



南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Embedded System and Microcomputer Principle

LAB6 Pulse Width Modulation

2021 Fall
wangq9@mail.sustech.edu.cn



CONTENTS

- 1 PWM Mode Description
- 2 TIM1 Registers
- 3 How to Program
- 4 Practice



01

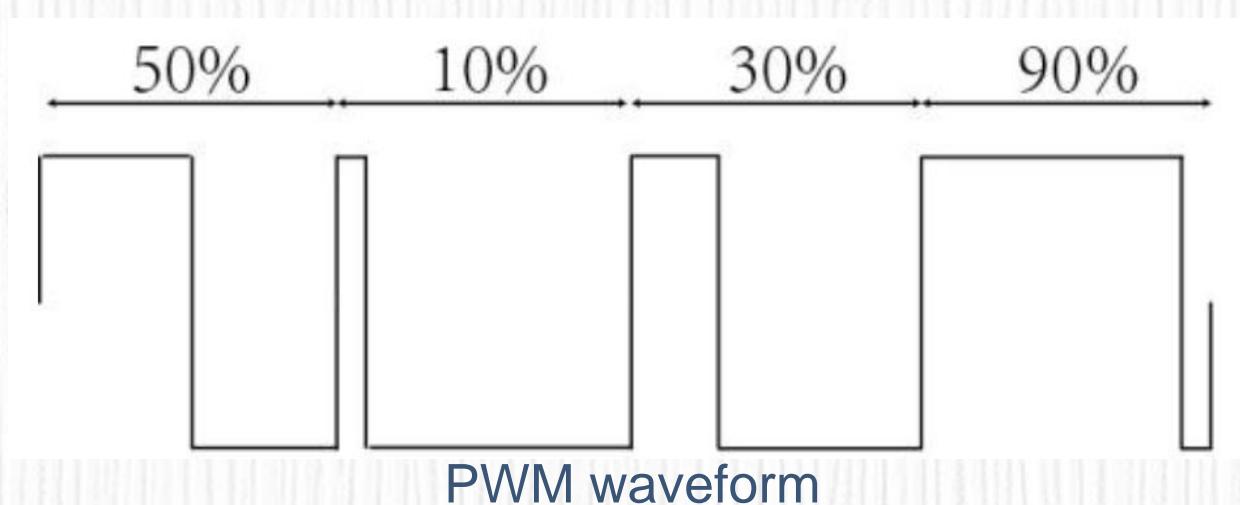
PWM Mode Description



1. PWM Mode Description

-- What is PWM

- 脉冲宽度调制 (Pulse width modulation, PWM)
- 通过对一系列脉冲的宽度进行调制，等效地获得所需要的波形
- 通过改变脉冲列的周期可以调频，改变脉冲的宽度或占空比可以调压，采用适当控制方法即可使电压与频率协调变化



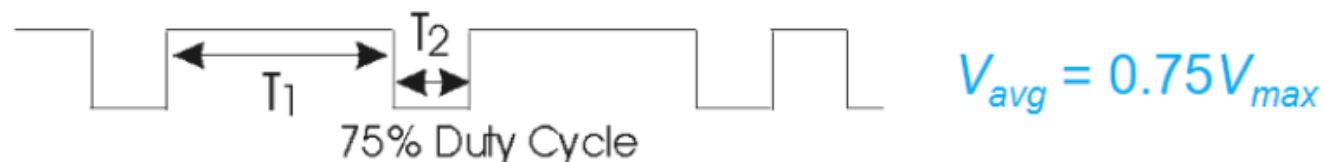


1. PWM Mode Description

-- What is Duty Cycle

T = *period* of waveform (constant)
 T_1 = duration of pulse ($T_2 = T - T_1$)
Duty Cycle = $T_1/T = T_1/(T_1+T_2)$

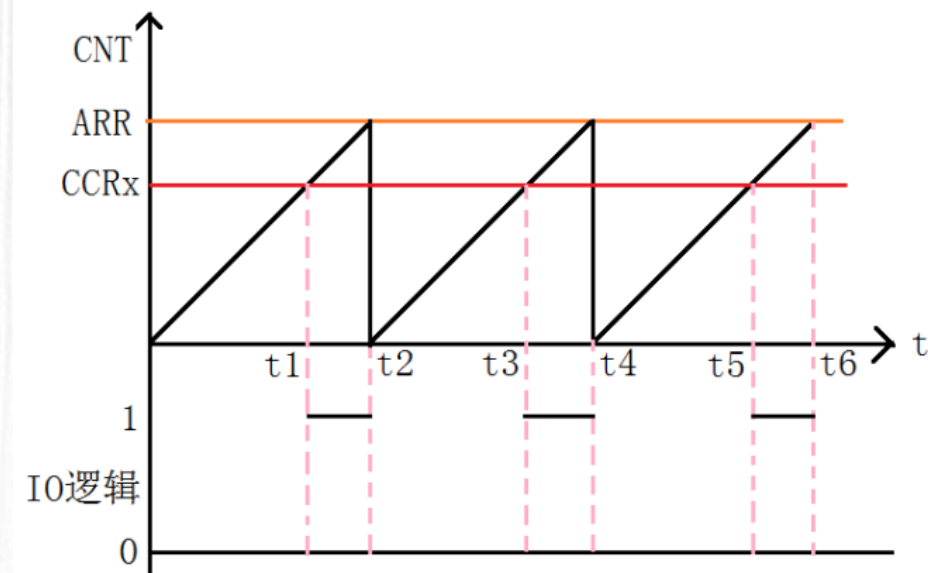
$$V_{avg} = V_{max} \times \text{Duty Cycle}$$



1. PWM Mode Description

-- PWM working principle

- 假定定时器工作在向上计数 PWM 模式，且当 $CNT < CCRx$ 时，输出“0”，当 $CNT \geq CCRx$ 时输出“1”
- 当 CNT 达到 ARR 值的时候，重新归零，再重新向上计数，依次循环
- 改变 CCRx 的值，就可以改变 PWM 输出的占空比
- 改变 ARR 的值，就可以改变 PWM 输出的频率



Schematic diagram of PWM

1. PWM Mode Description

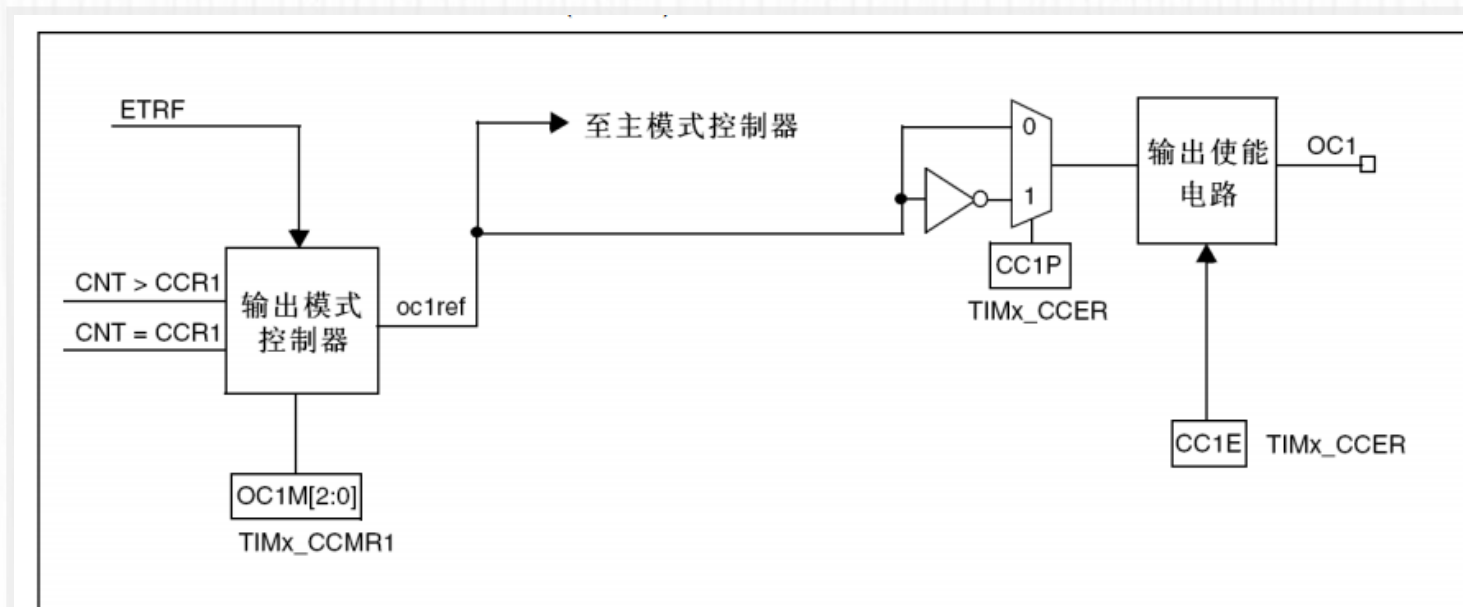
-- PWM working process

- 2个阶段

- 数值比较
- 控制输出

- 涉及寄存器

- TIMx_ARR
- TIMx_PSC
- TIMx_CNT
- TIMx_CCRx
- TIMx_CCMRx
- TIMx_CCER
- TIMx_BDTR(TIM1、TIM8需要)



PWM working process
(take channel 1 as an example)

1. PWM Mode Description

-- PWM working process(continued)

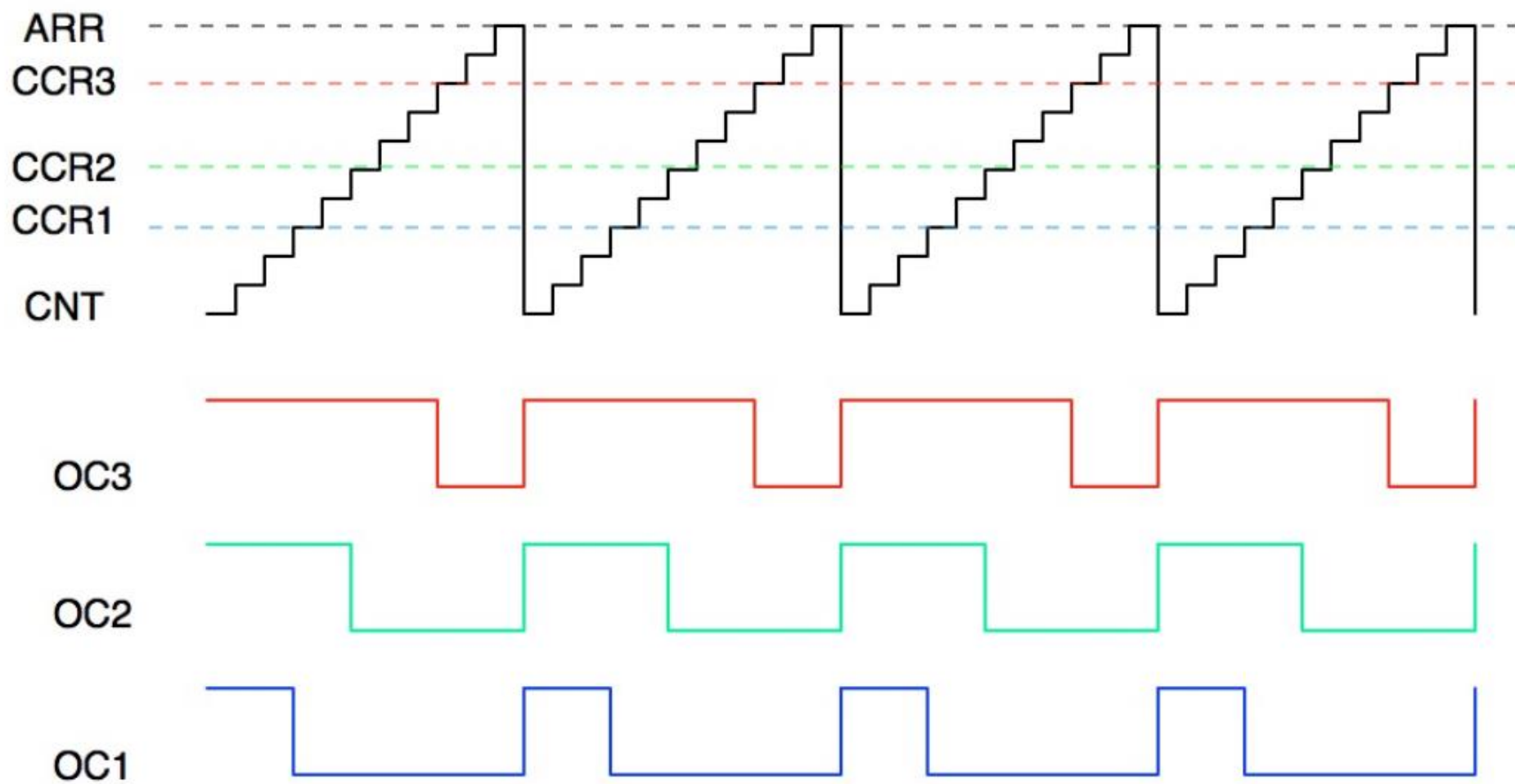
寄存器	位	说明	例子
TIM1_CR1	DIR	0: 计数器向上计数 1: 计数器向下计数	<p>(1) DIR = 0 -> OC1M = 110 -> TIM1_CNT < TIM1_CCR1 -> 通道1为有效电平 (OC1REF=1) -> CCIP = 0 -> OC1输出为高电平;</p> <p>(2) DIR = 1 -> OC1M = 110 -> TIM1_CNT > TIM1_CCR1 -> 通道1为无效电平 (OC1REF=0) -> CCIP = 1 -> OC1输出为高电平;</p> <p>(3) DIR = 0 -> OC1M = 111 -> TIM1_CNT < TIM1_CCR1 -> 通道1为无效电平 (OC1REF=0) -> CCIP = 0 -> OC1输出为低电平;</p> <p>(4) DIR = 1 -> OC1M = 111 -> TIM1_CNT > TIM1_CCR1 -> 通道1为有效电平 (OC1REF=1) -> CCIP = 1 -> OC1输出为低电平;</p>
TIM1_CCMR1	OC1M	110: PWM模式1 111: PWM模式2	
TIM1_CCER	CCIP	0: OC1高电平有效 1: OC1低电平有效	
TIM1_CCER	CC1E	0: 关闭, OC1禁止输出 1: 开启, OC1信号输出到对应引脚	



1. PWM Mode Description

-- PWM waveform examples

Three PWM signals from the Output Compare Channels of a general purpose timer





1. PWM Mode Description

-- PWM configuration steps

- 使能定时器和相关IO口时钟
- 初始化定时器：ARR,PSC等
- 初始化输出比较参数
- 使能预装载寄存器
- 使能定时器
- 不断改变比较值CCR_x，达到不同的占空比效果



02

TIM1 Registers



2. TIM1 Registers

-- TIM1_CR1

- TIM1 control register 1
- 控制寄存器1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKD[1:0]	ARPE		CMS[1:0]	DIR	OPM	URS	UDIS	CEN	
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位6:5	<p>CMS[1:0]: 选择中央对齐模式 (Center-aligned mode selection)</p> <p>00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。</p> <p>01: 中央对齐模式1。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx寄存器中CCxS=00)的输出比较中断标志位, 只在计数器向下计数时被设置。</p> <p>10: 中央对齐模式2。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx寄存器中CCxS=00)的输出比较中断标志位, 只在计数器向上计数时被设置。</p> <p>11: 中央对齐模式3。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx寄存器中CCxS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被设置。</p> <p>注: 在计数器开启时(CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。</p>
位4	<p>DIR: 方向 (Direction)</p> <p>0: 计数器向上计数;</p> <p>1: 计数器向下计数。</p> <p>注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。</p>



2. TIM1 Registers

-- TIM1_CCRx

- TIM1 capture/compare register ($x = 1/2/3/4$)
- 捕获/比较寄存器
- 该寄存器总共有 4 个，对应 4 个输通道 CH1~4
- 在输出模式下，该寄存器的值与 CNT 的值比较，根据比较结果产生相应动作。利用这点，我们通过修改这个寄存器的值，就可以控制 PWM 的输出脉宽了。



2. TIM1 Registers

-- TIM1_CCR1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
位15:0			<p>CCR1[15:0]: 捕获/比较1的值</p> <p>若CC1通道配置为输出:</p> <p>CCR1包含了装入当前捕获/比较1寄存器的值(预装载值)。</p> <p>如果在TIMx_CCMR1寄存器(OC1PE位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。</p> <p>当前捕获/比较寄存器参与同计数器TIMx_CNT的比较, 并在OC1端口上产生输出信号。</p> <p>若CC1通道配置为输入:</p> <p>CCR1包含了由上一次输入捕获1事件(IC1)传输的计数器值。</p>												



2. TIM1 Registers

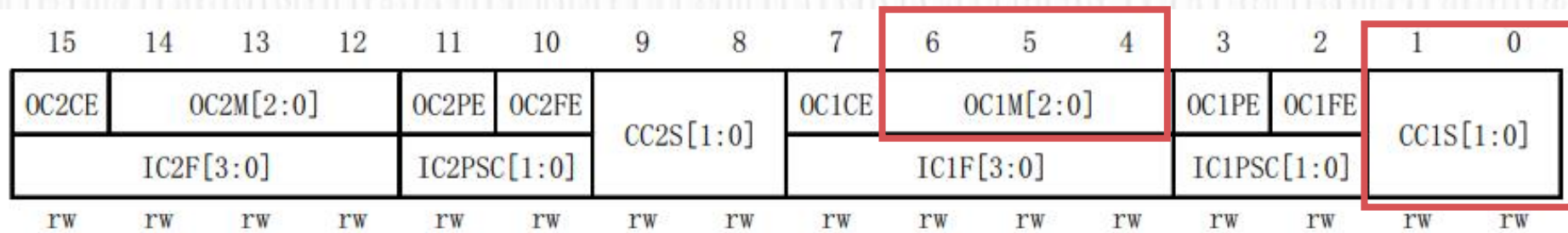
-- TIM1_CCMR1/2

- TIM1 capture/compare mode register 1/2
- 捕获/比较模式寄存器
- TIM1_CCMR1控制CH1和2，而TIM1_CCMR2控制CH3和4
- 通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的CCxS位定义。该寄存器其它位的作用在输入和输出模式下不同。
- OCxx描述了通道在输出模式下的功能， ICxx描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。



2. TIM1 Registers

-- TIM1_CCMR1



输出比较模式:

位15	OC2CE: 输出比较2清0使能 (Output Compare 2 clear enable)
位14:12	OC2M[2:0]: 输出比较2模式 (Output Compare 2 mode)
位11	OC2PE: 输出比较2预装载使能 (Output Compare 2 preload enable)
位10	OC2FE: 输出比较2快速使能 (Output Compare 2 fast enable)
位9:8	CC2S[1:0]: 捕获/比较2选择。(Capture/Compare 2 selection) 该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2上; 10: CC2通道被配置为输入, IC2映射在TI1上; 11: CC2通道被配置为输入, IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC2S仅在通道关闭时(TIMx_CCER寄存器的CC2E=0)才是可写的。



2. TIM1 Registers

-- TIM1_CCMR1 (continued)

位7	OC1CE : 输出比较1清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受ETRF输入的影响; 1: 一旦检测到ETRF输入高电平, 清除OC1REF=0。	位3	OC1PE : 输出比较1预装载使能 (Output Compare 1 preload enable) 0: 禁止TIMx_CCR1寄存器的预装载功能, 可随时写入TIMx_CCR1寄存器, 并且新写入的数值立即起作用。 1: 开启TIMx_CCR1寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1的预装载值在更新事件到来时被加载至当前寄存器中。 注1: 一旦LOCK级别设为3(TIMx_BDTR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出)则该位不能被修改。 注2: 仅在单脉冲模式下(TIMx_CR1寄存器的OPM=1), 可以在未确认预装载寄存器情况下使用PWM模式, 否则其动作不确定。
位6:4	OC1M[2:0] : 输出比较1模式 (Output Compare 1 mode) 该3位定义了输出参考信号OC1REF的动作, 而OC1REF决定了OC1、OC1N的值。OC1REF是高电平有效, 而OC1、OC1N的有效电平取决于CC1P、CC1NP位。 000: 冻结。输出比较寄存器TIMx_CCR1与计数器TIMx_CNT间的比较对OC1REF不起作用; 001: 匹配时设置通道1为有效电平。当计数器TIMx_CNT的值与捕获/比较寄存器1(TIMx_CCR1)相同时, 强制OC1REF为高。 010: 匹配时设置通道1为无效电平。当计数器TIMx_CNT的值与捕获/比较寄存器1(TIMx_CCR1)相同时, 强制OC1REF为低。 011: 翻转。当TIMx_CCR1=TIMx_CNT时, 翻转OC1REF的电平。 100: 强制为无效电平。强制OC1REF为低。 101: 强制为有效电平。强制OC1REF为高。 110: PWM模式1— 在向上计数时, 一旦TIMx_CNT<TIMx_CCR1时通道1为有效电平, 否则为无效电平; 在向下计数时, 一旦TIMx_CNT>TIMx_CCR1时通道1为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM模式2— 在向上计数时, 一旦TIMx_CNT<TIMx_CCR1时通道1为无效电平, 否则为有效电平; 在向下计数时, 一旦TIMx_CNT>TIMx_CCR1时通道1为有效电平, 否则为无效电平。 注1: 一旦LOCK级别设为3(TIMx_BDTR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出)则该位不能被修改。 注2: 在PWM模式1或PWM模式2中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时, OC1REF电平才改变。	位2	OC1FE : 输出比较1快速使能 (Output Compare 1 fast enable) 该位用于加快CC输出对触发输入事件的响应。 0: 根据计数器与CCR1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCFE只在通道被配置成PWM1或PWM2模式时起作用。
		位1:0	CC1S[1:0] : 捕获/比较1 选择。(Capture/Compare 1 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1通道被配置为输出; 01: CC1通道被配置为输入, IC1映射在TI1上; 10: CC1通道被配置为输入, IC1映射在TI2上; 11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC1S仅在通道关闭时(TIMx_CCER寄存器的CC1E=0)才是可写的。



2. TIM1 Registers

-- TIM1_CCER

- TIM1 capture/compare enable register
- 捕获/比较使能寄存器
- 该寄存器控制着各个输入输出通道的开关



2. TIM1 Registers

-- TIM1_CCER(continued)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位15:14	保留，始终读为0。
位13	CC4P ：输入/捕获4输出极性 (Capture/Compare 4 output polarity) 参考CC1P的描述。
位12	CC4E ：输入/捕获4输出使能 (Capture/Compare 4 output enable) 参考CC1E 的描述。
位11	CC3NP ：输入/捕获3互补输出极性 (Capture/Compare 3 complementary output polarity) 参考CC1NP的描述。
位10	CC3NE ：输入/捕获3互补输出使能 (Capture/Compare 3 complementary output enable) 参考CC1NE的描述。
位9	CC3P ：输入/捕获3输出极性 (Capture/Compare 3 output polarity) 参考CC1P的描述。



2. TIM1 Registers

-- TIM1_CCER(continued)

位8	CC3E : 输入/捕获3输出使能 (Capture/Compare 3 output enable) 参考CC1E 的描述。
位7	CC2NP : 输入/捕获2互补输出极性 (Capture/Compare 2 complementary output polarity) 参考CC1NP的描述。
位6	CC2NE : 输入/捕获2互补输出使能 (Capture/Compare 2 complementary output enable) 参考CC1NE的描述。
位5	CC2P : 输入/捕获2输出极性 (Capture/Compare 2 output polarity) 参考CC1P的描述。
位4	CC2E : 输入/捕获2输出使能 (Capture/Compare 2 output enable) 参考CC1E的描述。
位3	CC1NP : 输入/捕获1互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N高电平有效; 1: OC1N低电平有效。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为3或2且CC1S=00(通道配置为输出)则该位不能被修改。
位2	CC1NE : 输入/捕获1互补输出使能 (Capture/Compare 1 complementary output enable) 0: 关闭 — OC1N禁止输出, 因此OC1N的电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。 1: 开启 — OC1N信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。

位1	CC1P : 输入/捕获1输出极性 (Capture/Compare 1 output polarity) CC1通道配置为输出: 0: OC1高电平有效; 1: OC1低电平有效。 CC1通道配置为输入: 该位选择是IC1还是IC1的反相信号作为触发或捕获信号。 0: 不反相: 捕获发生在IC1的上升沿; 当用作外部触发器时, IC1不反相。 1: 反相: 捕获发生在IC1的下降沿; 当用作外部触发器时, IC1反相。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为3或2, 则该位不能被修改。
位0	CC1E : 输入/捕获1输出使能 (Capture/Compare 1 output enable) CC1通道配置为输出: 0: 关闭 — OC1禁止输出, 因此OC1的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。 1: 开启 — OC1信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。 CC1通道配置为输入: 该位决定了计数器的值是否能捕获入TIMx_CCR1寄存器。 0: 捕获禁止; 0: 捕获使能。



2. TIM1 Registers

-- TIM1_BDTR

- TIM1 break and dead-time register
- 刹车和死区寄存器
- 如果使用通用定时器，则无须配置该寄存器
- 如果使用高级定时器，则需要配置该寄存器



2. TIM1 Registers

-- TIM1_BDTR(continued)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]	DTG[7:0]								
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

注释: 根据锁定设置, AOE、BKP、BKE、OSSI、OSSR和DTG[7:0]位均可被写保护, 有必要在第一次写入TIMx_BDTR寄存器时对它们进行配置。

位15	MOE: 主输出使能 (Main output enable) 一旦刹车输入有效, 该位被硬件异步清'0'。根据AOE位的设置值, 该位可以由软件清'0'或被自动置1。它仅对配置为输出的通道有效。 0: 禁止OC和OCN输出或强制为空闲状态; 1: 如果设置了相应的使能位(TIMx_CCER寄存器的CCxE、CCxNE位), 则开启OC和OCN输出。 有关OC/OCN使能的细节, 参见13.4.9节, TIM1和TIM8捕获/比较使能寄存器(TIMx_CCER)。
位14	AOE: 自动输出使能 (Automatic output enable) 0: MOE只能被软件置'1'; 1: MOE能被软件置'1'或在下一个更新事件被自动置'1'(如果刹车输入无效)。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为'1', 则该位不能被修改。
位13	BKP: 刹车输入极性 (Break polarity) 0: 刹车输入低电平有效; 1: 刹车输入高电平有效。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为'1', 则该位不能被修改。 注: 任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。



2. TIM1 Registers

-- TIM1_BDTR(continued)

位12	BKE: 刹车功能使能 (Break enable) 0: 禁止刹车输入(BRK及CCS时钟失效事件); 1: 开启刹车输入(BRK及CCS时钟失效事件)。 注: 当设置了LOCK级别1时(TIMx_BDTR寄存器中的LOCK位), 该位不能被修改。 注: 任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。	位9:8	LOOK[1:0]: 锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别1, 不能写入TIMx_BDTR寄存器的DTG、BKE、BKP、AOE位和TIMx_CR2寄存器的OISx/OISxN位; 10: 锁定级别2, 不能写入锁定级别1中的各位, 也不能写入CC极性位(一旦相关通道通过CCxS位设为输出, CC极性位是TIMx_CCER寄存器的CCxP/CCxN位)以及OISR/OSSI位; 11: 锁定级别3, 不能写入锁定级别2中的各位, 也不能写入CC控制位(一旦相关通道通过CCxS位设为输出, CC控制位是TIMx_CCMRx寄存器的OCxM/OCxPE位); 注: 在系统复位后, 只能写一次LOCK位, 一旦写入TIMx_BDTR寄存器, 则其内容冻结直至复位。
位11	OSSR: 运行模式下“关闭状态”选择 (Off-state selection for Run mode) 该位用于当MOE=1且通道为互补输出时。没有互补输出的定时器中不存在OSSR位。 参考OC/OCN使能的详细说明(13.4.9节, TIM1和TIM8捕获/比较使能寄存器(TIMx_CCER))。 0: 当定时器不工作时, 禁止OC/OCN输出(OC/OCN使能输出信号=0); 1: 当定时器不工作时, 一旦CCxE=1或CCxNE=1, 首先开启OC/OCN并输出无效电平, 然后置OC/OCN使能输出信号=1。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为2, 则该位不能被修改。	位7:0	UTG[7:0]: 死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设DT表示其持续时间: $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times T_{dtg}, T_{dtg} = T_{DTS};$ $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times T_{dtg}, T_{dtg} = 2 \times T_{DTS};$ $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times T_{dtg}, T_{dtg} = 8 \times T_{DTS};$ $DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times T_{dtg}, T_{dtg} = 16 \times T_{DTS};$ 例: 若 $T_{DTS} = 125ns(8MHZ)$, 可能的死区时间为: 0到15875ns, 若步长时间为125ns; 16us到31750ns, 若步长时间为250ns; 32us到63us, 若步长时间为1us; 64us到126us, 若步长时间为2us; 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为1、2或3, 则不能修改这些位。
位10	OSSI: 空闲模式下“关闭状态”选择 (Off-state selection for Idle mode) 该位用于当MOE=0且通道设为输出时。 参考OC/OCN使能的详细说明(13.4.9节, TIM1和TIM8捕获/比较使能寄存器(TIMx_CCER))。 0: 当定时器不工作时, 禁止OC/OCN输出(OC/OCN使能输出信号=0); 1: 当定时器不工作时, 一旦CCxE=1或CCxNE=1, OC/OCN首先输出其空闲电平, 然后OC/OCN使能输出信号=1。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为2, 则该位不能被修改。		



03

How to Program



3. How to Program

- Our Goal
 - Use PWM signal to change the brightness of LED0
 - If we change the duty cycle of the PWM signal, we can change the average voltage, so as to change the brightness of the LED0



3. How to Program

- Choose the TIMER
 - Since LED0 is connected to TIM1_CH1, we should enable TIM1 and configure the channel 1 of TIM1 as PWM Generation CH1
 - Set the CH Polarity as low, because the LED is on only when the voltage is low

PA10/U1_RX/TIM1_CH3	43	PA10	U1_RXD
PA9/U1_TX/TIM1_CH2	42	PA9	U1_TXD
PA8/TIM1_CH1/MCO	41	PA8	LED0
PC9/TIM8_CH4/SDIO_D1	40	PC9	LCD_CS

3. How to Program

- Configure TIM1

TIM1 Mode and Configuration

Categories: A->Z

- System Core >
- Analog >
- Timers >

RTC

- TIM1**
- TIM2
- TIM3
- TIM4
- TIM5
- TIM6
- TIM7
- TIM8

Connectivity >

Multimedia >

Computing >

Mode

Trigger Source: Disable

Clock Source: Internal Clock

Channel1: PWM Generation CH1

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings

Configure the below parameters :

Search (Ctrl+F)

- > Counter Settings
- > Trigger Output (TRGO) Parameters
- > Break And Dead Time management - BRK Config...
- > Break And Dead Time management - Output Conf...
- > PWM Generation Channel 1
 - Mode: PWM mode 1
 - Pulse (16 bits value): 0
 - Output compare preload: Enable
 - Fast Mode: Disable
 - CH Polarity: Low
 - CH Idle State: Reset

TIM1 Mode and Configuration

Categories: A->Z

- System Core >
- Analog >
- Timers >

RTC

- TIM1**
- TIM2
- TIM3
- TIM4
- TIM5
- TIM6
- TIM7
- TIM8

Connectivity >

Mode

Trigger Source: Disable

Clock Source: Internal Clock

Channel1: PWM Generation CH1

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings

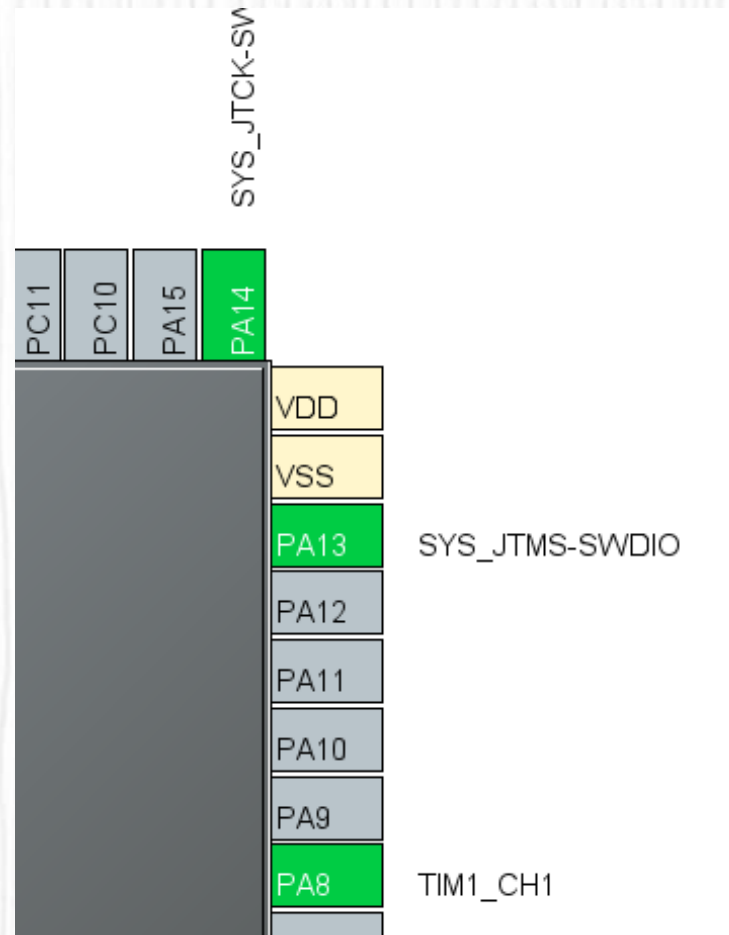
Configure the below parameters :

Search (Ctrl+F)

- > Counter Settings
 - Prescaler (PSC - 16 bits value): 0
 - Counter Mode: Up
 - Counter Period (AutoReload Register - 16 ... 899): 899
 - Internal Clock Division (CKD): No Division
 - Repetition Counter (RCR - 8 bits value): 0
 - auto-reload preload: Disable

3. How to Program

- Configure GPIO
 - PA8 is set as TIM1_CH1





3. How to Program

- Some functions we used
 - Call HAL_TIM_PWM_Start() function to start the PWM signal generation in main.c (after the initialization of TIM1)
 - Change CCR1 value to change the duty cycle of TIM1_CH1 output

```
HAL_StatusTypeDef HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim, uint32_t Channel)
```



3. How to Program

- Run the demo, the LED0 will gradually brighten and darken periodically
- The maximum duty cycle is set to 33% is because the relationship of brightness and voltage is not linear and when the duty cycle is 33%, the LED is bright enough.

```
int main(void)
{
    /* MCU Configuration-----
    /* Reset of all peripherals, Initializes the
    HAL_Init();
    /* Configure the system clock */
    SystemClock_Config();
    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_TIM1_Init();
    /* USER CODE BEGIN 2 */
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
    int mode = 1;
    int pulse = 0;
    /* USER CODE END 2 */

    /* Infinite loop */
    while (1)
    {
        /* USER CODE BEGIN 3 */
        HAL_Delay(10);
        if (mode) {
            pulse++;
        } else {
            pulse--;
        }
        if (pulse > 300) {
            mode = 0;
        }
        if (pulse <= 0) {
            mode = 1;
        }
        TIM1->CCR1 = pulse;
    }
    /* USER CODE END 3 */
}
```




04

Practice

4. Practice

- Run the demo on MiniSTM32 board
- Complete the quiz on Sakai site
- Complete the assignment on Sakai site and submit the whole project before DDL