# 一、GPT定时器简介

以前的延时函数就采用空指令执行来实现，延时肯定不准确！当我们修改了6ULL的主频以后，采用空指令的延时函数就不准了！因此我们需要高精度的延时函数，而且不随着主频的变化而改变。

STM32使用SYSTICK这个硬件定时器来实现高精度延时，因此我们可以在6U里面使用一个硬件定时器来实现高精度延时。

本讲我们使用6U的GPT定时器来实现高精度延时。

GPT定时器是32位向上计数器。

GPT定时器有捕获的功能。

GPT定时器支持比较输出或中断功能。

GPT定时器有一个12位的分频器。

GPT时钟源可以选择，这里我们使用ipg\_clk=66M作为GPT的时钟源。

GPT定时器有两种工作模式：restart和free-run。

Restart模式下：定时器计数值和比较寄存器OCR的值相等的话定时器就会重新从0开始计时。注意！只有比较通道1才有此功能。

Free-run模式：所有三个输出比较通道都适用。从0开始一直加到0xffffffff，然后重新从0开始，周而复始。

GPT\_CR寄存器，bit0为GPT使能位，为0的时候关闭GPT，为1的时候使能GPT。Bit1确定GPT定时器计数器的初始值，为0的时候表示GPT定时器计数值默认为上次关闭的时候遗留的值，为1的话计数值为0。Bit8~6为时钟源的选择，设置为1，表示GPT时钟源为ipg\_clk=66MHz。bit9设置GPT定时器工作模式，为0的时候工作在restart模式，为1的时候工作在free-run模式。Bit15软件复位。

GPT\_CR寄存器的bit11~0为分频值，可设置0-4095，表示1~4096分频。

GPT\_SR寄存器，bit5表示溢出发生，bit4和bit3分别为输入通道2和1的捕获中断标志位。Bit2~0,也就是OF3~OF1为比较中断。

GPT\_IR寄存器，也就是中断使能寄存器

# 二、GPT定时器使用

# 三、高精度延时原理简介

# 四、实验程序编写