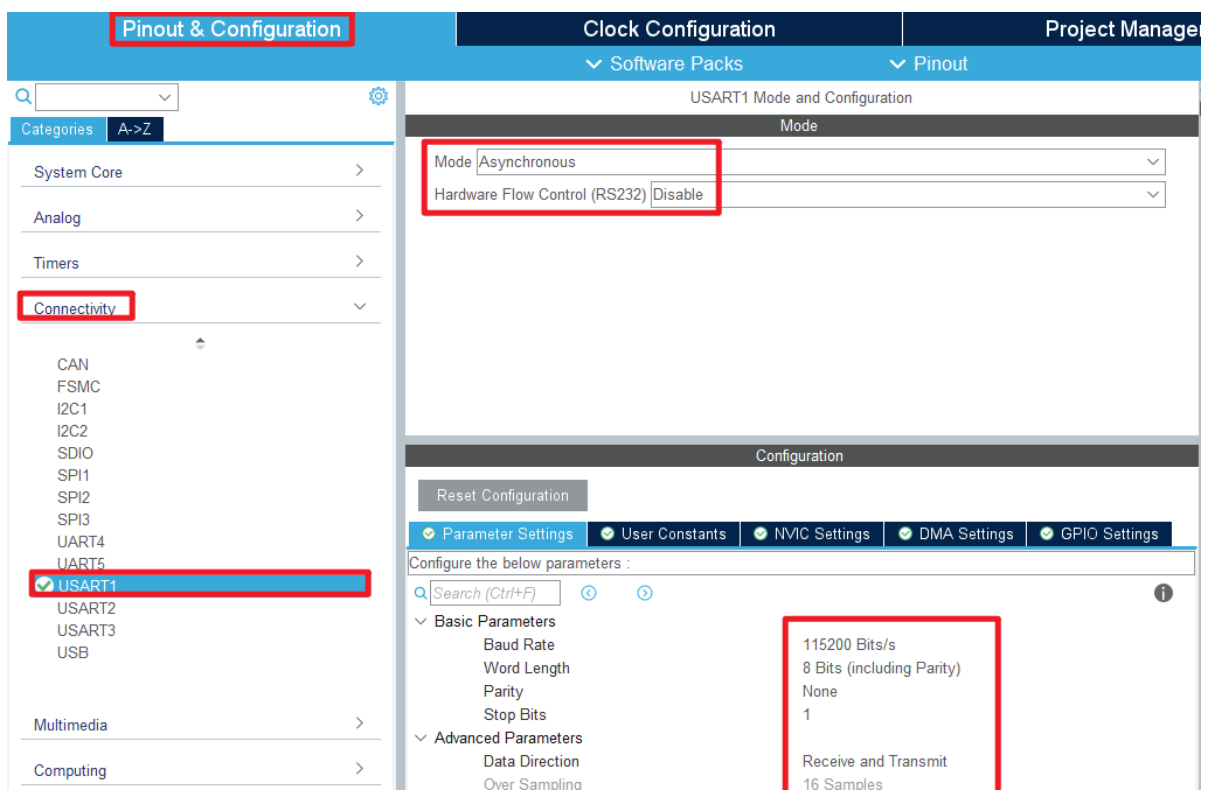
3.2、Pin configuration

- **Configure the specified pin function**

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



• USART



• TIM1

Pinout & Configuration

Search

Categories A-Z

System Core

Analog

Timers

RTC

TIM1

TIM2

TIM3

TIM4

TIM5

TIM6

TIM7

TIM8

Connectivity

Multimedia

Computing

Middleware and Software Packs

Clock Configuration

Software Packs

Pinout

Project Manager

TIM1 Mode and Configuration

Mode

Slave Mode

Reset Mode

Trigger Source

TI1FP1

Clock Source

Internal Clock

Channel1

Input Capture direct mode

Channel2

Input Capture indirect mode

Channel3

Disable

Channel4

Disable

Combined Channels

Disable

Configuration

Reset Configuration

Parameter Settings

User Constants

NVIC Settings

DMA Settings

GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - 16 bits value)

72-1

Counter Mode

Up

Counter Period (AutoReload Register - 16 bits value)

65535

Internal Clock Division (CKD)

No Division

Repetition Counter (RCR - 8 bits value)

0

auto-reload preload

Enable

Slave Mode Controller

Reset Mode

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Trigger Event Selection

Reset (UG bit from TIMx_EGR)

Input Capture Channel 1

Polarity Selection

Rising Edge

IC Selection

Direct

Prescaler Division Ratio

No division

Input Filter (4 bits value)

0

Input Capture Channel 2

Polarity Selection

Falling Edge

IC Selection

Indirect

Prescaler Division Ratio

No division

• TIM3

Pinout & Configuration

Search

Categories A-Z

System Core

Analog

Timers

RTC

TIM1

TIM2

TIM3

TIM4

TIM5

TIM6

TIM7

TIM8

Connectivity

Multimedia

Computing

Middleware and Software Packs

Clock Configuration

Software Packs

Pinout

Project Manager

TIM3 Mode and Configuration

Mode

Slave Mode

Disable

Trigger Source

Disable

Clock Source

Internal Clock

Channel1

Disable

Channel2

Disable

Channel3

PWM Generation CH3

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

GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - 16 bits value)

72-1

Counter Mode

Up

Counter Period (AutoReload Register - 16 bits value)

10000-1

Internal Clock Division (CKD)

No Division

auto-reload preload

Enable

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Trigger Event Selection

Reset (UG bit from TIMx_EGR)

PWM Generation Channel 3

Mode

PWM mode 1

Pulse (16 bits value)

5000

Output compare preload

Enable

Fast Mode

Disable

CH Polarity

High

• NVIC

The NVIC option allows you to change the priority

Pinout & Configuration

Categories

A-Z

System Core

DMA

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Analog

Timers

Connectivity

Multimedia

Computing

Middleware and Software Packs

Clock Configuration

Software Packs

Pinout

Project Manager

NVIC Mode and Configuration

Configuration

Priority Group

4 bits for pre-emption...

Sort by Preemption Priority and Sub Priority

Sort by interrupts names

Search

Search (Ctrl+F)

Show

available interrupts

Force DMA channels Interrupts

NVIC Interrupt Table

Enabled

Preemption Priority

Sub Priority

Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Prefetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	15	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
TIM1 break interrupt	<input type="checkbox"/>	0	0
TIM1 update interrupt	<input type="checkbox"/>	0	0
TIM1 trigger and commutation interrupts	<input type="checkbox"/>	0	0
TIM1 capture compare interrupt	<input checked="" type="checkbox"/>	4	0
TIM3 global interrupt	<input type="checkbox"/>	0	0
USART1 global interrupt	<input checked="" type="checkbox"/>	5	0

Advanced Settings

Pinout & Configuration

Clock Configuration

Project Manager

Project

Code Generator

Advanced Settings

Driver Selector

Search (Ctrl+F)

RCC

GPIO

TIM

USART

HAL

HAL

HAL

HAL

Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Instance Name	Do Not Generate Function Call	Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_Config	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_TIM1_Init	TIM1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	4	MX_TIM3_Init	TIM3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	5	MX_USART1_UART_Init	USART1	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Generating code



4、 Main function

For detailed code, you can open the project file provided by us and go to the Bsp folder to view the source code.

4.1、 User functions

HAL_TIM_IC_CaptureCallback

Function prototypes	void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
Functional Description	The input capture interrupt event used to process the timer
Input parameters	htim : Timer handle address
Return value	None

Function prototypes	void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
notes	Inside the function, you can write code to handle input capture events, such as reading the capture register value, calculating the pulse width, and so on

In the HAL_TIM_IC_CaptureCallback function, we read the counts of IC1 and IC2, which are multiplied by a single count plus 1 to give the period and pulse width.

4.2、 HAL library functions

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]

function: HAL_TIM_PWM_Init

Function prototypes	HAL_StatusTypeDef HAL_TIM_PWM_Init(TIM_HandleTypeDef *htim)
Functional Description	Initialize the timer's PWM output mode
Input parameters	htim : Timer handle address
Return value	HAL status value : HAL_OK、 HAL_ERROR、 HAL_BUSY、 HAL_TIMEOUT

function: HAL_TIM_MspPostInit

Function prototypes	void HAL_TIM_MspPostInit(TIM_HandleTypeDef* timHandle)
Functional Description	Peripheral clock, GPIO, and NVIC to initialize the timer
Input parameters	htim : Timer handle address
Return value	None
notes	The function performs additional initialization on top of HAL_TIM_Base_MspInit

function: HAL_TIM_PWM_Start

Function prototypes	HAL_StatusTypeDef HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim, uint32_t Channel)
Functional Description	Start PWM output

Function prototypes	HAL_StatusTypeDef HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim, uint32_t Channel)
Input parameters1	htim : Timer handle address
Input parameters2	Channel : Timer channel number
Return value	HAL status value : HAL_OK、 HAL_ERROR、 HAL_BUSY、 HAL_TIMEOUT

function: HAL_TIM_PWM_ConfigChannel

Function prototypes	HAL_StatusTypeDef HAL_TIM_PWM_ConfigChannel(TIM_HandleTypeDef *htim, const TIM_OC_InitTypeDef *sConfig, uint32_t Channel)
Functional Description	Set the PWM channel for the timer
Input parameters1	htim : Timer handle address
Input parameters2	sConfig : The timer outputs the comparison parameters
Input parameters3	Channel : Timer channel number
Return value	HAL status value : HAL_OK、 HAL_ERROR、 HAL_BUSY、 HAL_TIMEOUT

function: HAL_TIM_IC_Init

Function prototypes	HAL_StatusTypeDef HAL_TIM_IC_Init(TIM_HandleTypeDef *htim)
Functional Description	Initialize the input capture function of the timer
Input parameters	htim : Timer handle address
Return value	HAL status value : HAL_OK、 HAL_ERROR、 HAL_BUSY、 HAL_TIMEOUT

5、Experimental phenomenon

After downloading the program successfully, press the RESET button of the development board to open the serial debugging assistant to observe the phenomenon

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

Phenomenon: You can see that the data displayed by the serial port is the same as the PWM signal data generated by the universal timer (TIM3).

UART Assistant

UART Assist V5.0.3

COM Configs

ChannelCOM5 RU:▼

Baudrate115200▼

ParitybitsNONE▼

Databits8▼

Stopbits1▼

FlowctrlNONE▼

Close

Recv Options

ASCII

HEX

☒ Log Display Mode

☒ Auto Linefeed

☐ Hide Received Data

☐ Save Recv to File...

AutoScroll

Clear

AutoReply

Themes

BatchSend

Datastream

DataChart

Checksum

ASCII Map

Donate

Send Options

ASCII

HEX

☒ Use Escape Chars①

☐ Auto Append Bytes

☐ Send from File ...

☐ Cycle1000ms

Shortcut

History

Data Log

[2023-11-23 10:59:20.030]# RECV ASCII>
IC1Value = 9999
IC2Value = 4999
DutyCycle = 50.000000
Frequency = 100.000000

[2023-11-23 10:59:20.540]# RECV ASCII>
IC1Value = 9999
IC2Value = 4999
DutyCycle = 50.000000
Frequency = 100.000000

[2023-11-23 10:59:21.043]# RECV ASCII>
IC1Value = 9999
IC2Value = 4999
DutyCycle = 50.000000
Frequency = 100.000000

[2023-11-23 10:59:21.559]# RECV ASCII>
IC1Value = 9999
IC2Value = 4999
DutyCycle = 50.000000
Frequency = 100.000000

[2023-11-23 10:59:22.059]# RECV ASCII>
IC1Value = 9999
IC2Value = 4999
DutyCycle = 50.000000
Frequency = 100.000000

[2023-11-23 10:59:22.577]# RECV ASCII>
IC1Value = 9999
IC2Value = 4999
DutyCycle = 50.000000
Frequency = 100.000000

Data Send

1.DCD●2.RXD●3.TXD●4.DTR●5.GND●6.DSR●7.RTS●8.CTS●9.RI●

Clear

Clear

Send

Ready!

41/0

RX:3400

TX:0

Reset