STM32F1 系列中，除了互联型的产品，共有 8 个定时器，分为基本定时器，通用定时

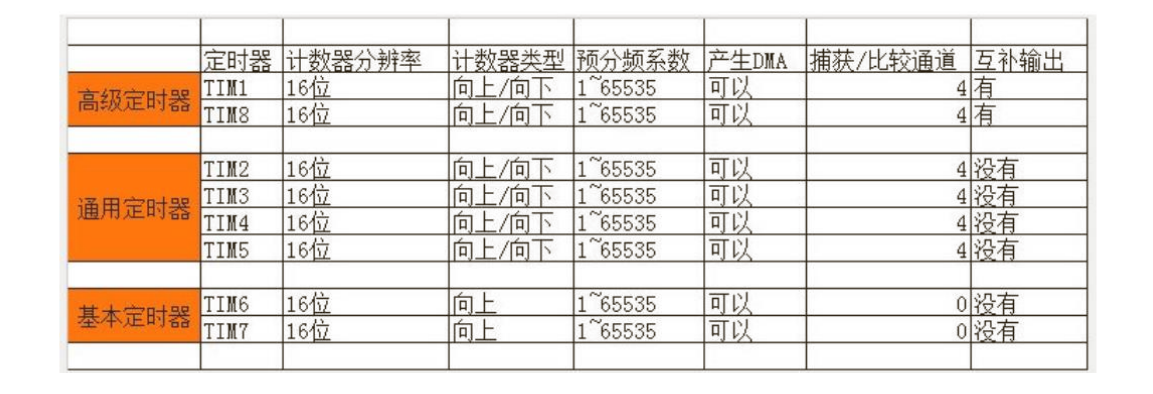
器和高级定时器。基本定时器 TIM6 和 TIM7 是一个 16 位的只能向上计数的定时器，只能

定时，没有外部 IO。通用定时器 TIM2/3/4/5 是一个 16 位的可以向上/下计数的定时器，可

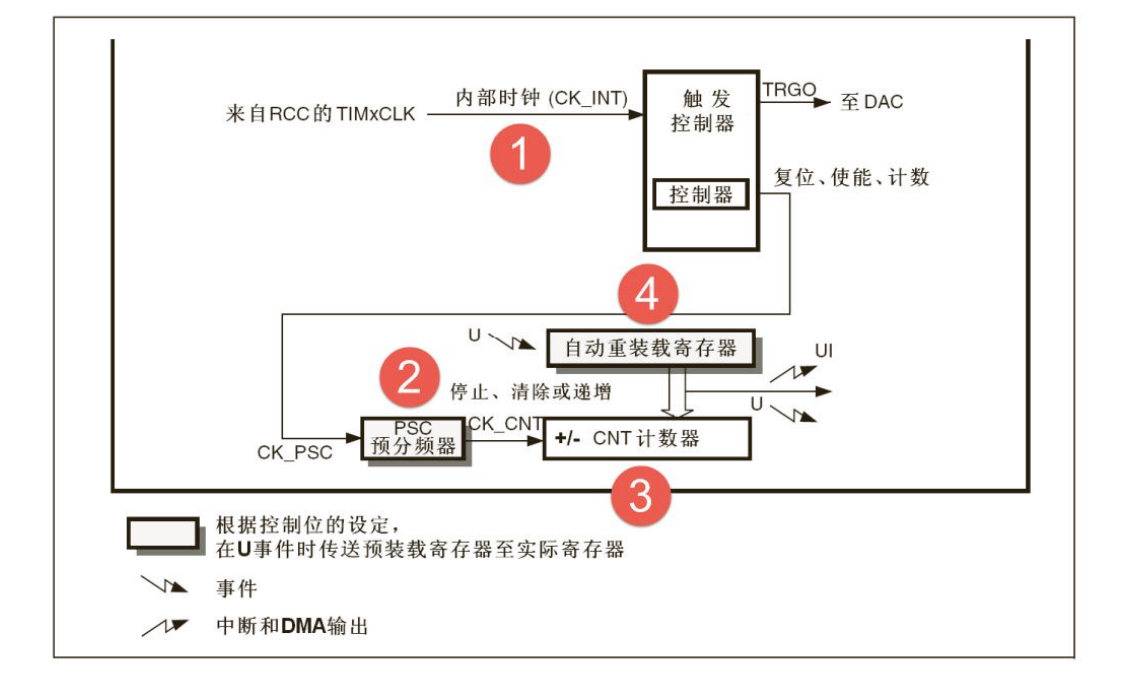
以定时，可以输出比较，可以输入捕捉，每个定时器有四个外部 IO。高级定时器 TIM1/8

是一个 16 位的可以向上/下计数的定时器，可以定时，可以输出比较，可以输入捕捉，还

可以有三相电机互补输出信号，每个定时器有 8 个外部 IO。



定时器的核心是时基，不仅基本定时器有，通用定时器和高级定时器也有。



1. ①时钟源

定时器时钟 TIMxCLK，即内部时钟 CK\_INT，经 APB1 预分频器后分频提供，如果

APB1 预分频系数等于 1，则频率不变，否则频率乘以 2，库函数中 APB1 预分频的系

数是 2，即 PCLK1=36M，所以定时器时钟 TIMxCLK=36\*2=72M 。

2. ②计数器时钟

定时器时钟经过 PSC 预分频器之后，即 CK\_CNT，用来驱动计数器计数。PSC 是一个

16 位的预分频器，可以对定时器时钟 TIMxCLK 进行 1~65536 之间的任何一个数进行分频。

具体计算方式为：CK\_CNT=TIMxCLK/(PSC+1)。

3. ③计数器

计数器 CNT 是一个 16 位的计数器，只能往上计数，最大计数值为 65535。当计数达

到自动重装载寄存器的时候产生更新事件，并清零从头开始计数。

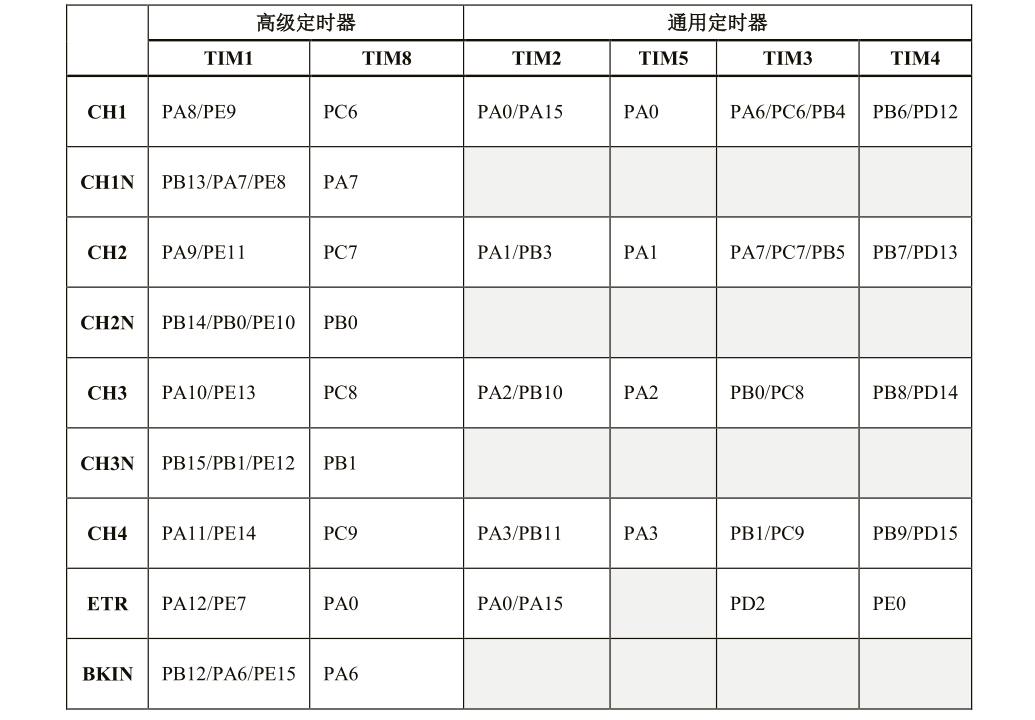
4. ④自动重装载寄存器

自动重装载寄存器 ARR 是一个 16 位的寄存器，这里面装着计数器能计数的最大数

值。当计数到这个值的时候，如果使能了中断的话，定时器就产生溢出中断。

STM32F103ZET6 的高级/通用定时器的 IO 分配具体见表。配套开发板因为 IO 资

源紧缺，定时器的 IO很多已经复用它途

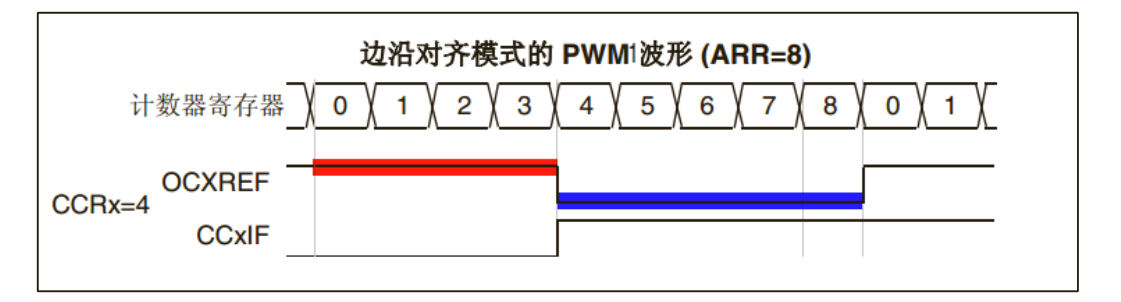


PWM输出就是对外输出脉宽（即占空比）可调的方波信号，信号频率由自动重装寄存

器 ARR的值决定，占空比由比较寄存器 CCR的值决定。下面我们以PWM1模式来讲解，以计数器CNT计数的方向不同还分为边沿对齐模式和中心对齐模式。PWM信号主要都是用来控制电机，一般的电机控制用的都是边沿对齐模式，FOC 电机一般用中心对齐模式。我们这里只分析这两种模式在信号感官上（即信号波形）的区别，具体在电机控制中的区别不做讨论，到了你真正需要使用的时候就会知道了。

PWM边沿对齐模式

在递增计数模式下，计数器从 0 计数到自动重载值（ TIMx\_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件



在边沿对齐模式下，计数器 CNT 只工作在一种模式，递增或者递减模式。这里我们以

CNT工作在递增模式为例，在中，ARR=8，CCR=4，CNT从 0开始计数，当 CNT<CCR 的

值时，OCxREF 为有效的高电平，于此同时，比较中断寄存器 CCxIF 置位。当

CCR=<CNT<=ARR时，OCxREF为无效的低电平。然后CNT又从0开始计数并生成计数器

上溢事件，以此循环往复。

定时器初始化结构体详解

在标准库函数头文件 stm32f4xx\_tim.h 中对定时器外设建立了四个初始化结构体，分别

为 时 基 初 始 化 结 构 体 TIM\_TimeBaseInitTypeDef 、 输 出 比 较 初 始 化 结 体

TIM\_OCInitTypeDef、输入捕获初始化结构体 TIM\_ICInitTypeDef 和断路和死区初始化结构

体 TIM\_BDTRInitTypeDef，高级控制定时器可以用到所有初始化结构体，通用定时器不能

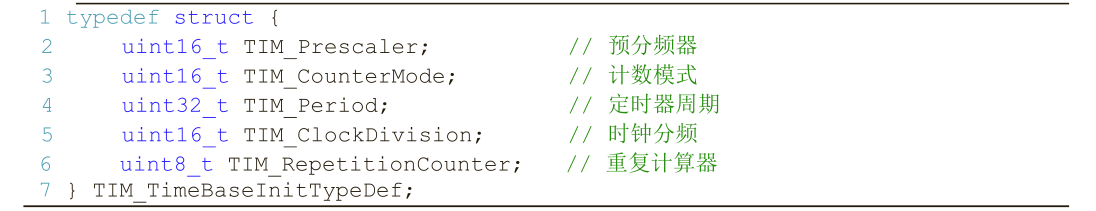
使用 TIM\_BDTRInitTypeDef 结构体，基本定时器只能使用时基结构体。接下来我们具体讲

解下这四个结构体

1. TIM\_TimeBaseInitTypeDef

时基结构体TIM\_TimeBaseInitTypeDef用于定时器基础参数设置，与TIM\_TimeBaseInit

函数配合使用完成配置。



(1) TIM\_Prescaler：定时器预分频器设置，时钟源经该预分频器才是定时器计数时钟

CK\_CNT，它设定 PSC 寄存器的值。计算公式为：计数器时钟频率 (f CK\_CNT ) 等于

f CK\_PSC / (PSC[15:0] + 1)，可实现 1至 65536 分频。

(2) TIM\_CounterMode：定时器计数方式，可设置为向上计数、向下计数以及中心对齐。

高级控制定时器允许选择任意一种。

(3) TIM\_Period：定时器周期，实际就是设定自动重载寄存器 ARR的值，ARR 为要装载

到实际自动重载寄存器（即影子寄存器）的值，可设置范围为 0至 65535。

(4) TIM\_ClockDivision：时钟分频，设置定时器时钟 CK\_INT 频率与死区发生器以及数字

滤波器采样时钟频率分频比。可以选择 1、2、4分频。

(5) TIM\_RepetitionCounter：重复计数器，只有 8 位，只存在于高级定时器，利用它可

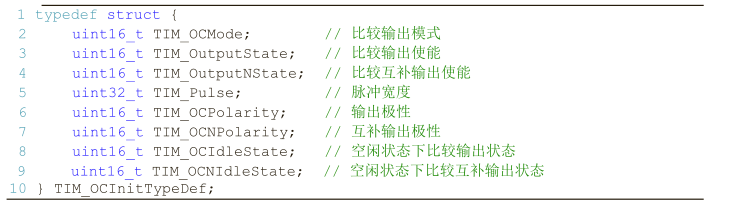
以非常容易控制输出 PWM 的个数。

2. TIM\_OCInitTypeDef

输出比较结构体 TIM\_OCInitTypeDef 用于输出比较模式，与 TIM\_OCxInit 函数配合使

用完成指定定时器输出通道初始化配置。高级控制定时器有四个定时器通道，使用时都必

须单独设置。



(1) TIM\_OCMode：比较输出模式选择，总共有八种，常用的为 PWM1/PWM2。它设定

CCMRx 寄存器 OCxM[2:0]位的值。

(2) TIM\_OutputState：比较输出使能，决定最终的输出比较信号 OCx是否通过外部引脚输

出。它设定 TIMx\_CCER寄存器 CCxE/CCxNE 位的值。

(3) TIM\_OutputNState:比较互补输出使能，决定 OCx的互补信号 OCxN是否通过外部引脚

输出。它设定 CCER 寄存器 CCxNE 位的值。

(4) TIM\_Pulse：比较输出脉冲宽度，实际设定比较寄存器 CCR 的值，决定脉冲宽度。可

设置范围为 0至 65535。

(5) TIM\_OCPolarity：比较输出极性，可选 OCx为高电平有效或低电平有效。它决定着定

时器通道有效电平。它设定 CCER寄存器的 CCxP位的值。

(6) TIM\_OCNPolarity：比较互补输出极性，可选 OCxN 为高电平有效或低电平有效。它

设定 TIMx\_CCER寄存器的 CCxNP 位的值。

(7) TIM\_OCIdleState：空闲状态时通道输出电平设置，可选输出 1 或输出 0，即在空闲状

态(BDTR\_MOE位为 0)时，经过死区时间后定时器通道输出高电平或低电平。它设定

CR2 寄存器的 OISx 位的值。

(8) TIM\_OCNIdleState：空闲状态时互补通道输出电平设置，可选输出 1或输出 0，即在

空闲状态(BDTR\_MOE 位为 0)时，经过死区时间后定时器互补通道输出高电平或低电

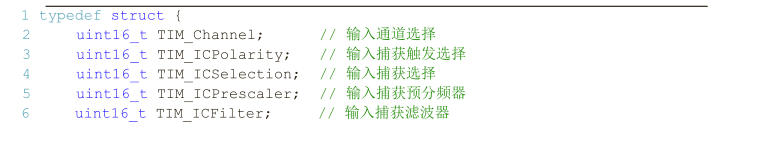
平，设定值必须与 TIM\_OCIdleState 相反。它设定是 CR2 寄存器的 OISxN 位的值。

3. TIM\_ICInitTypeDef

输入捕获结构体 TIM\_ICInitTypeDef 用于输入捕获模式，与 TIM\_ICInit 函数配合使用

完成定时器输入通道初始化配置。如果使用 PWM 输入模式需要与 TIM\_PWMIConfig 函数

配合使用完成定时器输入通道初始化配置。



(1) TIM\_Channel：捕获通道 ICx选择，可选 TIM\_Channel\_1、TIM\_Channel\_2、

TIM\_Channel\_3或 TIM\_Channel\_4四个通道。它设定 CCMRx 寄存器 CCxS 位 的值。

(2) TIM\_ICPolarity：输入捕获边沿触发选择，可选上升沿触发、下降沿触发或边沿跳变触

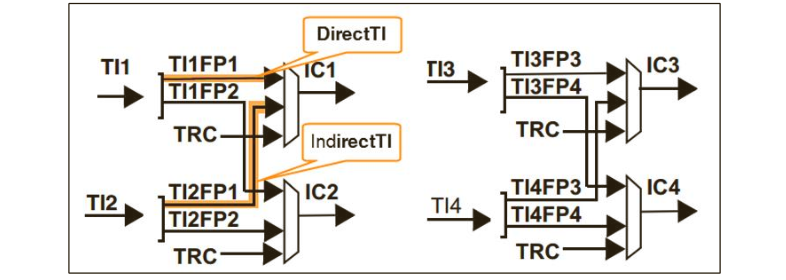
发。它设定 CCER寄存器 CCxP 位和 CCxNP 位的值。

(3) TIM\_ICSelection：输入通道选择，捕获通道 ICx的信号可来自三个输入通道，分别为

TIM\_ICSelection\_DirectTI、TIM\_ICSelection\_IndirectTI或 TIM\_ICSelection\_TRC，具

体的区别见图 32-15。如果是普通的输入捕获，4个通道都可以使用，如果是 PWM输

入则只能使用通道 1 和通道 2。它设定 CCRMx 寄存器的 CCxS[1:0]位的值。



(4) TIM\_ICPrescaler：输入捕获通道预分频器，可设置 1、2、4、8 分频，它设定 CCMRx

寄存器的 ICxPSC[1:0]位的值。如果需要捕获输入信号的每个有效边沿，则设置 1分频

即可。

(5) TIM\_ICFilter：输入捕获滤波器设置，可选设置 0x0至 0x0F。它设定 CCMRx寄存器

ICxF[3:0]位的值。一般我们不使用滤波器，即设置为 0。

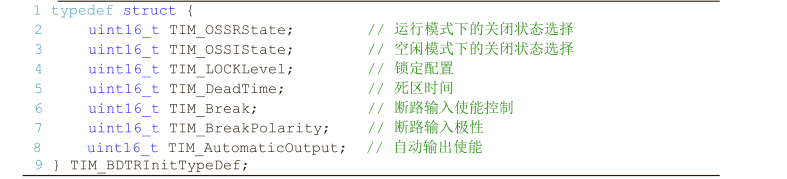
4. TIM\_BDTRInitTypeDef

断路和死区结构体 TIM\_BDTRInitTypeDef 用于断路和死区参数的设置，属于高级定时

器专用，用于配置断路时通道输出状态，以及死区时间。它与 TIM\_BDTRConfig 函数配置

使用完成参数配置。这个结构体的成员只对应 BDTR 这个寄存器，有关成员的具体使用配

置请参考手册 BDTR寄存器的详细描述。



(1) TIM\_OSSRState：运行模式下的关闭状态选择，它设定 BDTR寄存器 OSSR位的值。

(2) TIM\_OSSIState：空闲模式下的关闭状态选择，它设定 BDTR 寄存器 OSSI位的值。

(3) TIM\_LOCKLevel：锁定级别配置， BDTR寄存器 LOCK[1:0]位的值。

(4) TIM\_DeadTime：配置死区发生器，定义死区持续时间，可选设置范围为 0x0至

0xFF。它设定 BDTR 寄存器 DTG[7:0]位的值。

(5) TIM\_Break：断路输入功能选择，可选使能或禁止。它设定 BDTR寄存器 BKE 位的

值。

(6) TIM\_BreakPolarity：断路输入通道 BRK极性选择，可选高电平有效或低电平有效。它

设定 BDTR寄存器 BKP 位的值。

(7) TIM\_AutomaticOutput：自动输出使能，可选使能或禁止，它设定 BDTR寄存器 AOE

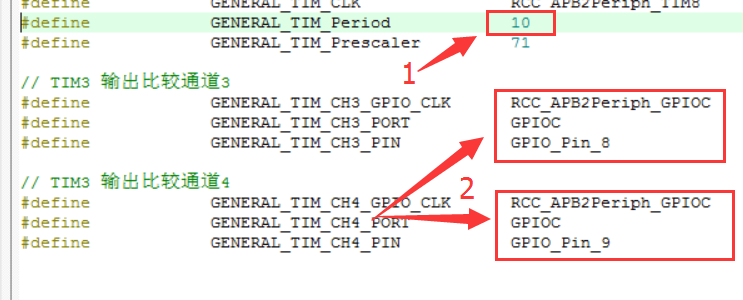
位的值。

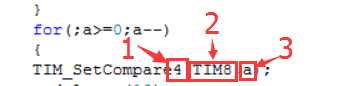
明白了这些 你就很容易明白例程怎么写的了吧 下面做简要讲解 课上会做详细说明

打开main.c 找到main函数（前两个函数用作gpio和简单软件延时的驱动 暂且不用管）



While里的a是传值作为占空比的变量，如果想要让对应可输出管脚输出相应占空比的pwm波，需要改这几个地方



1. 是更改脉冲的周期，现在是（71+1）= 72 分频，时钟为72M 所以计数器记一下时间用1us 这里的10是10us一个周期 你可以在65535之内任意改，比如1ms一个周期的话，你就要改成1000
2. 是更改输出pwm波的对应引脚的 前面gpio部分已经介绍，不过多介绍，强调一下就是你所改的引脚一定要是对应定时器的可输出引脚。
3. 要改变占空比 有很多种方式，这里介绍一种比较简单的
4. 
5. 这个函数中 1箭头指向你要更改的输出通道 可以是1234 ，2 是你所用的定时器 3是你所希望的高电平时间 承接上面的72分频的话，记一下时间为1us 所以如果你输入的a是5，那么就是5us高电平，（10-5）us的低电平