

# 追米集，DMA搬运至内存...

原创 耿云鹏 于 2021-01-12 13:47:11 发布 1158 收藏 8

版权

文章标签： 的注入通道和规则通道的区别

## 引言

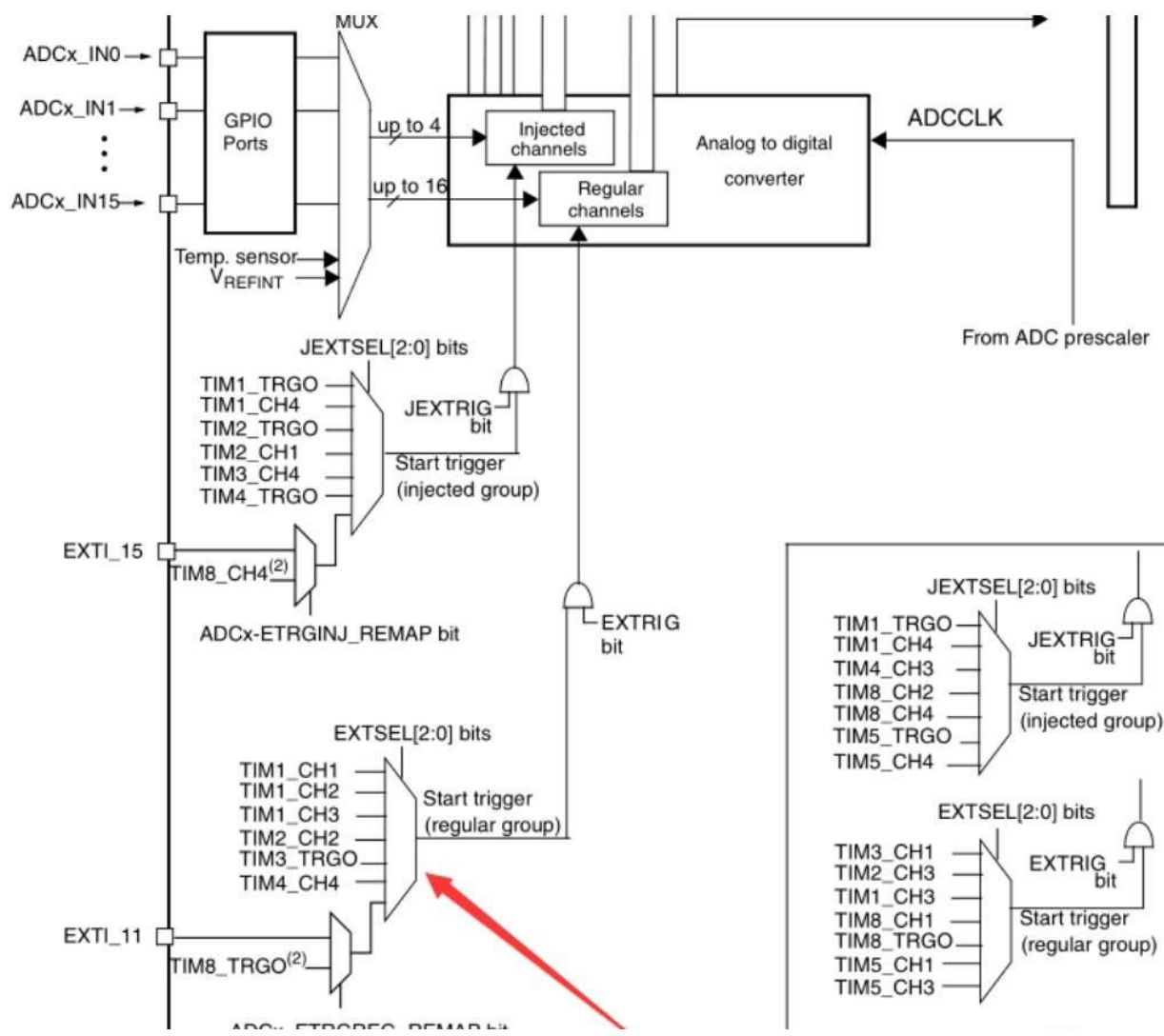
ADC<sup>Q</sup> 的功能是将模拟信号采样得到数字信号，而有些时候，我们需要使用到定时采样，比如在计算一个采集的波形的频率的时候，我们需要精确的知道采样频率，也就是 1 s 内采集的点数，这个时候，就需要使用到定时采样。定时采样有如下三种方法：

- 使用定时器 中断<sup>Q</sup>，每隔一段时间进行 ADC 转换，但是这样每次都必须读 ADC 的数据寄存器，非常浪费时间。
- 把 ADC 设置成连续转换模式，同时对应的 DMA<sup>Q</sup> 通道开启循环模式，这样 ADC 就一直在进行数据采集然后通过 DMA 把数据搬运至内存。这样进行处理的话，需要加一个定时中断，用来读取内存中的数据。
- 使用 ADC 的定时器触发 ADC 转换的功能，然后使用 DMA 进行数据的搬运。这样就只要设置好定时器的触发间隔，就能实现 ADC 定时采样转换的功能，然后使能 DMA 转换完成中断，这样每次转换完就会产生中断。

本文，笔者将采用第三种方法进行 AD 采集，使用 TIM 定时器触发 AD 采集，然后 DMA 搬运至 内存<sup>Q</sup>。

## ADC 简介

首先来看一下 ADC 的框图：



## ADC 框图

在本文中，我们使用的是规则通道进行转换，这里要指出的一点是规则通道和注入通道两者的区别，以下是关于两种通道的说明：

- 规则通道：我们平时使用的就是这个通道，就是规规矩矩的按照我们设定的转换顺序就行转换的通道。
- 注入通道：注入通道可以理解为是插入，也就是插队的意思，它是一种不安分的通道。它是一种在规则通道转换的时候强行插入要进行转换的一种，它的存在就像是程序中的中断一样，换个角度说，也就是注入通道只有在规则通道存在的情况下才会存在。

说了规则通道和注入通道的区别之后，我们来看我们在本文中所用到的规则通道的触发方式。我们最为常用的一种就是软件触发，即配置到 ADC 之后，就会自动地进行转换，然后去读 ADC 的数据寄存器就可以得到 ad 转换得到的数值。还有一种方法就是外部触发，而外部触发又包括定时器触发和外部 IO 触发，在本文中，我们使用的是定时器触发，通过上述的 ADC 功能框图，我们可以知道 ADC 的定时器触发又有如下几种类

- TIM1\_CH1 : 定时器 1 的通道 1 的 PWM 触发
- TIM1\_CH2 : 定时器 2 的通道 2 的 PWM 触发
- TIM1\_CH3: 定时器 1 的通道 3 的 PWM 触发
- TIM2\_CH2 : 定时器 2 的通道 2 的 PWM 触发
- TIM3\_TRGO: 定时器 3 触发，TRGO 属于内部触发，不需要配置对应的输出 IO 脚。相当于是 TIM3 的定时器内部计数一样，只是到了一定时间就触发 ADC 转换，而这个触发的实现，不依赖 IO 口的配置。
- TIM4\_CH4 : 定时器 4 的通道 4 的 PWM 触发

## 定时器配置

在进行了上述简单的介绍之后，我们来具体到代码的细节来看，本文采用的是 TIM4\_CH4 进行外部触发 ADC 采样。首先来看 TIM 的配置，代码如下：

```
void ADC1_External_T4_CC4_Init(void){    GPIO_InitTypeDef GPIO_InitStructure;    RCC_APB2PeriphCl
```

在这里需要注意的是和 sample\_psc 是个变量，而这个变量可以通过调用库函数 TIM\_SetIC4Prescaler(TIM\_TypeDef\* TIMx, uint16\_t TIM\_ICPSC) 重新配置 TIM 所产生的 pwm 的频率，详细的原理不在此处进行赘述了，既然都能够改变 TIM 产生的 PWM 的原理，那么也就能动态地改变 ADC 的采样频率，也就是决定 ADC 在 1 s 中能够采样多少个点，具体的原理在后续指出。还有一个需要注意的地方是 TIM\_Cmd(TIM4, DISABLE)，这里配置的是禁止 TIM 定时器使能，因为还有 ADC 和 DMA 还没有进行配置，因此，我们需要在 ADC 和 DMA 都配置好之后，再将 TIM4 进行使能。

## DMA 配置

因为笔者所涉及到的 ADC 的具体应用是这样的，也就是通过定时器触发 ADC 采集，然后采集一定数量的点数之后，在这里笔者每个 ADC 的通道是采集了 256 个点，然后对这 256 个点进行处理，处理完毕之后，再以一定时间间隔再采集 256 个点，周而复始地进行采集和处理。并且，这里需要的是同时采集 2 个通道的数



据，每个通道采集 256 个点，也就是说，我们一次性处理的是  $256 * 2 = 512$  个点的数据，采集完成之后，再通过 DMA 将数据搬运至内存，因此，也就有了如下所示的 DMA 配置：

```
static void ADC1_DMA1_Init(void){    DMA_InitTypeDef DMA_InitStructure;    NVIC_InitTypeDef NVIC_
```

代码比较直观，都是一些相关的配置，这里所要指出的一点是在第五行配置了中断服务函数 `DMA1_Channel1_IRQHandler`，具体的思路就是当采集的点数满足设定的点数时，就进入中断服务函数进行处理，在这里需要注意的是我们是从 ADC 外设将数据搬运至内存，所以 DMA 外设的地址是 ADC1 数据寄存器的地址，可以使用宏定义的方式定义如下：

```
#define ADC1_DR_Address    ((uint32_t)0x4001244C)
```

也可以直接取地址的方式设置，设置方式如下所示：

```
DMA_InitStructure.DMA_PeripheralBaseAddr = ( u32 ) ( & ( ADC_x->DR ) );
```

设置好外设的地址之后，我们就需要设置内存的地址，在这里，因为我们要采集两个通道的数据，并且每个通道要采集 256 个点的数据，所以在这里定义了一个如下所示的二维数组：

```
uint16_t ADC_ConvertedValue[ADC_BUFF_LEN][2] = {0};
```

上述中的 `ADC_BUFF_LEN` 就是一个通道要采集的点数，也就是 256 个，2 所代表的就是有两个通道。在这里需要稍微思考一下是二维数组的定义方式，为什么定义成的是 256 行 2 列的二维数组，而不是 2 行 256 列的二维数组，我们来看一下 256 行 2 列的数组的布局如下：

	0	1
0		
1		
2		
3		
	•	•
	•	•
	•	•
255		

二维数组内存分布

根据二维数组的大小也解释了 DMA 的 `Buffer_size` 是 `ADC_BUFF_LEN * 2`，同时，由于在下面设置了内存地址是递增的，而又有两个通道，那么他的转换顺序是这样的，也就是先转换通道 1 的值存入数组，然后再转换通道 2 的数据存入数组，然后，以一定时间间隔地转换 512 次，然后发生 DMA 中断，这样也就能够说明数组为什么是定义成 256 行 2 列了。

## ADC 配置

在配置了定时器和 DMA 之后，我们接下来来进行 ADC 的配置，上文中，我们配置的是使用 TIM4 的 4 通道产生 PWM 来触发 ADC 进行采集，然后设置了 DMA 来进行数据的搬运，因此，ADC 相关的配置如下所示：

```
void ADC_init(void){    GPIO_InitTypeDef GPIO_InitStructure;    RCC_APB2PeriphClockCmd(RCC_APB2Pe
```

配置过程比较简单，没有什么逻辑性可言，不在这里进行赘述，这里需要指出的一点是因为我们设置的是 2 个通道的采集，所以，在这里应该使能 ADC 的扫描模式，另一方面，我们采用的是 TIM 产生 pwm 触发 adc 进行采集，所以要禁止 ADC 的连续转换模式，这就是两个需要注意的地方。

## DMA 中断服务函数

在前文我们说了，我们通过 pwm 触发 ADC 采集，当采集了规定的点数之后，就会产生 DMA 中断，然后在 DMA 中断里面去处理数据，但是由于中断服务函数的要求是执行时间尽可能短，所以，我们可以在中断服务函数里置位数据采集完成标志位的方式来使得主程序进行数据处理，程序代码如下所示：

```
void ADC1_DMA1_IT_Hander(void){    if (DMA_GetFlagStatus(DMA1_FLAG_TC1))    {        DMA_Clea
```

上述代码中，被注释掉的部分是释放信号量，这个是使用 RTOS 是用来同步线程的一个操作，其功能与裸机的标志位是相同的。

## 总结

上述便是本次分享的内容，其实现的一个功能便是使用 PWM 触发 ADC 多通道采集，并使用 DMA 进行搬运，通过这样子就可以精确地控制 ADC 的采样频率，也就是控制 1 s 钟可以采集多少个点。最后，而这个采样频率就是 pwm 的频率，但是为了更加精确的计算其真实的采样频率还应该加上 ADC 通道的转换一个数据的转换时间，这样才是最为精确的采样频率。在下一篇文章中，笔者将继续介绍基于这篇文章的应用，也就是根据采样得到的点，计算波形的频谱，计算波形的频率。

**相关资源：** [...ADC的规则通道和注入通道有什么区别\\_stm32注入通道和规则通道...](#)