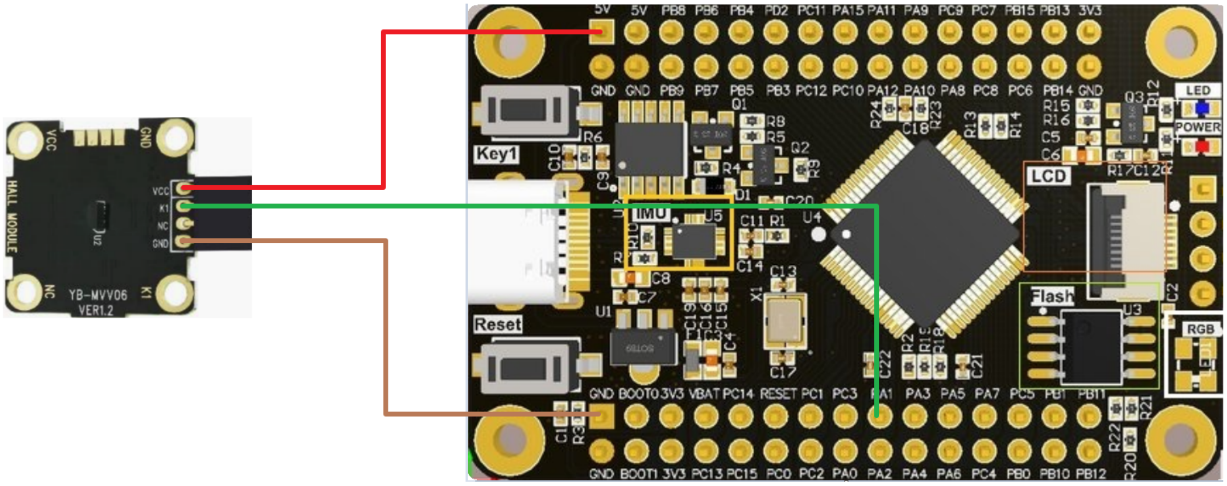
Building block servo module: timer (PWM)

Hardware wiring


VCC
GND
K1
NC
HALL MODULE
YB-MVV06
VER1.2
5V 5V PB8 PB6 PB4 PD2 PC11 PA15 PA11 PA9 PC9 PC7 PB15 PB13 3V3
GND GND PB9 PB7 PB5 PB3 PC12 PC10 PA12 PA10 PA8 PC8 PC6 PB14 GND
GND BOOT0 3V3 VBAT PC14 RESET PC1 PC3 PA1 PA3 PA5 PA7 PC5 PB1 PB11
GND BOOT1 3V3 PC13 PC15 PC0 PC2 PA0 PA2 PA4 PA6 PC4 PB0 PB10 PB12
Key1
Reset
IMU
LCD
Flash
RGB
LED
POWER

Note: The block servo B01/B02 is wired the same, and the pin connection is made according to the color.

Building block servo module	STM32F103RCT6
VCC(Red)	5V/3.3V
SIG(Yellow)	PB4
GND(Brown)	GND

Brief principle

Generally speaking, the PWM signal received by the servo has a frequency of 50HZ, that is, the period is 20ms.

When the pulse width of the high level is different, the servo can rotate to different angles.

B01（270°)	High level pulse width(us)
-45°	600
90°	1500
225°	2400

B02（360°)	High level pulse width（us)
0°	500
90°	1000

B02（360°）	High level pulse width（us）
180°	1500
270°	2000
360°	2500

Main code

The program code of the building block servo B01/B02 is slightly different except for the main function (main.c) code, and the other code files are the same.

B01 main.c

```
#include "stm32f10x.h"
#include "PWM.h"
#include "SysTick.h"

int main(void)
{

    SysTick_Init();//滴答定时器初始化
    TIM3_PWM_Init();//定时器PWM输出初始化(TIM3_CH1)

    while(1)
    {
        /* 转动角度-45度 */
        TIM_SetCompare1(TIM3, 600);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
        /* 转动角度90度 */
        TIM_SetCompare1(TIM3, 1500);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
        /* 转动角度225度 */
        TIM_SetCompare1(TIM3, 2400);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
    }
}
```

B02 main.c

```
#include "stm32f10x.h"
#include "PWM.h"
#include "SysTick.h"

int main(void)
{

    SysTick_Init();//滴答定时器初始化
    TIM3_PWM_Init();//定时器PWM输出初始化(TIM3_CH1)
    while(1)
    {
        /* 转动角度0度 */
        TIM_SetCompare1(TIM3, 500);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
        /* 转动角度90度 */
        TIM_SetCompare1(TIM3, 1000);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
        /* 转动角度180度 */
        TIM_SetCompare1(TIM3, 1500);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
        /* 转动角度270度 */
        TIM_SetCompare1(TIM3, 2000);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
        /* 转动角度360度 */
        TIM_SetCompare1(TIM3, 2500);//定时器设置比较值函数和通道有关 时间参数单位是us
        Delay_us(1000000);
    }
}
```

SysTick.c

```
#include "SysTick.h"

unsigned int Delay_Num;

void SysTick_Init(void)//滴答定时器初始化
{
    while(SysTick_Config(72));//设置重装载值 72 对应延时函数为微秒级
    //若将重装载值设置为72000 对应延时函数为毫秒级
    SysTick->CTRL &= ~(1 << 0);//定时器初始化后关闭，使用再开启
}

void Delay_us(unsigned int NCount)//微秒级延时函数
{
    Delay_Num = NCount;
    SysTick->CTRL |= (1 << 0);//开启定时器
    while(Delay_Num);
    SysTick->CTRL &= ~(1 << 0);//定时器初始化后关闭，使用再开启
}

void SysTick_Handler(void)
{
    if(Delay_Num != 0)
    {
        Delay_Num--;
    }
}
```

SysTick.h

```
#ifndef __SYSTICK_H__
#define __SYSTICK_H__

#include "stm32f10x.h"

void SysTick_Init(void);//滴答定时器初始化
void Delay_us(unsigned int NCount);//微秒级延时函数

#endif
```

PWM.c

```
#include "PWM.h"

void TIM3_PWM_Init(void)//定时器PWM输出初始化(TIM3_CH1)
{
    TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
    TIM_OCInitTypeDef  TIM_OCInitStructure;
    GPIO_InitTypeDef GPIO_InitStructure;

    /* TIM3 clock enable */
    /* TIM3 时钟使能 */
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);

    /* GPIOB and AFIO clock enable */
    /* 使能GPIOB端口时钟和AFIO时钟 */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB | RCC_APB2Periph_AFIO, ENABLE);

    /*GPIOB Configuration: TIM3 channel1 */
    /* PB4配置复用推挽输出模式 */
    GPIO_InitStructure.GPIO_Pin =  GPIO_Pin_4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    /* JTAG-DP Disabled and SW-DP Enabled */
    /* 禁用JTAG 启用SWD */
    GPIO_PinRemapConfig(GPIO_Remap_SWJ_JTAGDisable, ENABLE);
    /* 把TIM3_CH1部分映射到PB4引脚上 */
    GPIO_PinRemapConfig(GPIO_PartialRemap_TIM3, ENABLE);

    /* Time base configuration */
    /* 定时计数器配置 频率 100Hz*/
    TIM_TimeBaseStructure.TIM_Period = 20000 - 1;//舵机控制脉冲周期20ms 对应20000us
    TIM_TimeBaseStructure.TIM_Prescaler = 72 - 1;//72MHz / 72 =1MHz 1
    TIM_TimeBaseStructure.TIM_ClockDivision = 0;
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
    /* PWM1 Mode configuration: Channel1 */
    /* PWM1 模式配置：TIM3_CH1 */
    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;//PWM模式1
    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;//PWM输出使能
    TIM_OCInitStructure.TIM_Pulse = 0;//设置比较值 决定占空比
    TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;//有效电平 高电平
    TIM_OC1Init(TIM3, &TIM_OCInitStructure);

    /* 使能预装载寄存器 */
    TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable);

    /* 使能自动重装载寄存器 */
    TIM_ARRPreloadConfig(TIM3, ENABLE);

    /* TIM3 enable counter */
    /* 使能TIM3计数 */
    TIM_Cmd(TIM3, ENABLE);
}
```

PWM.h

```
#ifndef __PWM_H__
#define __PWM_H__

#include "stm32f10x.h"

void TIM3_PWM_Init(void);//定时器PWM输出初始化(TIM3_CH1)

#endif
```

Phenomenon

B01: After downloading the program, the brick helmsman rotates from -45°→90°→225°.

B02: After downloading the program, the brick helmsman rotates from 0°→90°→180°→270°→360°.