

《手把手教你学STM32》



主讲人：正点原子团队
硬件平台：正点原子STM32开发板
版权所有：广州市星翼电子科技有限公司
淘宝店铺：<http://eboard.taobao.com>
技术论坛：www.openedv.com 开源电子网
公众平台：“正点原子”
官方网站：www.alientek.com
联系电话：13922348612

ALIENTEK



《手把手教你学STM32》



■ 定时器输入捕获实验

适用平台

~~✓ STM32F1xx
开发板
(正点原子)~~

✓ STM32F4xx
开发板
(正点原子)



✓ 输入捕获实验



■ 参考资料:

● 探索者STM32F4开发板

《STM32F4开发指南-库函数版本》-第15章 输入捕获实验

□ STM32F4xx官方资料:

《STM32F4中文参考手册》-第15章 通用定时器

目录



1

通用定时器输入捕获概述

2

常用寄存器和库函数配置

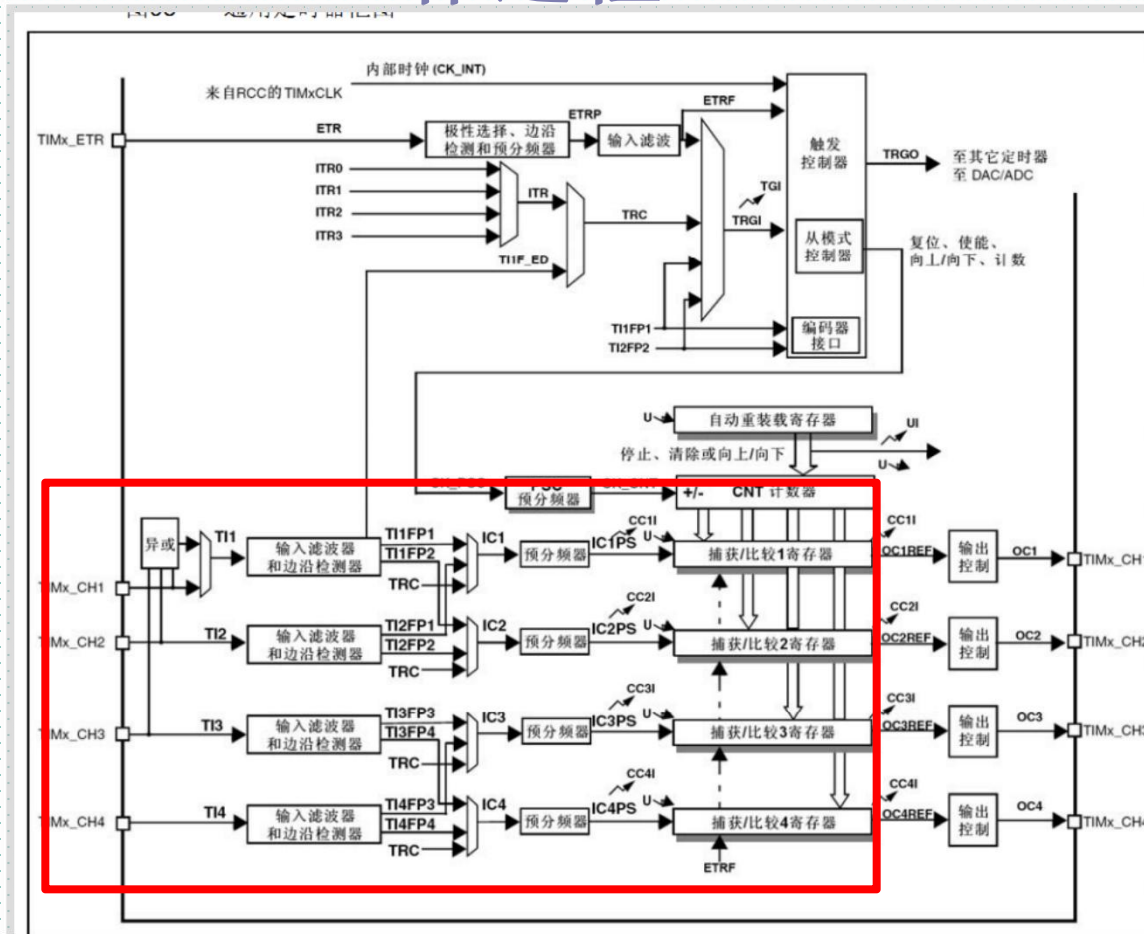
3

输入捕获实验讲解

✓ 通用定时器PWM概述



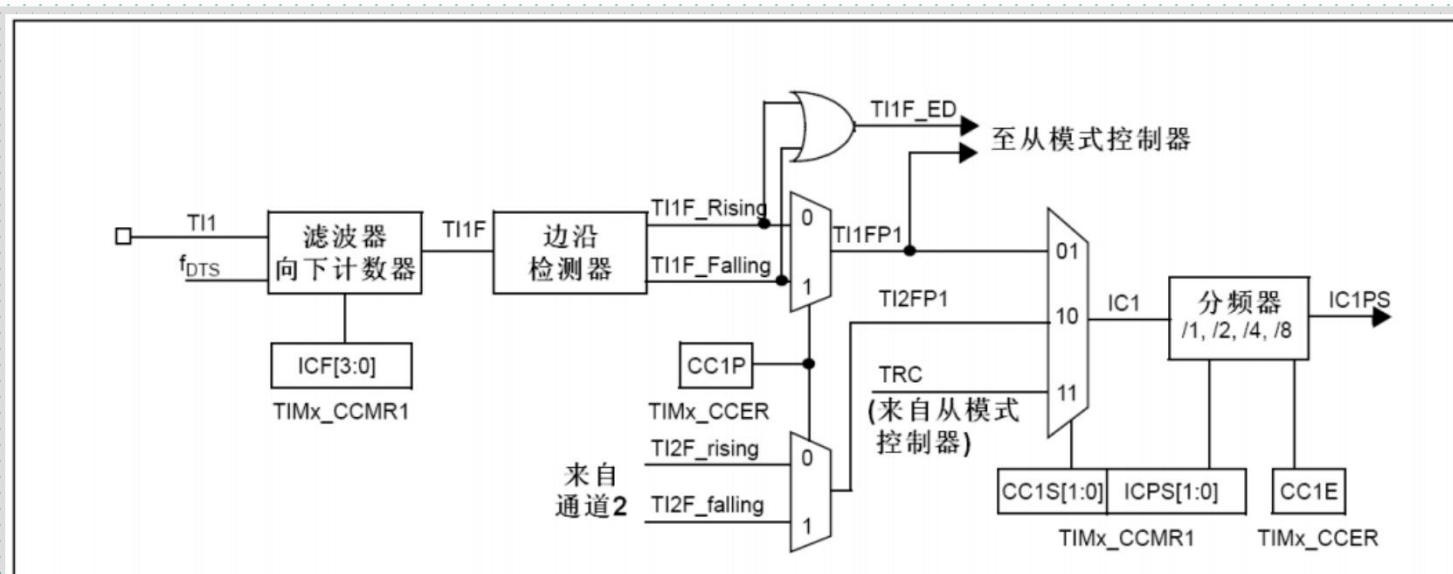
◆ STM32 PWM工作过程



✓ 通用定时器PWM概述



◆ STM32 输入捕获工作过程（通道1为例）

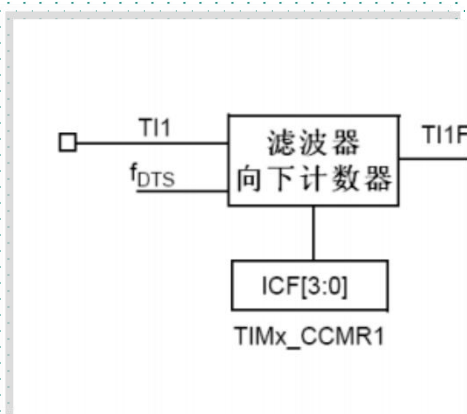


一句话总结工作过程：通过检测**TIMx_CHx**上的边沿信号，在边沿信号发生跳变（比如上升沿/下降沿）的时候，将当前定时器的值(**TIMx_CNT**)存放到对应的捕获/比较寄存器（**TIMx_CCRx**）里面，完成一次捕获。

✓ 通用定时器PWM概述



◆ 步骤1：设置输入捕获滤波器（通道1为例）

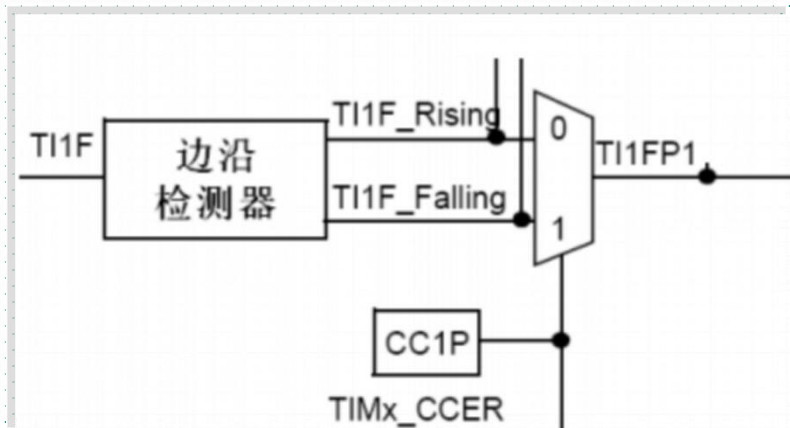


输入捕获 1 滤波器 ICF[3:0], 这个用来设置输入采样频率和数字滤波器长度。其中, f_{CK_INT} 是定时器的输入频率 (TIMxCLK), 一般为 72Mhz, 而 f_{DTS} 则是根据 TIMx_CR1 的 CKD[1:0] 的设置来确定的, 如果 CKD[1:0] 设置为 00, 那么 $f_{DTS} = f_{CK_INT}$ 。N 值就是滤波长度, 举个简单的例子: 假设 ICF[3:0]=0011, 并设置 IC1 映射到通道 1 上, 且为上升沿触发, 那么在捕获到上升沿的时候, 再以 f_{CK_INT} 的频率, 连续采样到 8 次通道 1 的电平, 如果都是高电平, 则说明却是一个有效的触发, 就会触发输入捕获中断 (如果开启的话)。这样可以滤除那些高电平脉宽低于 8 个采样周期的脉冲信号, 从而达到滤波的效果。这里, 我们不做滤波处理, 所以设置 ICF[3:0]=0000, 只要采集到上升沿, 就触发捕获。

✓ 通用定时器PWM概述



◆ 步骤2：设置输入捕获极性（通道1为例）



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。

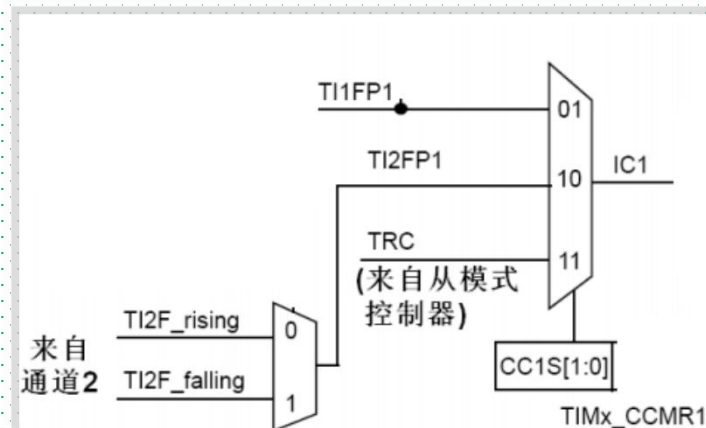
0: 不反相: 捕获发生在IC1的上升沿; 当用作外部触发器时, IC1不反相。

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。

✓ 通用定时器PWM概述



◆ 步骤三：设置输入捕获映射通道（通道1为例）



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;

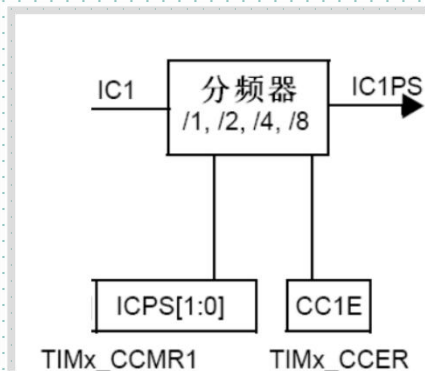
11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。

注: CC1S仅在通道关闭时(TIMx_CCER寄存器的CC1E='0')才是可写的。

✓ 通用定时器PWM概述



◆ 步骤四：设置输入捕获分频器（通道1为例）



位3:2 **IC1PSC[1:0]**: 输入/捕获1预分频器 (Input capture 1 prescaler)
这2位定义了CC1输入(IC1)的预分频系数。
一旦CC1E='0'(TIMx_CCER寄存器中), 则预分频器复位。
00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;
01: 每2个事件触发一次捕获;
10: 每4个事件触发一次捕获;
11: 每8个事件触发一次捕获。

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。
1: 开启— OC1信号输出到对应的输出引脚。
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;
1: 捕获使能。

✓ 通用定时器PWM概述



◆ 步骤五：捕获到有效信号可以开启中断

DMA/中断使能寄存器(TIMx_DIER)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TDE	保留	CC4DE	CC3DE	CC2DE	CC1DE	UDE	保留	TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rW		rW	rW	rW	rW	rW		rW		rW	rW	rW	rW	rW
位4		CC4IE : 允许捕获/比较4中断 (Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。													
位3		CC3IE : 允许捕获/比较3中断 (Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。													
位2		CC2IE : 允许捕获/比较2中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。													
位1		CC1IE : 允许捕获/比较1中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。													
位0		UIE : 允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。													

✓ 通用定时器PWM概述



◆ 最后：看看定时器通道对应引脚TIM5为例

Pin number					Pin name (function after reset) ⁽¹⁾	Pin type	I / O structure	Notes	Alternate functions	Additional functions
LQFP64	LQFP100	LQFP144	UFBGA176	LQFP176						
14	23	34	N3	40	PA0-WKUP (PA0)	I/O	FT	(5)	USART2_CTS / UART4_TX / ETH_MII_CRS / TIM2_CH1_ETR / TIM5_CH1 / TIM8_ETR / EVENTOUT	ADC123_IN0/WKUP ⁽⁴⁾
15	24	35	N2	41	PA1	I/O	FT	(4)	USART2_RTS / UART4_RX / ETH_RMII_REF_CLK / ETH_MII_RX_CLK / TIM5_CH2 / TIMM2_CH2 / EVENTOUT	ADC123_IN1
16	25	36	P2	42	PA2	I/O	FT	(4)	USART2_TX / TIM5_CH3 / TIM9_CH1 / TIM2_CH3 / ETH_MDIO / EVENTOUT	ADC123_IN2
-	-	-	F4	43	PH2	I/O	FT		ETH_MII_CRS / EVENTOUT	
-	-	-	G4	44	PH3	I/O	FT		ETH_MII_COL / EVENTOUT	

✓ 输入捕获关键库函数



◆ 输入捕获通道初始化函数:

void TIM_ICInit(TIM_TypeDef TIMx, TIM_ICInitTypeDef* TIM_ICInitStruct);*

```
typedef struct
{
    uint16_t TIM_Channel; //捕获通道1-4
    uint16_t TIM_ICPolarity; //捕获极性
    uint16_t TIM_ICSelection; //映射关系
    uint16_t TIM_ICPrescaler; //分频系数
    uint16_t TIM_ICFilter; //滤波器
} TIM_ICInitTypeDef;
```

```
TIM5_ICInitStruct.TIM_Channel = TIM_Channel_1;
TIM5_ICInitStruct.TIM_ICPolarity = TIM_ICPolarity_Rising;
TIM5_ICInitStruct.TIM_ICSelection = TIM_ICSelection_DirectTI;
TIM5_ICInitStruct.TIM_ICPrescaler = TIM_ICPSC_DIV1;
TIM5_ICInitStruct.TIM_ICFilter = 0x00;
TIM_ICInit(TIM5, &TIM5_ICInitStruct);
```


✓ 通用定时器PWM概述



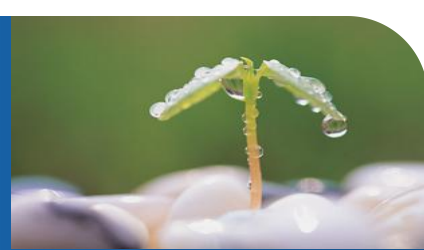
◆ 通道极性设置独立函数:

```
void TIM_OCxPolarityConfig(TIM_TypeDef* TIMx, uint16_t TIM_OCpolarity);
```

◆ 获取通道捕获值

```
uint32_t TIM_GetCapture1(TIM_TypeDef* TIMx);
```


✓ 通用定时器PWM概述



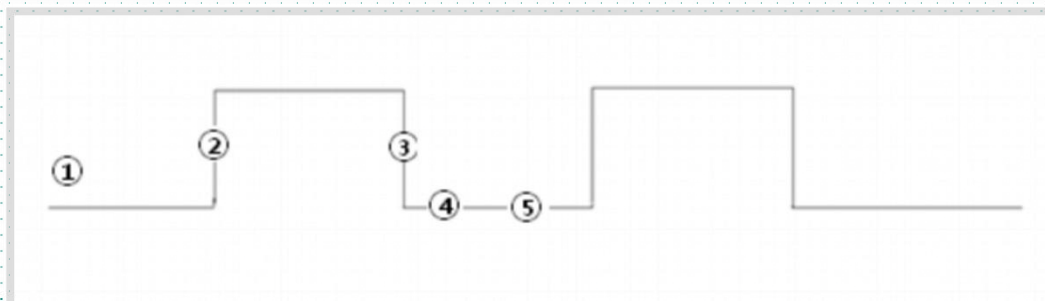
■ 输入捕获的一般配置步骤:

- ① 初始化定时器和通道对应IO的时钟。
- ② 初始化IO口，模式为复用：**GPIO_Init();**
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
- ③ 设置引脚复用映射：
GPIO_PinAFConfig();
- ④ 初始化定时器ARR，PSC
TIM_TimeBaseInit();
- ⑤ 初始化输入捕获通道
TIM_ICInit();
- ⑥ 如果要开启捕获中断，
TIM_ITConfig();
NVIC_Init();
- ⑦ 使能定时器：**TIM_Cmd();**
- ⑧ 编写中断服务函数：**TIMx_IRQHandler();**

✓ 通用定时器PWM概述

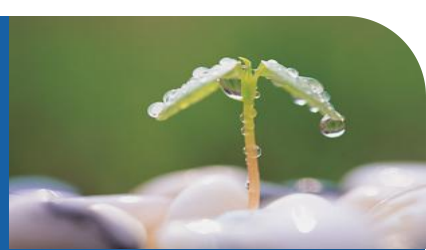


实验目的：测量信号的脉冲宽度



TIM5CH1_CAPTURE_STA		
bit7	bit6	bit5~0
捕获完成标志	捕获到高电平标志	捕获高电平后定时器溢出的次数

✓ 通用定时器PWM概述



TIM5_Cap_Init 函数执行的时候就设置好了，然后等待上升沿中断到来，当捕获到上升沿中断，此时如果 TIM5CH1_CAPTURE_STA 的第 6 位为 0，则表示还没有捕获到新的上升沿，就先把 TIM5CH1_CAPTURE_STA、TIM5CH1_CAPTURE_VAL 和 TIM5->CNT 等清零，然后再设置 TIM5CH1_CAPTURE_STA 的第 6 位为 1，标记捕获到高电平，最后设置为下降沿捕获，等待下降沿到来。如果等待下降沿到来期间，定时器发生了溢出，就在 TIM5CH1_CAPTURE_STA 里面对溢出次数进行计数，当最大溢出次数来到的时候，就强制标记捕获完成（虽然此时还没有捕获到下降沿）。当下降沿到来的时候，先设置 TIM5CH1_CAPTURE_STA 的第 7 位为 1，标记成功捕获一次高电平，然后读取此时的定时器值到 TIM5CH1_CAPTURE_VAL 里面，最后设置为上升沿捕获，回到初始状态。

这样，我们就完成一次高电平捕获了，只要 TIM5CH1_CAPTURE_STA 的第 7 位一直为 1，那么就不会进行第二次捕获，我们在 main 函数处理完捕获数据后，将 TIM5CH1_CAPTURE_STA 置零，就可以开启第二次捕获。

✓ 3. 输入捕获实验讲解



输入捕获实验

GO!!

感谢您对“正点原子”团队的支持



硬件平台：正点原子STM32开发板
版权所有：广州市星翼电子科技有限公司
淘宝店铺：<http://eboard.taobao.com>
技术论坛：www.openedv.com



淘宝店铺：<http://eboard.taobao.com>

技术论坛：www.openedv.com