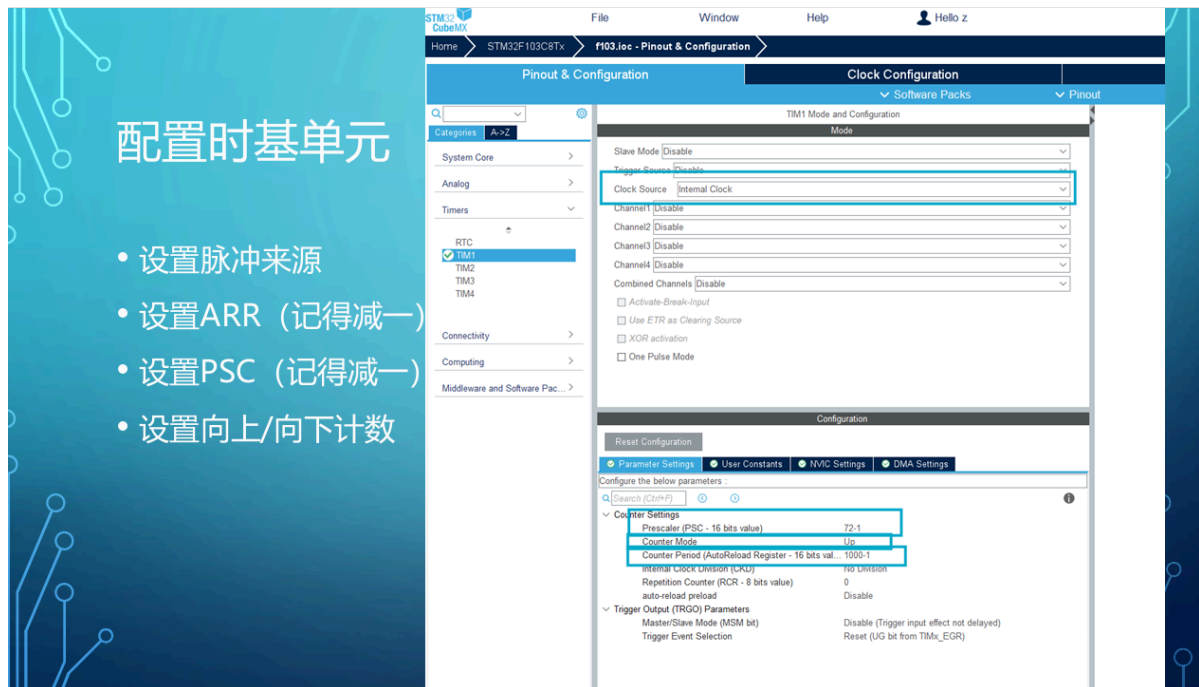


# dk作业3关于PSC和ARR的配置问题说明

看到很多同学的中断频率配置和计数器配置都长下图这样 为防误会作一下说明：



首先其实这张图里面的PSC和ARR其实只是一个配置示范，没有要作为哪个项目实现的教程步骤的意思

然后：

- 1.PSC和ARR是16位寄存器，范围0-65535，频率计算公式参考文档
- 2.中断所用的TIM频率不是这样配的！**如图配置的溢出频率为1ms！**不符合每秒转一次向的要求！！
- 3.延迟函数所用的时基可以这样配，但是要注意一下为什么可以：文档中的微秒延迟函数的关键在于CNT的增长频率必须是1MHz，所以关键在于PSC必须为72-1，但ARR其实不是非得要1000-1，此处对ARR没有特别强调的原因是延迟函数中有这一句-----
- 4.以上的2，3点都建立在时钟树的主频确实是配置为72MHz的基础上

```
void delay_us(uint16_t us)
{
    uint16_t counter = 0;
    __HAL_TIM_SetAutoreload(&htim2, us); //设置ARR值
    __HAL_TIM_SetCounter(&htim2, counter); //设置CNT初始值
    HAL_TIM_Base_Start(&htim2); //启动时基，开始计时

    while(counter != us) //counter是读取CNT所得到的值，即若CNT还没有到所设定的us，则一直计数，直到时间到了再跳出循环
    {
        counter = __HAL_TIM_GetCounter(&htim2); // 获取定时器当前计数
    }
    HAL_TIM_Base_Stop(&htim2); // 停止定时器
}
```

也就是说它其实在代码里手动更改了ARR值，所以在cube里面可以随意。

也可以选择代码里把上面那一行删掉，直接将ARR设为65535（最大限度地防止它溢出）

总而言之请大家务必知道频率都是如何计算的，要知道这些框框里的数为什么是这些数，是不是一定得是这个数，以及假如要实现其它频率的中断，或者其它单位的延时器又该如何配置