STM32Cube高效开发教程（基础篇）

第10章 通用定时器

王维波
中国石油大学（华东）控制科学与工程学院

STM32Cube高效开发教程（基础篇）

作者：王维波，鄢志丹，王钊

人民邮电出版社

2021年9月出版

如果有读者需要本书课件的PPT版本用于备课，可以给作者发邮件免费获取，并可加入专门的教学和技术交流QQ群

邮箱：wangwb@upc. edu. cn

第10章 通用定时器

10.1 通用定时器功能概述

10.2 典型功能原理和HAL驱动

10.3 生成PWM波示例

10.1.1 功能概述

通用定时器TIM2-TIM5，TIM9-TIM14，计数器位数，通道数不同

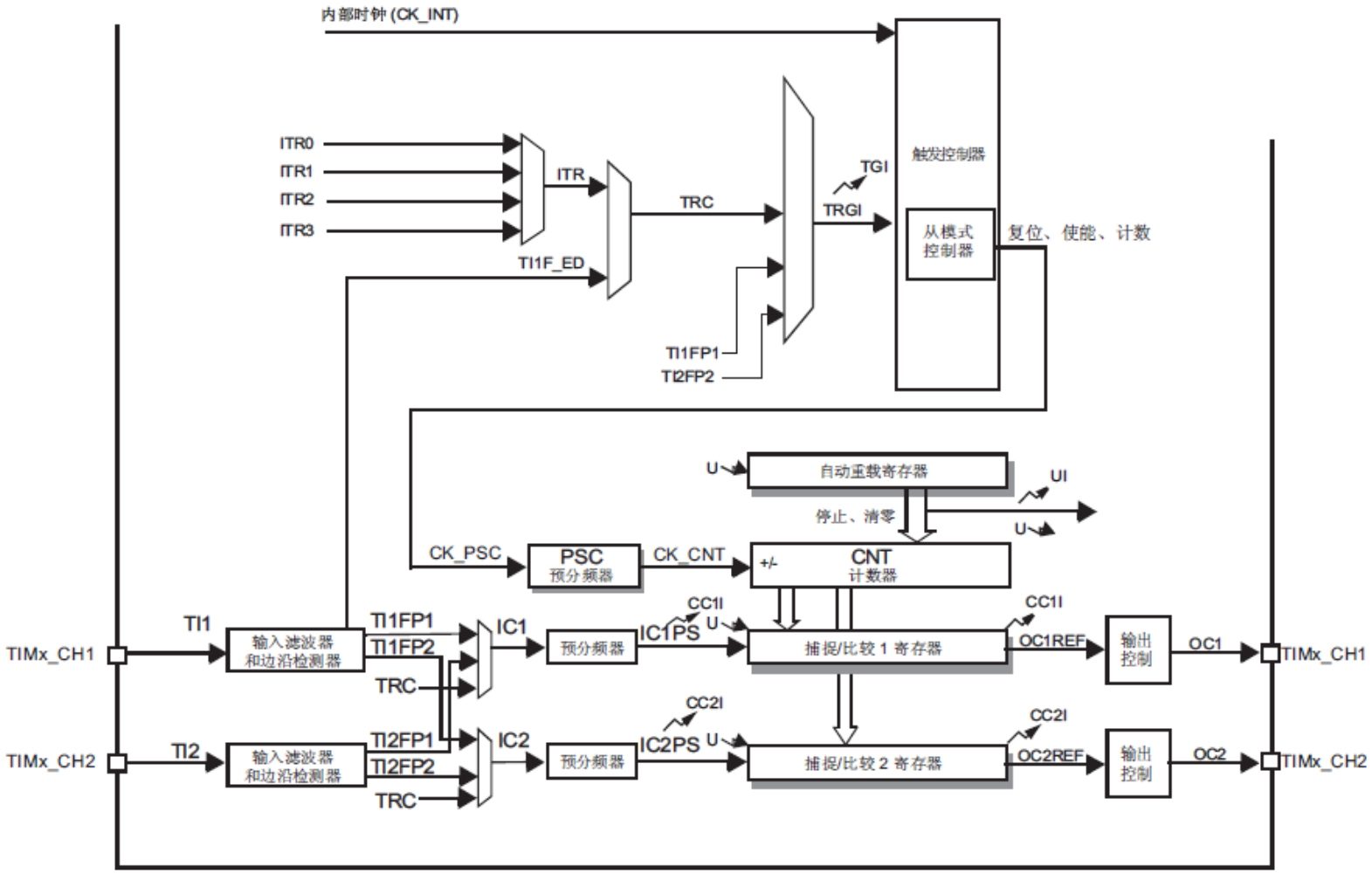
类型	定时器	计数器长度	计数类型	DMA请求生成	捕获/比较通道数	挂载总线
基础	TIM6，TIM7	16位	递增	有	0	APB1
通用	TIM2，TIM5	32位	递增、递减、递增/递减	有	4	APB1
	TIM3，TIM4	16位	递增、递减、递增/递减	有	4	APB1
	TIM9	16位	递增	无	2	APB2
	TIM12	16位	递增	无	2	APB1
	TIM10，TIM11	16位	递增	无	1	APB2
	TIM13，TIM14	16位	递增	无	1	APB1
高级控制	TIM1，TIM8	16位	递增、递减、递增/递减	有	4	APB2

通用定时器的功能：

- 输入捕获（Input Captur）
- 输出比较（Output Compare）
- 生成PWM波
- 测量PWM波周期和占空比
- 定时器同步
- 定时器串联

10.1.2 结构框图

以两通道定时器为例



1. 时钟信号和触发控制器

定时器可以使用内部时钟（CK_INT）驱动，内部时钟来源于APB1或APB2总线的定时器时钟信号。如果定时器设置为从模式，还可以使用其他定时器输出的触发信号作为时钟信号，也就是图10-1中的ITR0、ITR1等。

触发控制器用于选择定时器的时钟信号，并且可以控制定时器的复位、使能、计数等。触发控制器输出时钟信号CK_PSC，若选择使用内部时钟，则CK_PSC就等于CK_INT。

2. 时基单元工作原理

时基单元包括3个寄存器：

（1）计数寄存器（CNT），这个寄存器存储计数器当前的计数值，可以在运行时被读取。

（2）预分频寄存器（PSC），寄存器数值范围0至65535，对应于分频系数1至65536。

（3）自动重载寄存器（ARR），这个寄存器存储的是定时器计数周期。

3. 捕获/比较通道

捕获/比较通道由3个阶段组成。

- 输入阶段：通道作为输入引脚。
- 捕获/比较主电路。捕获/比较寄存器CCR与计数器进行比较。
- 输出阶段：输出阶段就是根据设置的工作模式和控制逻辑，控制输出引脚的电平。

使用捕获/比较通道，通用定时器可以实现如下的功能：

- 输入捕获，可以用于测量一个时钟信号的频率，脉冲宽度等。
- 输出比较，将计数器CNT的值与CCR寄存器比较，控制输出引脚的电平。
- PWM生成，通过设置ARR寄存器和CCR寄存器的值，在计数器的值CNT变化过程中，输出PWM波。PWM波的频率由ARR寄存器决定，占空比由CCR寄存器决定。

第10章 通用定时器

10.1 通用定时器功能概述

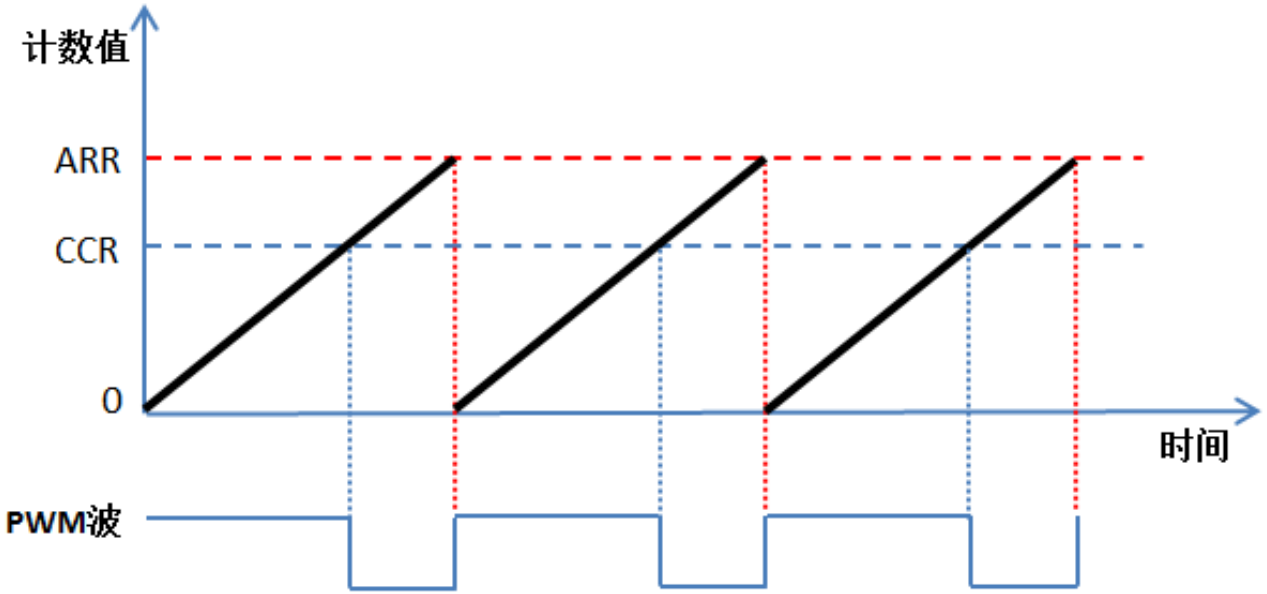
10.2 典型功能原理和HAL驱动

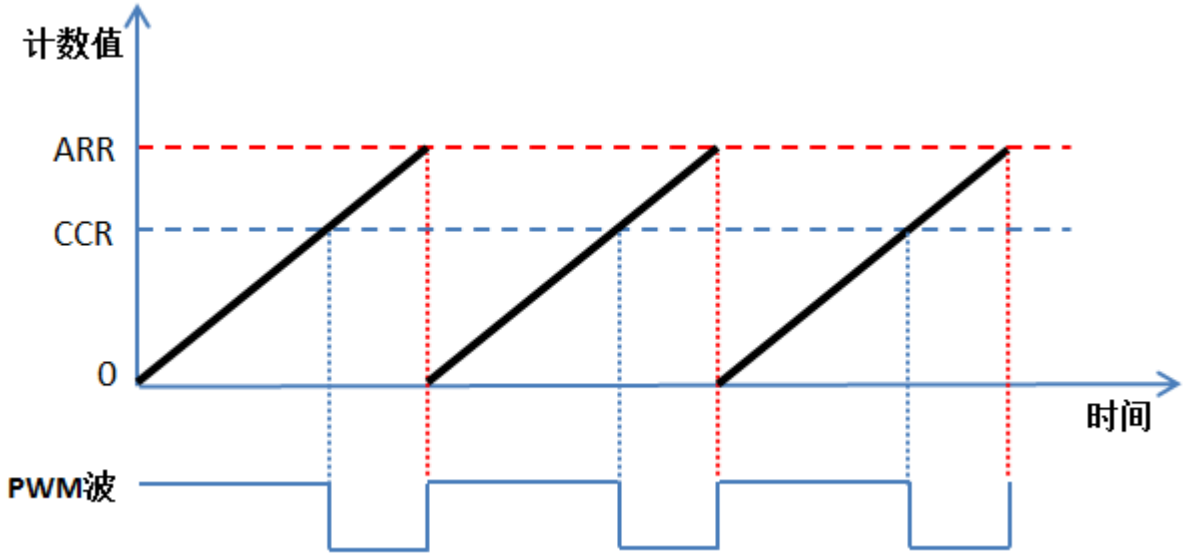
10.3 生成PWM波示例

10.2.1 生成PWM波

1. 生成PWM波的原理

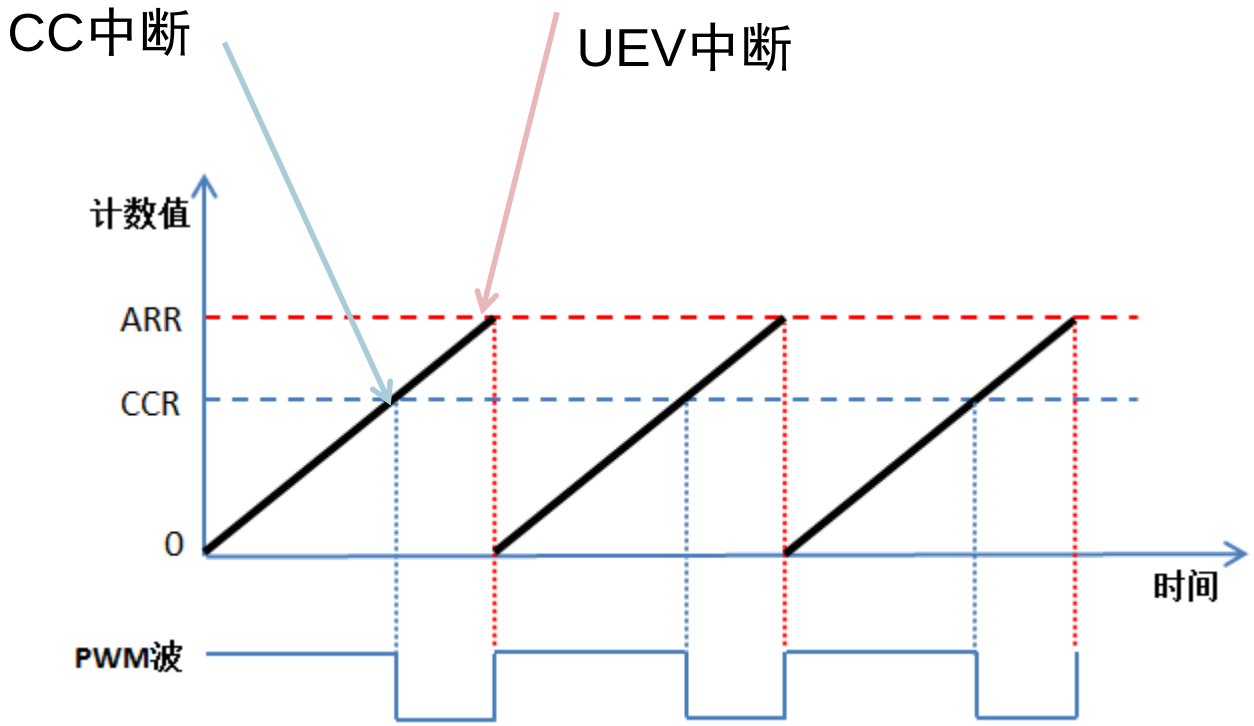
PWM（Pulse Width Modulation）就是脉冲宽度调制，PWM波就是具有一定占空比的方波信号，通过定时器的设置可以控制方波的频率和占空比。





- 自动重载寄存器ARR的值决定了PWM波一个周期的长度，比如PWM一个周期是100ms。

- 捕获/比较寄存器CCR的值决定了占空比，比如通过设置CCR的值，使得一个周期内高电平时长为70ms，则占空比为70%。



- 在计数器的值达到ARR值时产生UEV事件。CCR寄存器具有预装载功能。

2. 生成PWM波相关HAL函数

函数名	功能描述
HAL_TIM_PWM_Init()	生成PWM波的配置初始化，需先执行HAL_TIM_Base_Init()进行定时器初始化
HAL_TIM_PWM_ConfigChannel()	配置PWM输出通道
HAL_TIM_PWM_Start()	启动生成PWM波，需要先执行HAL_TIM_Base_Start()启动定时器
HAL_TIM_PWM_Stop()	停止生成PWM波
HAL_TIM_PWM_Start_IT()	以中断方式启动生成PWM波，需要先执行HAL_TIM_Base_Start_IT()启动定时器
HAL_TIM_PWM_Stop_IT()	停止生成PWM波
HAL_TIM_PWM_GetState()	返回定时器状态，与HAL_TIM_Base_GetState()功能相同
__HAL_TIM_ENABLE_OCxPRELOAD()	使能CCR寄存器的预装载功能，为CCR设置的新值在下个UEV事件发生时才更新到CCR寄存器
__HAL_TIM_DISABLE_OCxPRELOAD()	禁止CCR寄存器的预装载功能，为CCR设置的新值立刻更新到CCR寄存器
__HAL_TIM_ENABLE_OCxFAST()	启用一个通道的快速模式
__HAL_TIM_DISABLE_OCxFAST()	禁用一个通道的快速模式
HAL_TIM_PWM_PulseFinishedCallback()	当计数器的值等于CCR寄存器的值时产生输出比较事件，这是对应的回调函数

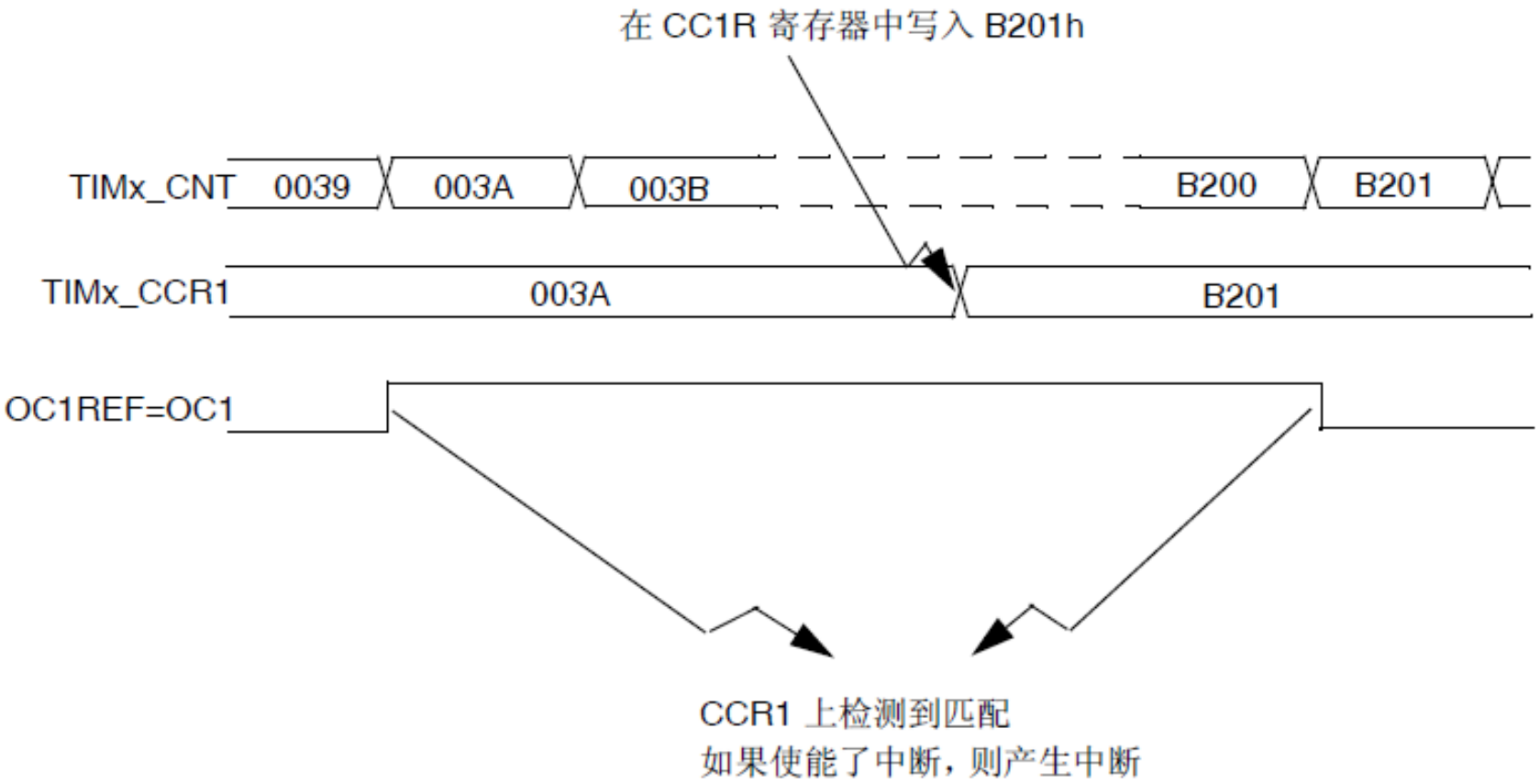
10.2.2 输出比较

1. 输出比较的原理

输出比较（Output Compare）用于控制输出波形，或指示经过了某一段时间。

工作原理：用寄存器CCR的值与计数器值CNT比较，当两个寄存器的值匹配时，产生输出比较结果OCxREF，比较结果可以输出到通道的引脚。比较匹配时，输出引脚电平可以是：

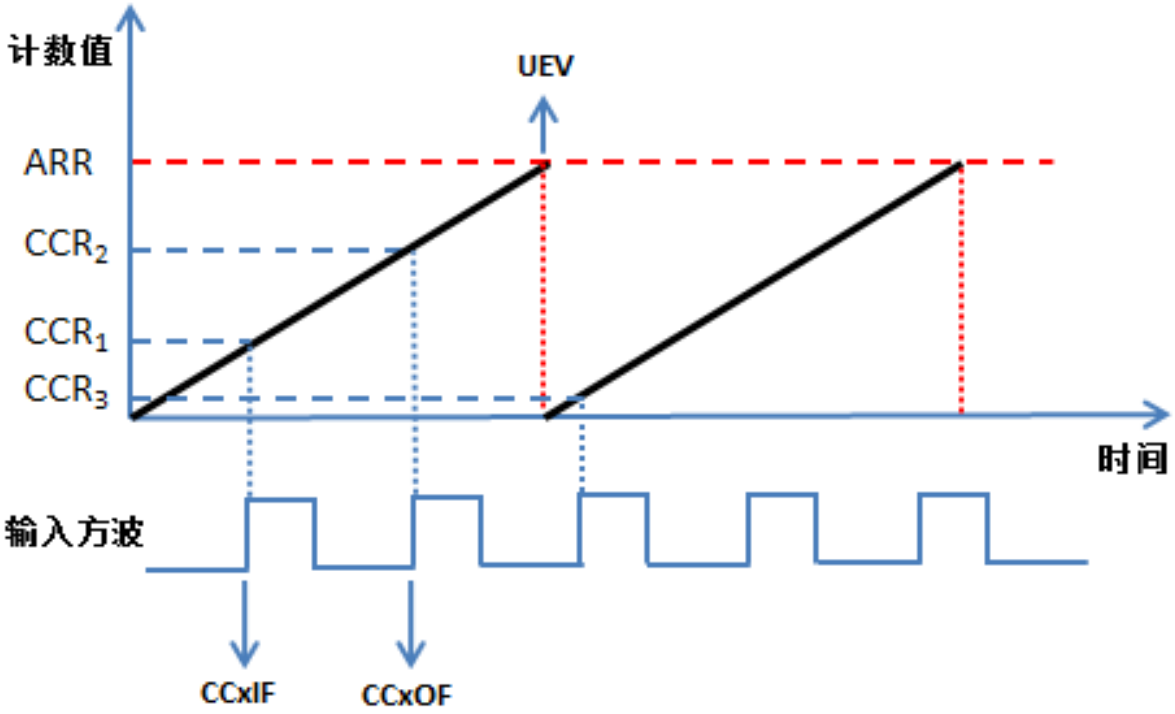
- 冻结（Frozen），即保持其电平
- 有效电平（Active level），有效电平由设置的通道极性决定
- 无效电平（Inactive Level）
- 翻转（Toggle）



输出比较工作时序图，翻转输出，TIMx_CCR1无预装载功能

10.2.3 输入捕获

1. 输入捕获的原理



输入捕获（Input Capture）就是检测输入通道方波信号的跳变沿，并将发生跳变时的计数器的值锁存到捕获/比较寄存器CCR。

10.2.6 通用定时器中断事件和回调函数

定时器中断事件类型宏定义

```
#define TIM_IT_UPDATE   TIM_DIER_UIE      //更新中断（Update interrupt）
#define TIM_IT_CC1      TIM_DIER_CC1IE    //捕获/比较1中断
#define TIM_IT_CC2      TIM_DIER_CC2IE    //捕获/比较2中断
#define TIM_IT_CC3      TIM_DIER_CC3I E   //捕获/比较3中断
#define TIM_IT_CC4      TIM_DIER_CC4IE    //捕获/比较4中断
#define TIM_IT_COM      TIM_DIER_COMIE    //换相中断（Commutation interrupt）
#define TIM_IT_TRIGGER TIM_DIER_TIE       //触发中断（Trigger interrupt）
#define TIM_IT_BREAK    TIM_DIER_BIE      //刹车中断（Break interrupt）
```

函数HAL_TIM_IRQHandler()的代码框架：

```
3167 void HAL_TIM_IRQHandler(TIM_HandleTypeDef *htim)
3168 {
3169   /* Capture compare 1 event */
3170   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_CC1) != RESET)..
3202   /* Capture compare 2 event */
3203   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_CC2) != RESET)..
3232   /* Capture compare 3 event */
3233   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_CC3) != RESET)..
3262   /* Capture compare 4 event */
3263   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_CC4) != RESET)..
3292   /* TIM Update event */
3293   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_UPDATE) != RESET)..
3305   /* TIM Break input event */
3306   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_BREAK) != RESET)..
3318   /* TIM Trigger detection event */
3319   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_TRIGGER) != RESET)..
3331   /* TIM commutation event */
3332   if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_COM) != RESET)..
3344 }
```

定时器中断事件类型与回调函数

中断事件类型	事件名称	回调函数
TIM_IT_CC1	CC1通道输入捕获	HAL_TIM_IC_CaptureCallback(htim)
	CC1通道输出比较	HAL_TIM_OC_DelayElapsedCallback(htim); HAL_TIM_PWM_PulseFinishedCallback(htim);
TIM_IT_CC2	CC2通道输入捕获	HAL_TIM_IC_CaptureCallback(htim)
	CC2通道输出比较	HAL_TIM_OC_DelayElapsedCallback(htim); HAL_TIM_PWM_PulseFinishedCallback(htim);
TIM_IT_CC3	CC3通道输入捕获	HAL_TIM_IC_CaptureCallback(htim)
	CC3通道输出比较	HAL_TIM_OC_DelayElapsedCallback(htim); HAL_TIM_PWM_PulseFinishedCallback(htim);
TIM_IT_CC4	CC4通道输入捕获	HAL_TIM_IC_CaptureCallback(htim)
	CC4通道输出比较	HAL_TIM_OC_DelayElapsedCallback(htim); HAL_TIM_PWM_PulseFinishedCallback(htim);
TIM_IT_UPDATE	更新事件（UEV）	HAL_TIM_PeriodElapsedCallback(htim);
TIM_IT_TRIGGER	TRGI触发事件	HAL_TIM_TriggerCallback(htim);
TIM_IT_BREAK	短路输入事件	HAL_TIMEx_BreakCallback(htim);
TIM_IT_COM	换相事件	HAL_TIMEx_CommutCallback(htim);

第10章 通用定时器

10.1 通用定时器功能概述

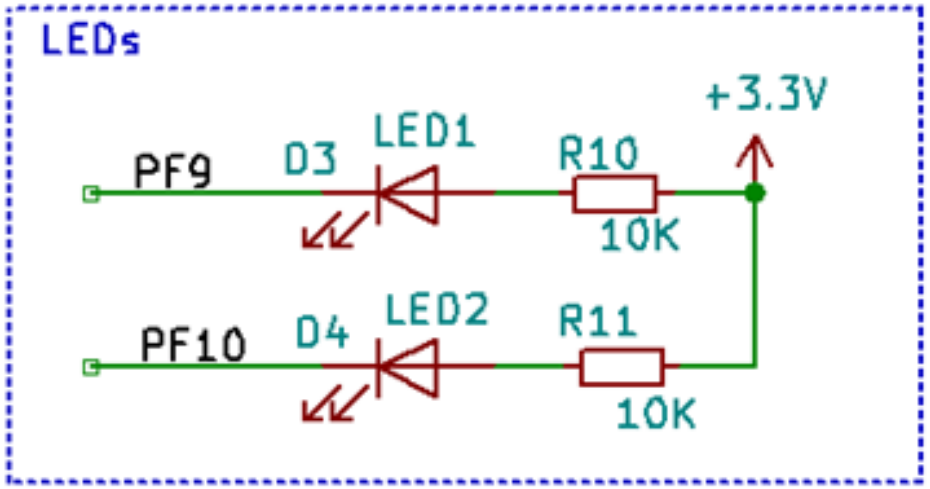
10.2 典型功能原理和HAL驱动

10.3 生成PWM波示例

10.3.1 电路原理和CubeMX项目配置

LED1连接的引脚PF9可以作为定时器TIM14的通道CH1，使用TIM14输出PWM波可以控制LED1的亮度。

使用TIM14的CH1通道生成PWM波，首先输出固定占空比的PWM波，然后再改动程序后输出可变占空比的PWM波。

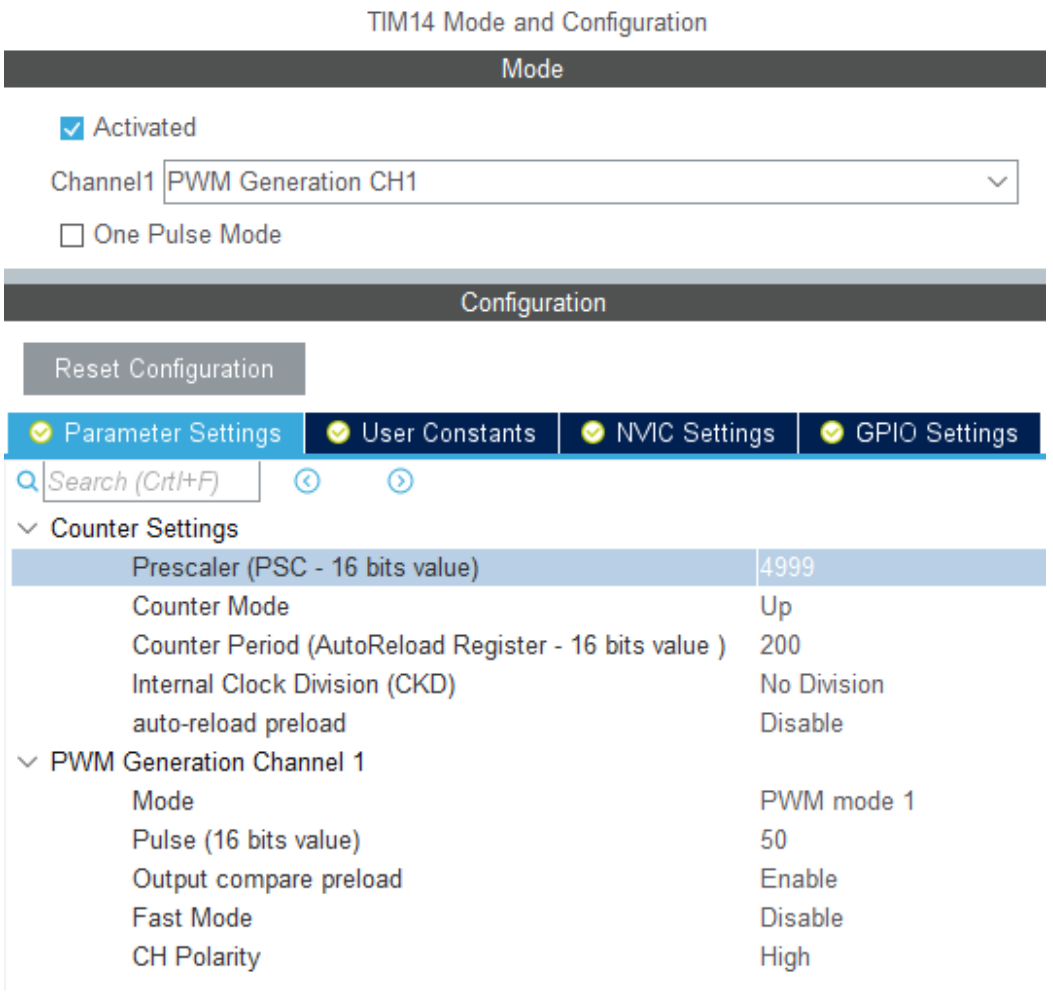


1. 基本设置

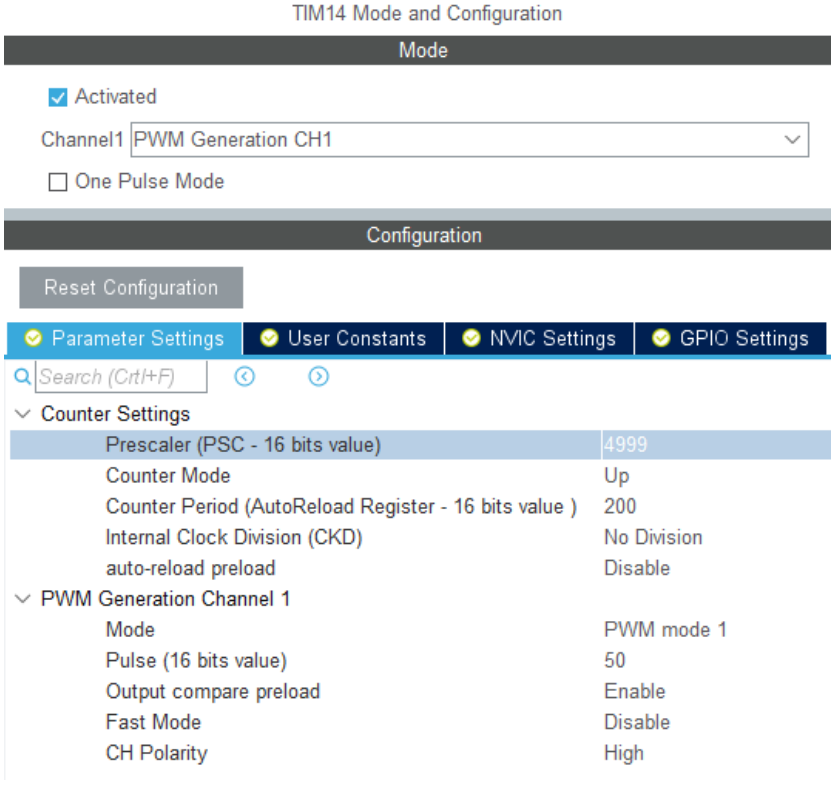
- 创建一个项目
- 然后导入TFT LCD的CubeMX项目
- 将4个按键和2个LED的引脚恢复为初始状态
- 在时钟树上设置HSE为8 MHz，HCLK为100 MHz，APB1和APB2总线定时器时钟频率都设置为50 MHz

2. 定时器TIM14的设置

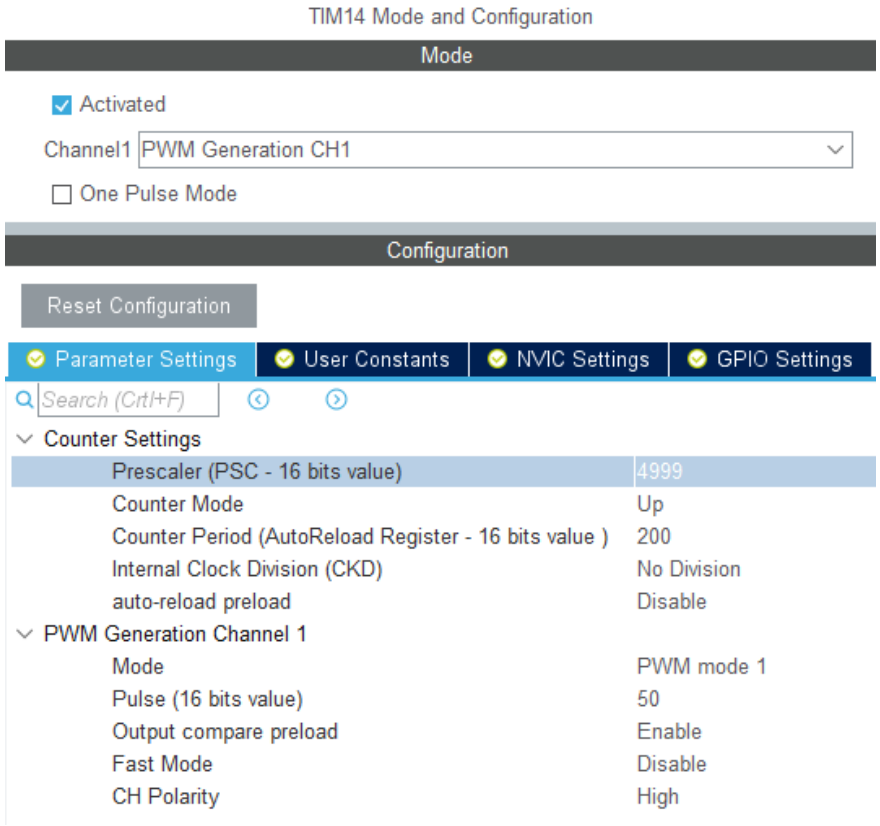
模式设置为PWM Generation CH1

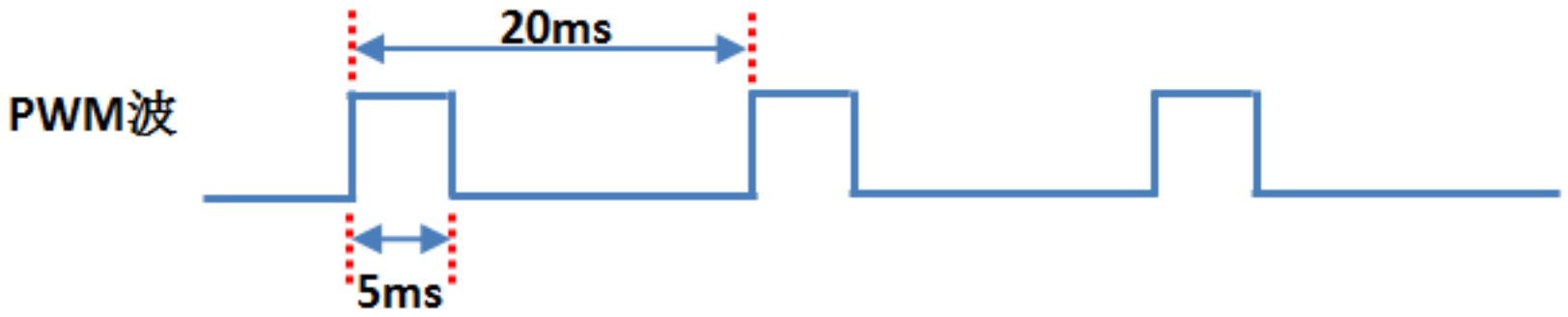


- Prescaler，预分频寄存器值，设置为4999，所以预分频系数为5000。定时器使用内部时钟信号频率为50 MHz，经过预分频后进入计数器的时钟频率就10 kHz
- Counter Mode，计数模式，设置为递增计数。
- Counter Period，计数周期（即自动装载寄存器ARR的值），设置为200，所以一个计数周期是200ms
- Internal Clock Division，内部时钟分频
- auto-reload preload，自动重载预装载，即设置TIM14_CR1寄存器中的ARPE位



- Mode，PWM模式，选项有PWM Mode 1和PWM Mode 2
- Pulse，PWM脉冲宽度，就是设置16位的捕获/比较寄存器CCR的值。脉冲宽度的值应该小于计数周期的值，这里设置为50，因为计数器的时钟频率是10kHz，所以脉冲宽度为5ms
- Output compare preload，输出比较预装载
- Fast Mode，是否使用输出比较快速使能
- CH Polarity，通道极性，就是CCR与CNT比较输出的有效状态，可以设置为高电平（High）或低电平（Low）





经过这样的设置，在引脚PF9（TIM14_CH1通道）上输出的PWM波形如图所示。

通道极性为高，PWM模式为1。PWM波的周期为20ms，由ARR寄存器的值决定；高电平脉冲宽度为5ms，由CCR寄存器决定。

10.3.2 输出固定占空比PWM波

1. 主程序

必须调用函数启动定时器，再启动定时器的PWM输出。

```
HAL_TIM_Base_Start_IT(&htim14); //以中断方式启动TIM14

HAL_TIM_PWM_Start_IT(&htim14,TIM_CHANNEL_1);
//TIM14通道1, 启动生成PWM
```

代码较长，看源程序main.c

2. 定时器TIM14初始化

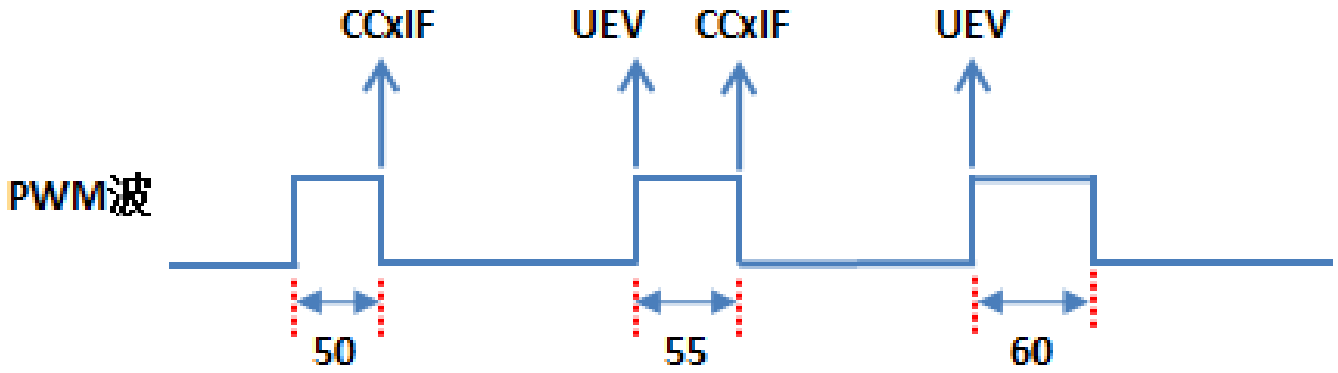
代码较长，看源程序文件tim.c中的完整代码，看书上的代码分析

3. 下载与测试

输出固定占空比PWM波，LED1固定亮度

10.3.3 输出可变占空比PWM波

1. 功能和原理



在程序中动态改变PWM波的占空比，就是要修改TIMx_CCR1寄存器的值，在本示例中，PWM波的周期是200个时钟周期，在发生比较匹配事件时，会产生CCxIF中断，可以在此中断里修改CCR寄存器的值。

2. 重新实现回调函数

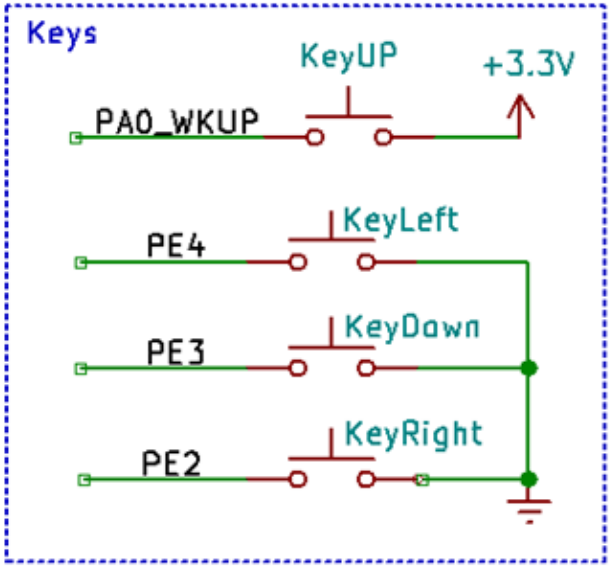
重新实现回调函数HAL_TIM_PWM_PulseFinishedCallback()，在此回调函数里编写代码改变占空比。

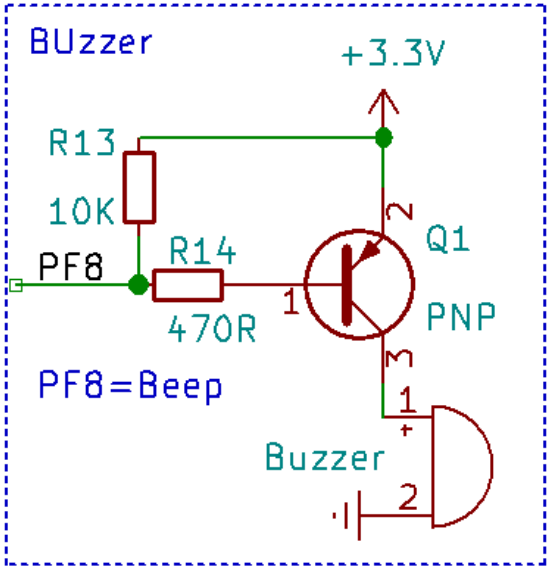
代码较长，看文件tim.c中的完整源代码

小作业

连接蜂鸣器的PF8引脚可以作为TIM13_CH1。

- 在TIM13_CH1输出PWM波
- 使用KeyUp和KeyDown改变PWM频率，发出简谱1~7的音调
- 使用KeyLeft和KeyDown改变PWM波占空比，看有什么影响





C 调各音符频率对照表

音符	频率 Hz	周期 μs
低 1Do	262	3816
低 2Re	294	3401
低 3Mi	330	3030
低 4Fa	349	2865
低 5So	392	2551
低 6La	440	2272
低 7Si	494	2024
中 1Do	523	1912
中 2Re	587	1703
中 3Mi	659	1517
中 4Fa	698	1432
中 5So	784	1275
中 6La	880	1136
中 7Si	988	1012
高 1Do	1047	955
高 2Re	1175	851
高 3Mi	1319	758
高 4Fa	1397	751
高 5So	1568	637
高 6La	1760	568
高 7Si	1967	508

- 按键循环发出简谱声音
- 根据乐谱演奏音乐，比如《两只老虎》

https://blog.csdn.net/epsilono1/article/details/85133242