

TIMER

1, 定时器概述 (了解)

1.1, 软件定时原理	使用纯软件 (CPU死等) 的方式实现定时 (延时) 功能	1, 延时不精准 2, CPU死等
1.2, 定时器定时原理	使用精准的时基, 通过硬件的方式, 实现定时功能	
1.3, STM32定时器分类	定时器核心就是计数器	
	基本定时器	
	通用定时器	
	高级定时器	
	独立看门狗	
	窗口看门狗	
	实时时钟	
	低功耗定时器	
	内核定时器	SysTick定时器
1.4, STM32定时器特性表	可参考芯片数据手册或者各个开发板对应的开发指南	
1.5, STM32基本、通用、高级定时器的功能整体区别	基础定时器	没有输入输出通道, 常用作时基, 即定时功能
	通用定时器	具有多路独立通道, 可用于输入捕获/输出比较, 也可用作时基
	除具备通用定时器所有功能外, 还具备带死区控制的互补信号输出、刹车输入等功能 (可用于电机控制、数字电源设计等)	

2, 基本定时器 (掌握)

2.1, 基本定时器简介 (了解)	1, 基本定时器	TIM6/TIM7
	2, 主要特性	16位递增计数器 (计数值: 0~65535) 16位预分频器 (分频系数: 1~65536) 可用于触发DAC 在更新事件 (计数器溢出) 时, 会产生中断/DMA请求
2.2, 基本定时器框图 (熟悉)	时钟源、控制器、时基单元	
2.3, 定时器计数模式及溢出条件 (熟悉)	计数器模式	递增计数模式 溢出条件: CNT==ARR 递减计数模式 溢出条件: CNT==0 中心对齐模式 溢出条件: CNT==ARR-1、CNT==1
2.4, 定时器中断实验相关寄存器 (了解)		
2.5, 定时器溢出时间计算方法 (掌握)	定时器溢出时间计算公式	$T_{out} = \frac{(ARR + 1) * (PSC + 1)}{F_t}$
2.6, 定时器中断实验配置步骤 (掌握)	1, 配置定时器基础工作参数	HAL_TIM_Base_Init()
	2, 定时器基础MSP初始化	HAL_TIM_Base_MspInit() 配置NVIC、CLOCK等
	3, 使能更新中断并启动计数器	HAL_TIM_Base_Start_IT()
	4, 设置优先级, 使能中断	HAL_NVIC_SetPriority(), HAL_NVIC_EnableIRQ()
	5, 编写中断服务函数	TIMx_IRQHandler()等 --> HAL_TIM_IRQHandler()
	6, 编写定时器更新中断回调函数	HAL_TIM_PeriodElapsedCallback()
2.7, 编程实战: 定时器中断实验 (掌握)	使用定时器6, 实现500ms定时器更新中断, 在中断里翻转LEDO	

3, 通用定时器 (掌握)

3.1, 通用定时器简介 (了解)	1, 通用定时器	TIM2/TIM3 /TIM4 /TIM5	F1为例
	2, 主要特性	16位递增、递减、中心对齐计数器 (计数值: 0~65535) 16位预分频器 (分频系数: 1~65536) 可用于触发DAC、ADC 在更新事件、触发事件、输入捕获、输出比较时, 会产生中断/DMA请求 4个独立通道, 可用于: 输入捕获、输出比较、输出PWM、单脉冲模式 使用外部信号控制定时器且可实现多个定时器互连的同步电路 支持编码器和霍尔传感器电路等	
3.2, 通用定时器框图 (熟悉)	时钟源、控制器、时基单元、输入捕获、捕获/比较 (公共)、输出比较		
3.3, 计数器时钟源 (掌握)	①内部时钟(CK_INT), 来自外设总线APB提供的时钟	设置TIMx_SMCR的SMS=000	
	②外部时钟模式1: 外部输入引脚(TIMx), 来自定时器通道1或者通道2引脚的信号	设置TIMx_SMCR的SMS=111	
	③外部时钟模式2: 外部触发输入(ETR), 来自可以复用为TIMx_ETRx的IO引脚	设置TIMx_SMCR的ECE=1	
	④内部触发输入(ITRx), 用于与芯片内部其它通用/高级定时器级联	设置可参考STM32F10xxx参考手册_V10 (中文版).pdf, 14.3.15节	
	外部时钟源模式1		
	外部时钟源模式2		
	使用一个定时器作为另一个定时器的预分频器		
	解读通用定时器中断实验	不同点: 基本定时器只能递增计数, 而通用定时器计数模式有三种	
3.4, 通用定时器PWM输出实验 (掌握)	3.4.1, 通用定时器输出比较部分框图介绍 (熟悉)	捕获/比较通道1的主电路---输出部分 捕获/比较通道的输出部分 (通道1)	
	3.4.2, 通用定时器输出PWM原理 (掌握)	总结: PWM波周期或频率由ARR决定, PWM波占空比由CCRx决定	参考PPT
	3.4.3, PWM模式 (熟悉)	PWM模式1 PWM模式2	参考PPT
	3.4.4, 通用定时器PWM输出实验配置步骤 (掌握)	1, 配置定时器基础工作参数 HAL_TIM_PWM_Init() 2, 定时器PWM输出MSP初始化 HAL_TIM_PWM_MspInit() 配置NVIC、CLOCK、GPIO等 3, 配置PWM模式/比较值等 HAL_TIM_PWM_ConfigChannel() 4, 使能输出并启动计数器 HAL_TIM_PWM_Start() 5, 修改比较值控制占空比(可选) _HAL_TIM_SET_COMPARE() 6, 使能通道预装载(可选) _HAL_TIM_ENABLE_OCxPRELOAD()	
	3.4.5, 编程实战: 通用定时器PWM输出实验 (掌握)	通过定时器输出的PWM控制LEDO, 实现类似手机呼吸灯的效果	
3.5, 通用定时器输入捕获实验 (掌握)	3.5.1, 通用定时器输入捕获部分框图介绍 (熟悉)	捕获/比较通道的输入部分 (通道1) 捕获/比较通道1的主电路---输入部分	
	3.5.2, 通用定时器输入捕获脉冲测量原理 (掌握)	高电平期间, 计时器计数的个数: N * (ARR+1) + CCRx2	参考PPT
	3.5.3, 通用定时器输入捕获实验配置步骤 (掌握)	1, 配置定时器基础工作参数 HAL_TIM_IC_Init() 2, 定时器输入捕获MSP初始化 HAL_TIM_IC_MspInit() 配置NVIC、CLOCK、GPIO等 3, 配置输入通道映射、捕获边沿等 HAL_TIM_IC_ConfigChannel() 4, 设置优先级, 使能中断 HAL_NVIC_SetPriority(), HAL_NVIC_EnableIRQ() 5, 使能定时器更新中断 HAL_TIM_ENABLE_IT() 6, 使能捕获、捕获中断及计数器 HAL_TIM_IC_Start_IT() 7, 编写中断服务函数 TIMx_IRQHandler()等 --> HAL_TIM_IRQHandler() 8, 编写更新中断和捕获回调函数 HAL_TIM_PeriodElapsedCallback() HAL_TIM_IC_CaptureCallback()	
	3.5.4, 编程实战: 通用定时器输入捕获实验 (掌握)	通过定时器5通道1来捕获按键高电平脉宽时间, 通过串口打印出来	
3.6, 通用定时器脉冲计数实验 (掌握)	3.6.1, 脉冲计数实验原理 (熟悉)		参考PPT
	3.6.2, 通用定时器脉冲计数实验配置步骤 (掌握)	1, 配置定时器基础工作参数 HAL_TIM_IC_Init() 2, 定时器输入捕获MSP初始化 HAL_TIM_IC_MspInit() 配置NVIC、CLOCK、GPIO等 3, 配置定时器从模式等 HAL_TIM_SlaveConfigSynchro() 4, 使能输入捕获并启动计数器 HAL_TIM_IC_Start() 5, 获取计数器的值 _HAL_TIM_GET_COUNTER() 6, 设置计数器的值 _HAL_TIM_SET_COUNTER()	
	3.6.3, 编程实战: 通用定时器脉冲计数实验 (掌握)	将定时器2通道1输入的高电平脉冲作为定时器2的时钟, 并通过串口打印脉冲数	

4, 高级定时器 (掌握)

4.1, 高级定时器简介 (了解)

2, 主要特性

1, 高级定时器

TIM1/TIM8

16位递增、递减、中心对齐计数器 (计数值: 0~65535)  
16位预分频器 (分频系数: 1~65536)  
可用于触发DAC、ADC  
在更新事件、触发事件、输入捕获、输出比较时, 会产生中断/DMA请求  
4个独立通道, 可用于: 输入捕获、输出比较、输出PWM、单脉冲模式  
使用外部信号控制定时器且可实现多个定时器互连的同步电路  
支持编码器和霍尔传感器电路等  
重复计数器  
死区时间带可编程的互补输出  
断路输入, 用于将定时器的输出信号置于用户可选的安全配置中

4.2, 高级定时器框图 (熟悉)

重复计数器、输出比较、断路功能

4.3, 高级定时器输出指定个数PWM实验 (掌握)

4.3.1, 重复计数器特性 (熟悉)

4.3.2, 高级定时器输出指定个数PWM实验原理 (掌握)

4.3.3, 高级定时器输出指定个数PWM实验配置步骤 (掌握)

4.3.4, 编程实战: 高级定时器输出指定个数PWM实验 (掌握)

计数器每次上溢或下溢都能使重复计数器减1, 减到0时, 产生更新事件  
如果设置RCR为N, 更新事件将在N+1次溢出发送

1, 配置边沿对齐模式输出PWM  
2, 指定输出N个PWM, 则把N-1写入RCR  
3, 在更新中断内, 关闭计数器

参考PPT

1, 配置定时器基础工作参数  
2, 定时器PWM输出MSP初始化  
3, 配置PWM模式/比较值等  
4, 设置优先级, 使能中断  
5, 使能定时器更新中断  
6, 使能输出、主输出、计数器  
7, 编写中断服务函数  
8, 编写更新中断和捕获回调函数

HAL\_TIM\_PWM\_Init()  
HAL\_TIM\_PWM\_MspInit() 配置NVIC、CLOCK、GPIO等  
HAL\_TIM\_PWM\_ConfigChannel()  
HAL\_NVIC\_SetPriority(), HAL\_NVIC\_EnableIRQ()  
\_HAL\_TIM\_ENABLE\_IT()  
HAL\_TIM\_PWM\_Start()  
TIMx\_IRQHandler()等 --> HAL\_TIM\_IRQHandler()  
HAL\_TIM\_PeriodElapsedCallback()

通过定时器8通道1实现指定个数PWM输出, 用于控制LED1的亮灭

4.4, 高级定时器输出比较模式实验 (掌握)

4.4.1, 高级定时器输出比较模式实验原理 (掌握)

4.4.2, 高级定时器输出比较模式实验配置步骤 (掌握)

4.4.3, 编程实战: 高级定时器输出比较模式实验 (掌握)

总结: PWM波周期或频率由ARR决定, 占空比固定50%, 相位由CCRx决定

参考PPT

1, 配置定时器基础工作参数  
2, 定时器输出比较MSP初始化  
3, 配置输出比较模式等  
4, 使能通道预装载  
5, 使能输出、主输出、计数器  
6, 修改捕获/比较寄存器的值(可选)

HAL\_TIM\_OC\_Init()  
HAL\_TIM\_OC\_MspInit() 配置NVIC、CLOCK、GPIO等  
HAL\_TIM\_OC\_ConfigChannel()  
\_HAL\_TIM\_ENABLE\_OCxPRELOAD()  
HAL\_TIM\_OC\_Start()  
\_HAL\_TIM\_SET\_COMPARE()

通过定时器8通道1/2/3/4输出相位分别为25%、50%、75%、100%的PWM

4.5, 高级定时器互补输出带死区控制实验 (掌握)

4.5.1, 互补输出, 还带死区控制, 什么意思? (了解)

4.5.2, 带死区控制的互补输出应用之H桥 (了解)

4.5.3, 捕获/比较通道的输出部分(通道1至3) (熟悉)

4.5.4, 死区时间计算 (掌握)

4.5.5, 刹车(断路)功能 (熟悉)

4.5.6, 高级定时器互补输出带死区控制实验配置步骤 (掌握)

4.5.7, 编程实战: 高级定时器互补输出带死区控制实验 (掌握)

使能刹车功能: 将TIMx\_BDTR的BKE位置1, 刹车输入信号极性由BKP位设置  
使能刹车功能后: 由TIMx\_BDTR的MOE、OSSI、OSSR位, TIMx\_CR2的OISx、OISxN位, TIMx\_CCR6的CCxE、CCxNE位控制OCx和OCxN输出状态  
无论何时, OCx和OCxN输出都不能同时处在有效电平  
参考: STM32F10xxx参考手册\_V10 (中文版).pdf, 13.4.9节 表75 (其它系列也在CCER后面)  
1, MOE位被清零, OCx和OCxN输出无效、空闲或复位状态(OSSI位选择)  
2, OCx和OCxN的输出状态: 由相关控制位状态决定,  
当使用互补输出时: 根据情况自动控制输出电平, 参考参考手册使用刹车(断路)功能小节  
3, BIF位置1, 如果使能了BIE位, 还会产生刹车中断; 如果使能了TDE位, 会产生DMA请求  
4, 如果AOE位置1, 在下一个更新事件UEV时, MOE位被自动置1

刹车(断路)功能  
发生刹车后, 会怎么样?

1, 配置定时器基础工作参数  
2, 定时器PWM输出MSP初始化  
3, 配置PWM模式/比较值等  
4, 配置刹车功能、死区时间等  
5, 使能输出、主输出、计数器  
6, 使能互补输出、主输出、计数器

HAL\_TIM\_PWM\_Init()  
HAL\_TIM\_PWM\_MspInit() 配置NVIC、CLOCK、GPIO等  
HAL\_TIM\_PWM\_ConfigChannel()  
HAL\_TIMEx\_ConfigBreakDeadTime()  
HAL\_TIM\_PWM\_Start()  
HAL\_TIMEx\_PWMN\_Start()

通过定时器1通道1输出频率为1KHz, 占空比为70%的PWM, 使能互补输出, 并设置死区时间控制: 设置DTG为100等, 进行验证死区时间是否正确  
使能刹车功能: 刹车输入信号高电平有效, 配置输出空闲状态等, 最后用示波器验证

4.6, 高级定时器PWM输入模式实验 (掌握)

4.6.1, PWM输入模式工作原理 (熟悉)

4.6.2, PWM输入模式时序 (熟悉)

4.6.3, 高级定时器PWM输入模式实验配置步骤 (掌握)

4.6.4, 编程实战: 高级定时器PWM输入模式实验 (掌握)

1, 配置定时器基础工作参数  
2, 定时器输入捕获MSP初始化  
3, 配置IC1/2映射、捕获边沿等  
4, 配置从模式, 触发源等  
5, 设置优先级, 使能中断  
6, 使能捕获、更新/捕获中断及计数器  
7, 编写中断服务函数  
8, 编写捕获回调函数

HAL\_TIM\_IC\_Init()  
HAL\_TIM\_IC\_MspInit() 配置NVIC、CLOCK、GPIO等  
HAL\_TIM\_IC\_ConfigChannel()  
HAL\_TIM\_SlaveConfigSynchro()  
HAL\_NVIC\_SetPriority(), HAL\_NVIC\_EnableIRQ()  
HAL\_TIM\_IC\_Start\_IT(), HAL\_TIM\_IC\_Start()  
TIMx\_IRQHandler()等 --> HAL\_TIM\_IRQHandler()  
HAL\_TIM\_IC\_CaptureCallback()

通过定时器3通道2 (PB5) 输出PWM  
将PWM输入到定时器8通道1 (PC6), 测量PWM的频率/周期、占空比等信息

5, 课堂总结 (掌握)