目录

第7章 Cortex-M4-基本定时器

7.1 定时器概述

7.1.1 定时器本质

计数：1234567---数数—逐步累积递增。
定时：有规律地计数就是定时
定时 =要数多少下*数一下要多长时间
10S=1S 数一下，要数 10 下---一秒来一次时钟，就需要计数 10 次。

10S=100MS 数一下，要数 100 下

7.1.2 STM32 的定时器

在 STM32 中，定时器分成了三大类：（片上外设）
基本定时器：主要做基本的定时功能。还可以触发 ADC/DAC 开启转换。
通用定时器：具有基本定时器所有功能，并且还具有比较输出和捕获输入功能。
高级定时器：具有通用定时器所有功能，并且还具有死区功能和刹车功能。

7.1.3 STM32 的基本定时器

主要做定时的功能。

7.1.3.1 基本定时器介绍

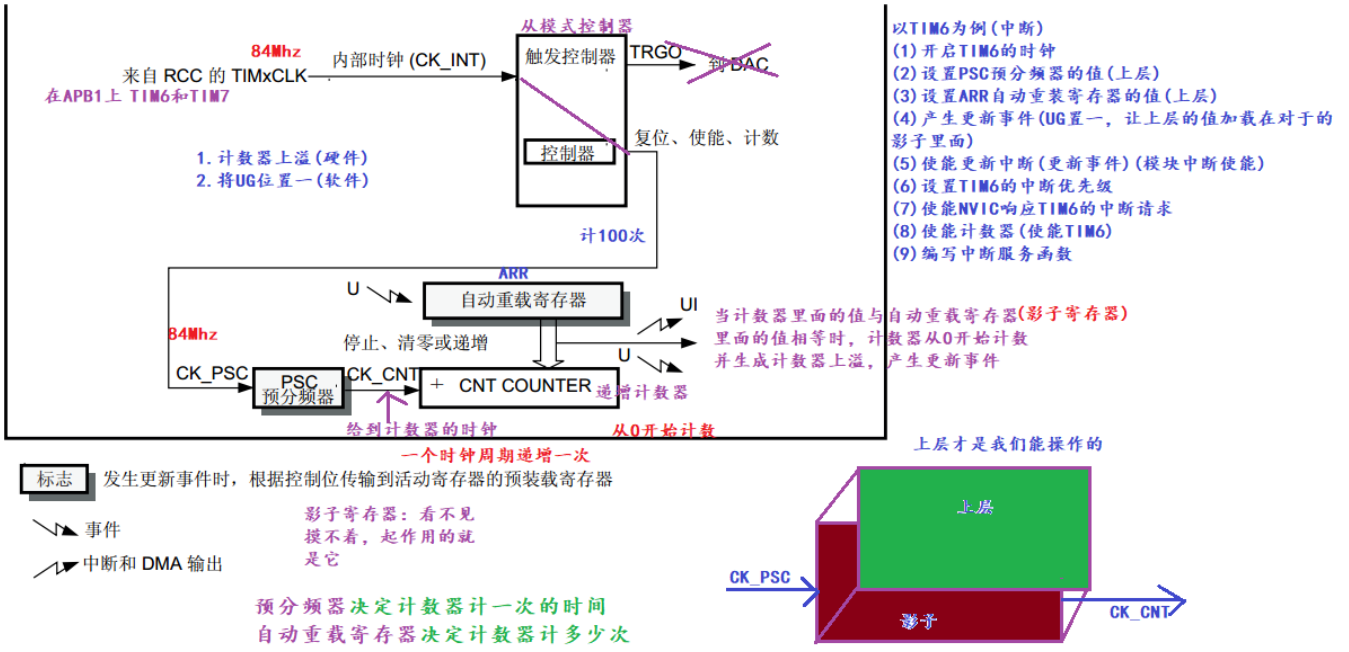
系统嘀嗒定时器的递减计数器 24bit
基本定时器的递增计数器 16bit
基本定时器 TIM6 和 TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。此类定时器不仅可用作通用定时器以生成时基， 还可以专门用于驱动数模转换器 (DAC)。实际上，此类定时器内部连接到 DAC 并能够通过其触发输出驱动 DAC。
这些定时器彼此完全独立，不共享任何资源。
TIM6 1S
TIM7 800ms
USART1 9600
USART2 115200

7.1.3.2 基本定时器特征

基本定时器（TIM6 和 TIM7）的特性包括：
- 16 位自动重载递增计数器
- 16 位可编程预分频器(可以自己决定这个分频系数)，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 和 65536 之间
- 用于触发 DAC 的同步电路
- 发生更新事件时会生成(会让标志位置一)中断请求：计数器上溢

0 → 100
计数器上溢，产生更新事件，就让标志位置起来(时间到了)，如果使能中断，NVIC 就会帮你抢 CPU

7.2 基本定时器框架（重点）



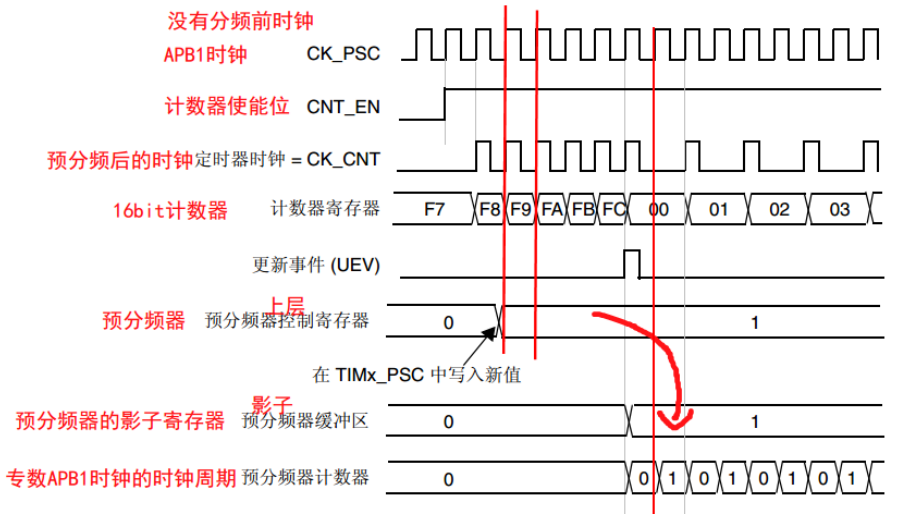
之前系统滴答定时器直接将时钟给递减计数器

现在基本定时器的时钟先经过预分器，再到达递增计数器

7.3 基本定时器定时预装载过程分析

7.3.1 预分频器

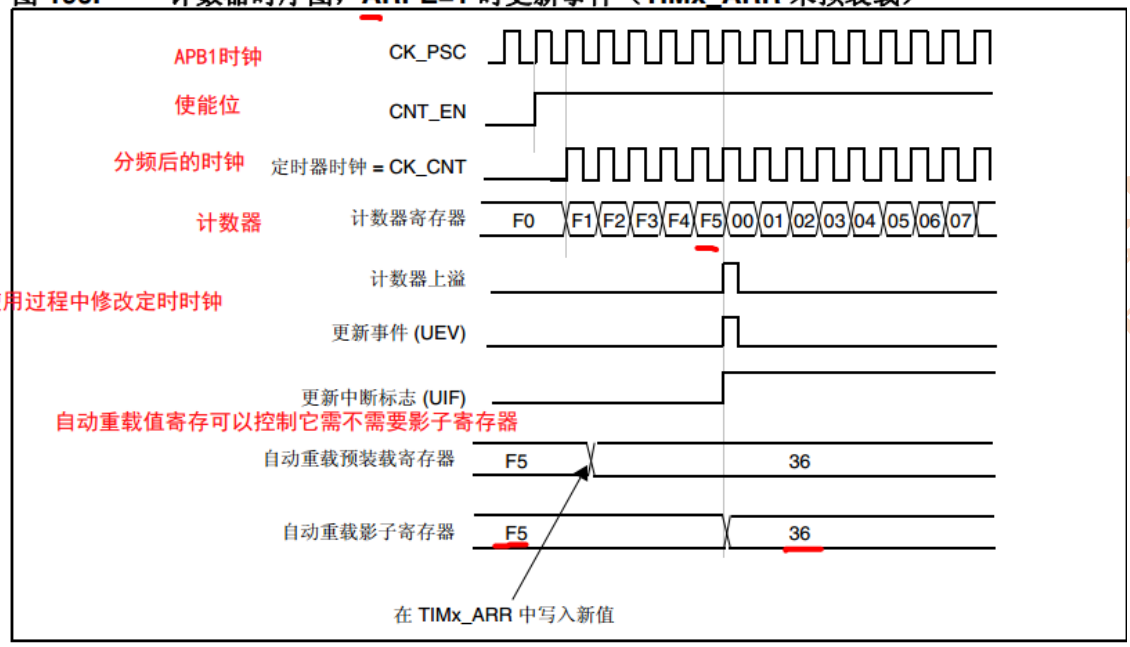
图 189. 预分频器分频由 1 变为 2 时的计数器时序图



7.3.2 自动重载寄存器

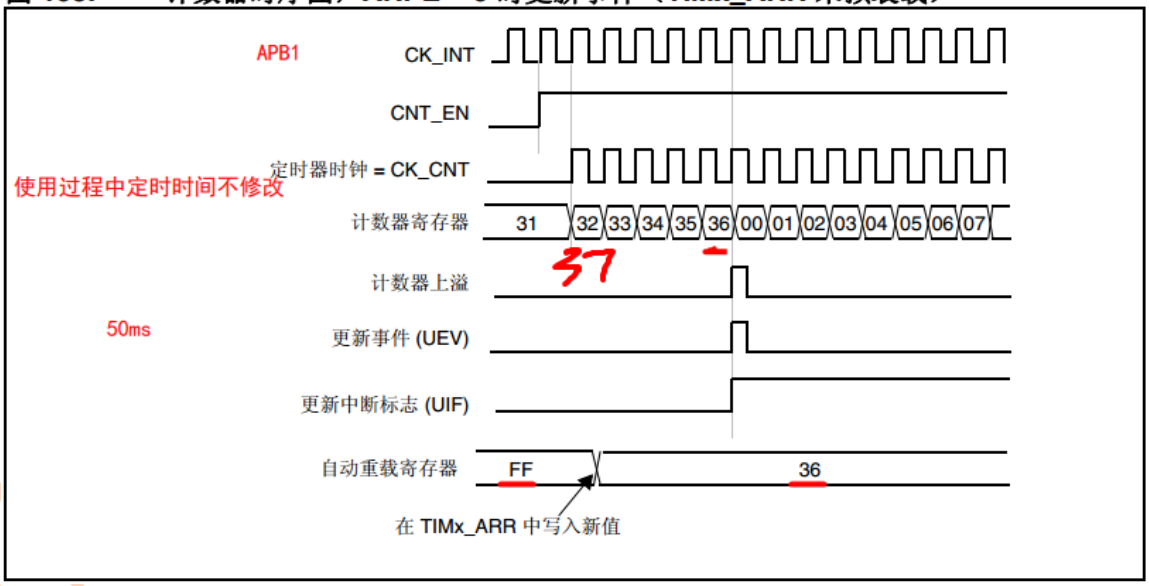
自动重载寄存器进行缓冲

图 196. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 未预装载）



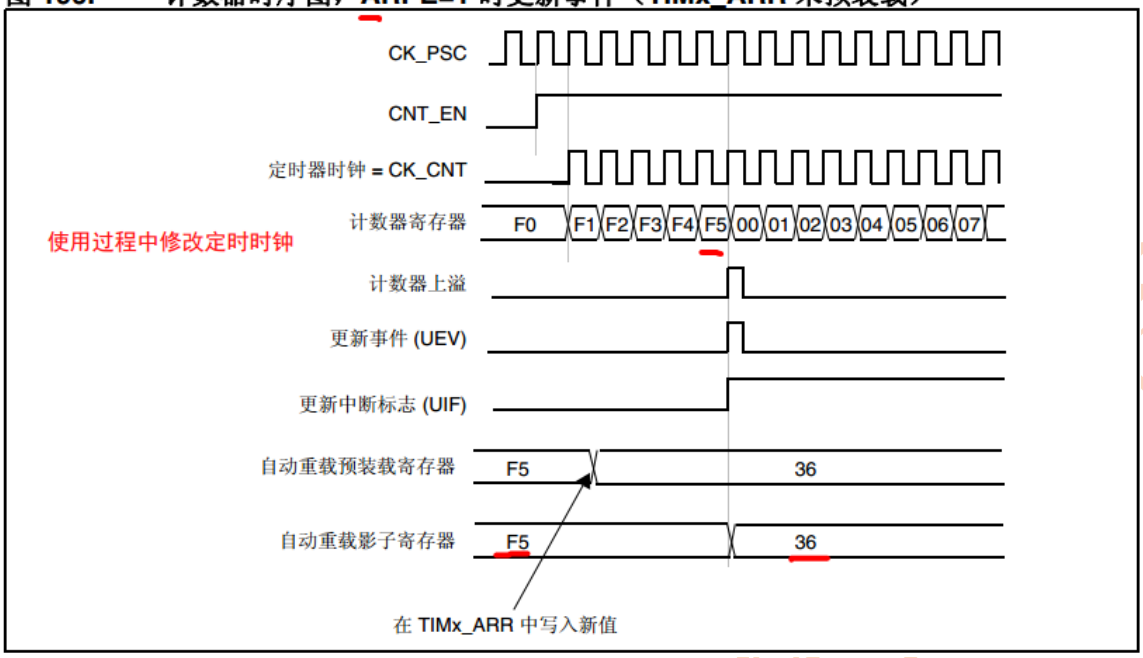
自动重载寄存器不进行缓冲

图 195. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载）



自动重载寄存器进行缓冲

图 196. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 未预装载）



自动重装寄存器哪种情况用影子，哪种情况不需要影子

定时时间需要中途改变 → 需要影子寄存器

定时时间不需要中途改变 → 不需要影子寄存器

自动重装寄存器可以由我们自己决定它有没有影子寄存器

预分频器一定是有影子寄存器

7.4 基本时基单元

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

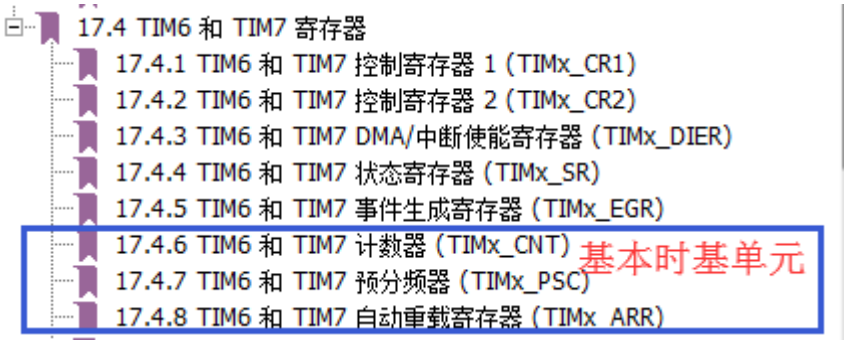
时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

预分器寄存器：决定计数器寄存器计一次的时间

自动重载寄存器：决定了计数器寄存器计多少次

7.5 基本定时器相关寄存器



7.5.1 TIM6 和 TIM7 控制寄存器 1 (TIMx_CR1)

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ARPE	Reserved			OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲。

1：TIMx_ARR 寄存器进行缓冲。

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数 连续模式

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。 单次模式

位 2 **URS**：更新请求源 (Update request source) 更新事件来源的选择

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

- 计数器上溢/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable) 允不允许产生更新事件

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

- 计数器上溢/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR 和 PSC）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

7.5.2 TIM6 和 TIM7 DMA/中断使能寄存器 (TIMx_DIER)

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							UDE	Reserved							UIE
							rw								rw

位 0 **UIE**：更新中断使能 (Update interrupt enable)

0：禁止更新中断。

1：使能更新中断。

7.5.3 TIM6 和 TIM7 状态寄存器 (TIMx_SR)

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															rc_w0

位 15:1 保留，必须保持复位值。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– 上溢或下溢并且当 TIMx_CR1 寄存器中 UDIS = 0 时。

– 当由于 TIMx_CR1 寄存器中 URS = 0 且 UDIS = 0 而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

7.5.4 TIM6 和 TIM7 事件生成寄存器 (TIMx_EGR)

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UG
															w

位 15:1 保留，必须保持复位值。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化定时器计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

7.5.5 TIM6 和 TIM7 计数器 (TIMx_CNT) 给大家说了：1-65536

偏移地址：0x24

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

7.5.6 TIM6 和 TIM7 预分频器 (TIMx_PSC)

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

7.5.7 TIM6 和 TIM7 自动重载寄存器 (TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的详细信息，请参见*第 17.3.1 节：第 484 页的时基单元*。

当自动重载值为空时，计数器不工作。

7.6 基本定时器实验

定时时长=计数一次的时间*计多少数（测试的时候，就发现能达到 1us）

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

假设预分器的值 84 分频，定时 10ms
预分频器里面的值和自动重装寄存器里面的值分别是什么？
预分频从 0 开始　83　　1000 000 分子 1 = 1us
10ms = 1us * 10 000
PSC = 83
ARR = 9999　　从 0 开始计数：10 000　-1= 9999

0.1us　0.1*10 0000

f=84Mhz 要 42 分频　(2Mhz　　T = 1/f = 0.5us)，定时器 20ms

PSC = 41
ARR = 20 000us = 0.5us * 40000 -1= 39999
84 分频
PSC = 83
ARR = 19999
1us *20000

7.6.1 查询方式

延时　　1S 翻转一次 LED1

1. 开启定时器 6 的时钟
2. 配置 CR1 寄存器（不缓冲、连续模式 URS = 1）
3. 设置重载值
4. 设置分频值
5. UG 置一（将上层寄存器里面的值加载到对应影子寄存器中，对计数器清零）
6. 使能定时器(CR1[0] = 1)
7. 等待 UIF 标志位置一
8. 关闭定时器

7.6.2 中断方式

1. 开启定时器 6 的时钟
2. 配置 CR1 寄存器（不缓冲、连续模式 URS = 1）
3. 设置重载值
4. 设置分频值
5. UG 置一（将上层寄存器里面的值加载到对应影子寄存器中，对计数器清零）
6. 使能更新中断 – 模块级的中断使能
7. 设置 TIM6 中断优先级

8. 使能 NVIC 响应请求—核心级中断使能
9. 使能定时器(CR1[0] = 1)
10. 编写中断服务函数

7.6.3 作业

1. 定时器 7 中断点灯--1s (定时 1s)

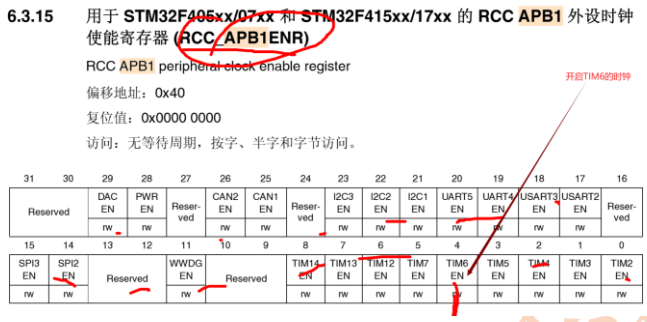
PSC = ?

ARR = ?

1、 基本定时器查询功能实验

第一步： //打开 TIM6 的时钟

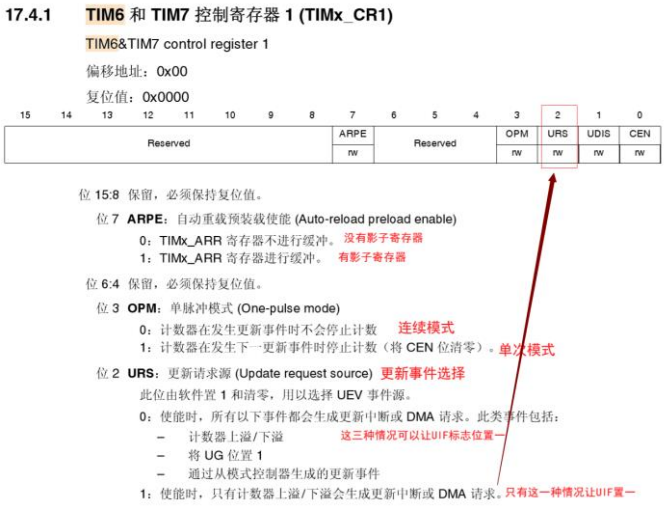
RCC->APB1ENR |= (0x1 << 4);



第二步：//配置 CR1 寄存器

TIM6->CR1 = 0;//整体清零

TIM6->CR1 |= (0x1 << 2);//只有计数器上溢才能让 UIF 置一



第三步： //配置时基单元

TIM6->PSC = psc - 1;//设置预分频值 确定好计一次的时间

17.4.7 TIM6 和 TIM7 预分频器 (TIMx_PSC)

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

TIM6->ARR = arr - 1;//确定计数器计的次数

17.4.8 TIM6 和 TIM7 自动重载寄存器 (TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的详细信息，请参见*第 17.3.1 节：第 484 页的时基单元*。

当自动重载值为空时，计数器不工作。

第四步：//产生更新事件(初始化计数器，将上层寄存器的值加载到影子中 PSC)

TIM6->EGR |= (0x1 << 0);

17.4.5 TIM6 和 TIM7 事件生成寄存器 (TIMx_EGR)

TIM6&TIM7 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UG
															w

位 15:1 保留，必须保持复位值。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化定时器计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

第五步： //使能定时器

TIM6->CR1 |=(0x1 << 0);

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

第六步：//等待 UIF 位标志位置一

```
while( (TIM6->SR & (0x1 << 0)) == 0 );
TIM6->SR &= ~(0x1 << 0);     //对中断标志位清零
```

17.4.4 TIM6 和 TIM7 状态寄存器 (TIMx_SR)

TIM6&TIM7 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															rc_w0

位 15:1 保留，必须保持复位值。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- 上溢或下溢并且当 TIMx_CR1 寄存器中 UDIS = 0 时。
- 当由于 TIMx_CR1 寄存器中 URS = 0 且 UDIS = 0 而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

第 7 步：//关闭定时器

TIM6->CR1 &=~(0x1 << 0);

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

2、 基本定时器中断方式实验

第一步： //打开 TIM6 的时钟

RCC->APB1ENR |= (0x1 << 4);

6.3.15 用于 STM32F405xx/07xx 和 STM32F415xx/17xx 的 RCC APB1 外设时钟使能寄存器 (RCC_APB1ENR)

RCC APB1 peripheral clock enable register

偏移地址：0x40

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问。

开启TIM6的时钟

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	Reser-ved	CAN2 EN	CAN1 EN	Reser-ved	I2C3 EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Reser-ved
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved		TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

第二步：//配置 CR1 寄存器

TIM6->CR1 = 0;//整体清零

TIM6->CR1 |= (0x1 << 2);//只有计数器上溢才能让 UIF 置一

17.4.1 TIM6 和 TIM7 控制寄存器 1 (TIMx_CR1)

TIM6&TIM7 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ARPE	Reserved			OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲。 没有影子寄存器

1：TIMx_ARR 寄存器进行缓冲。 有影子寄存器

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数 连续模式

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。单次模式

位 2 **URS**：更新请求源 (Update request source) 更新事件选择

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢 这三种情况可以让UIF标志位置一

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。只有这一种情况让UIF置一

第三步： //配置时基单元

TIM6->PSC = psc - 1;//设置预分频值 确定好计一次的时间

17.4.7 TIM6 和 TIM7 预分频器 (TIMx_PSC)

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

TIM6->ARR = arr - 1;//确定计数器计的次数

17.4.8 TIM6 和 TIM7 自动重载寄存器 (TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的详细信息，请参见*第 17.3.1 节：第 484 页的时基单元*。

当自动重载值为空时，计数器不工作。

第四步：//产生更新事件(初始化计数器，将上层寄存器的值加载到影子中 PSC)

TIM6->EGR |= (0x1 << 0);

17.4.5 TIM6 和 TIM7 事件生成寄存器 (TIMx_EGR)

TIM6&TIM7 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UG
															w

位 15:1 保留，必须保持复位值。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化定时器计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

第五步： //使能更新中断

TIM6->DIER |= (0x1 << 0);

17.4.3 TIM6 和 TIM7 DMA/中断使能寄存器 (TIMx_DIER)

TIM6&TIM7 DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							UDE	Reserved							UIE
							rw								rw

位 0 **UIE**：更新中断使能 (Update interrupt enable)

0：禁止更新中断。

1：使能更新中断。

第六步：//设置 TIM6 的中断优先级

NVIC_SetPriority(TIM6_DAC_IRQn, NVIC_EncodePriority (7-2, 1, 2));

第七步：//使能 NVIC 响应 TIM6 的中断请求

NVIC_EnableIRQ(TIM6_DAC_IRQn);

第八步：//使能计数器

TIM6->CR1 |= (0x1 << 0);

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

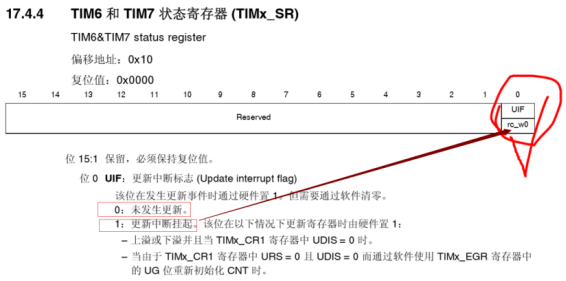
注意：只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

第九步：编写中断服务函数

```
//TIM6 的中断服务函数
void TIM6_DAC_IRQHandler(void)
{
```

```
if( TIM6->SR & (0x1 << 0) )//更新中断标志
{
    TIM6->SR &= ~(0x1 << 0);//清更新中断标志
```




```
    LED1 = !LED1;

    printf("%d\r\n",i);
    i++;
}

}
```