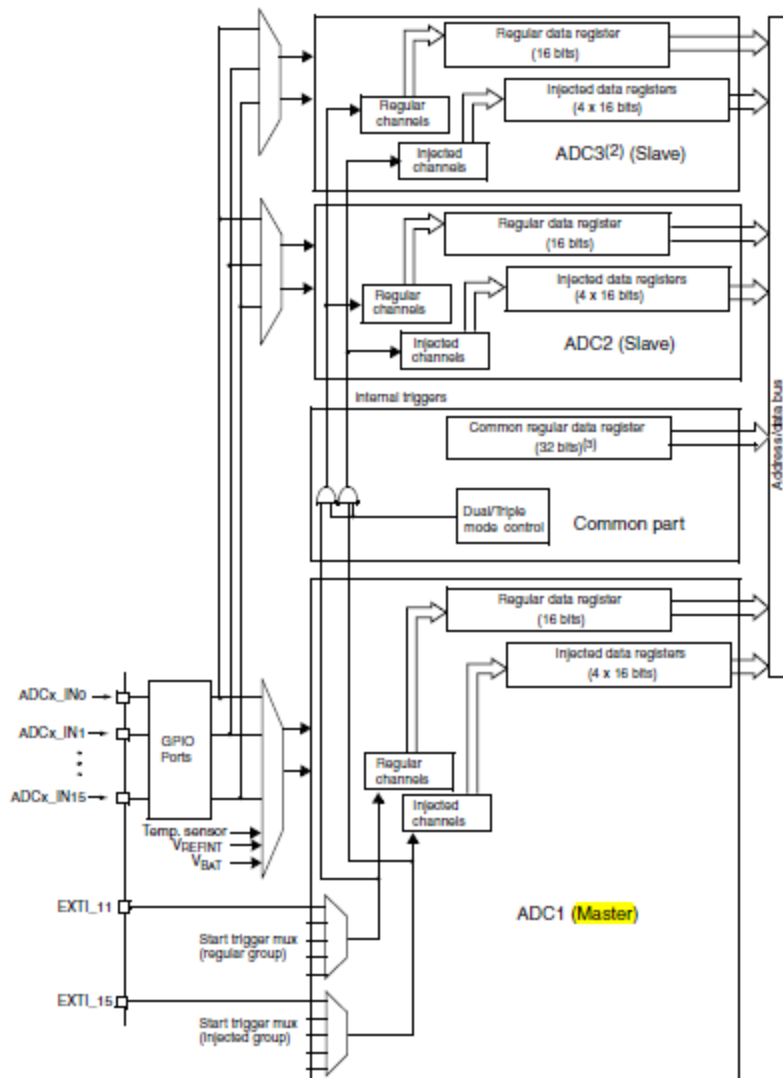


STM32 多个 ADC 模块同时采样转换的应用示例

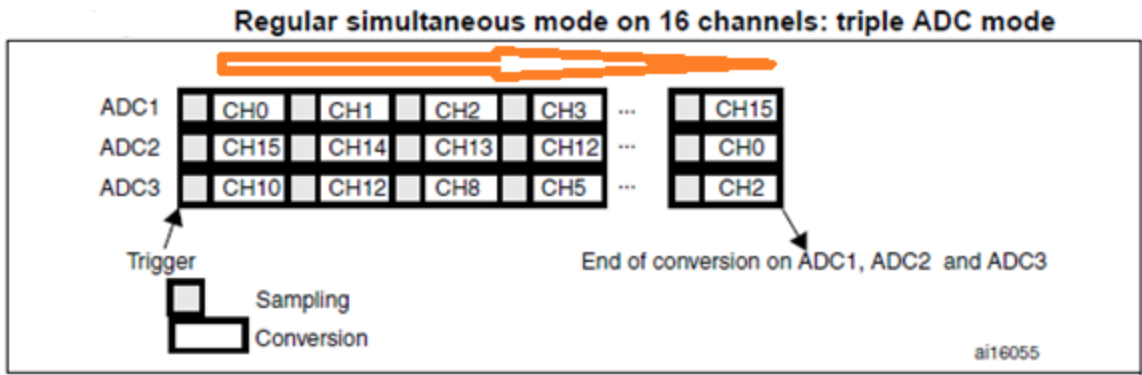
在 STM32 家族里，多数系列芯片内含 2 到 3 个 ADC 模块，有的甚至更多，比方 G4 系列可以有 5 个 ADC 模块。其中，通道数因不同的系列或型号多少不等，几个到几十个的都有。有时，我们可能需要多个 ADC 模块同时工作，比方 3 个 ADC 模块同时采样转换。这时如果芯片内含有 3 个 ADC 模块，并支持同时采样转换就很方便。STM32 家族里的 STM32F4 系列、STM32F7 等其它系列都含有 3 个 ADC 模块，并支持同时 AD 采样转换。

这里就 3 个 ADC 模块同时进行采样转换应用，做个简单实现演示，以供有需要的用户参考。

此时 3 个 ADC 模块会建立主从关系。以 STM32F4 芯片为例。内部大致框架如下：



采样转换时按如下图示操作，每次对属于 3 个 AD 模块的 3 个通道进行 AD 转换。如果多组的话，依次扫描进行。

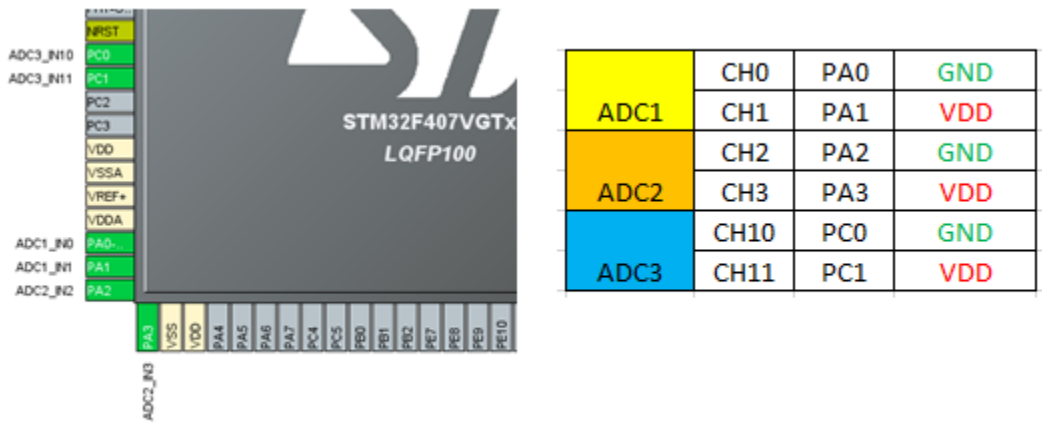


各通道转换结束时产生 DMA 请求，DMA 按照 ADC1、ADC2、ADC3 的顺序依次将数据取走，然后放到指定的内存空间。

好，大致原理就介绍这么多。更多细节还是请阅读 STM32 参考手册的 ADC 相关章节。

这里在 ADC1/ADC2/ADC3 三个模块各选择 2 各通道，它们的通道号及相关输入连接如下：

【注：ADC 模块的参考电压也选用 VDD. 下面实验基于 STM32F407 DISCOVERY 板来进行。】



另外，我这里使用 STM32 的 TIM3 的更新事件触发 ADC 转换。

- 一、使用 STM32CubeMx 图形化配置工具完成基本配置
 - 1.1 RCC/SYS 等必需配置项目【略】
 - 1.2 对 ADC 进行配置。
 - 1.2.1 对 ADC1 及相关 DMA 进行配置。

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F) < >

ADC1

ADCs_Common_Settings

- Mode Triple regular simultaneous mode only
- DMA Access Mode DMA access mode 1
- Delay between 2 sampling phases 5 Cycles

ADC_Settings

- Clock Prescaler PCLK2 divided by 8
- Resolution 12 bits (15 ADC Clock cycles)
- Data Alignment Right alignment
- Scan Conversion Mode Enabled
- Continuous Conversion Mode Disabled
- Discontinuous Conversion Mode Disabled
- DMA Continuous Requests Enabled
- End Of Conversion Selection EOC flag at the end of all conversions

ADC_Regular_ConversionMode

- Number Of Conversion 2
- External Trigger Conversion Source Timer 3 Trigger Out event
- External Trigger Conversion Edge Trigger detection on the rising edge

Rank

- Channel 1 Channel 0
- Sampling Time 56 Cycles

Rank

- Channel 2 Channel 1
- Sampling Time 56 Cycles

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

DMA Request	Stream	Direction	Priority
ADC1	DMA2 Stream 0	Peripheral To Memory	Low

Add Delete

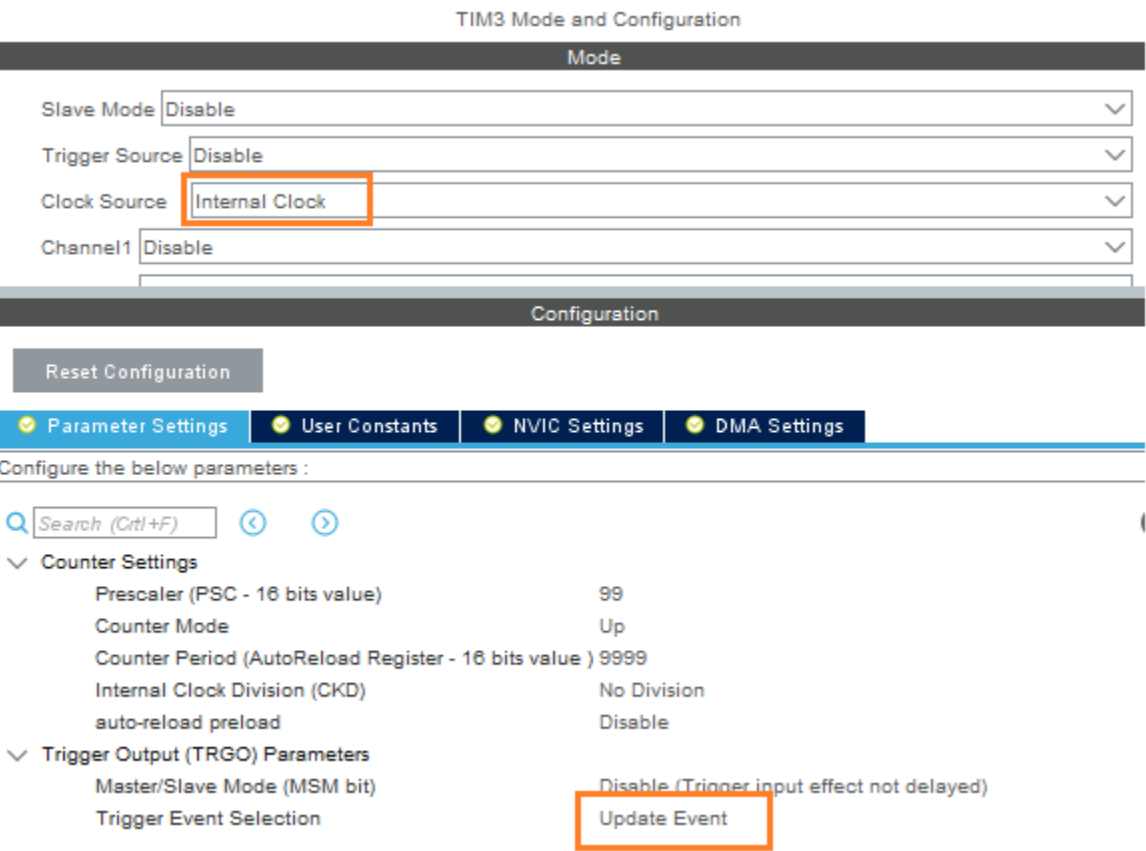
DMA Request Settings

		Peripheral	Memory
Mode	Circular	Increment Address ☐	☒
Use Fifo ☐	Threshold	Data Width	Word

至于对 ADC2 和 ADC3 及相关 DMA 进行配置,主要参数和配置流程跟 ADC1 一样,它们做从 ADC,少了个触发选择项。注意选择对应的 ADC 通道及采样时间。考虑到版面,这里就不重复贴图了。

1.3 对 TIM3 进行配置。

对 TIM3 的配置比较简单，安排你需要的时基参数，选择合适的触发输出即可。ADC 转换靠它定期触发。



上面配置中，DMA 传输中断默认使能了，至于其它，你根据需要选择使能。

2、基于上面的 CubeMx 配置生成 C 代码工程。

3、添加用户代码。

3.1 为 DMA 传输准备一个内存数组，用来存放转换结果。

```
__IO uint32_t ADC_Result[6];
```

3.2 添加启动 ADC3/ADC2/ADC1 以及 TIM3 的相关代码。

```
if(HAL_ADC_Start(&hadc3) != HAL_OK)
{
    /* Start Error */
    Error_Handler();
}

if(HAL_ADC_Start(&hadc2) != HAL_OK)
{
    /* Start Error */
    Error_Handler();
}
if(HAL_ADCEx_MultiModeStart_DMA(&hadc1, (uint32_t*)ADC_Result, 6) != HAL_OK)
{
    /* Start Error */
    Error_Handler();
}

HAL_TIM_Base_Start(&htim3);
```

4、编译、运行、验证。

结果如下，结果是正确的。到此整个演示过程完毕。

				Name	Value	Type
				ADC_Result	0x20000010 ADC_Result	unsigned int[6]
ADC1	CH0	PA0	GND	[0]	0x00000001	unsigned int
	CH1	PA1	VDD	[1]	0x00000001	unsigned int
ADC2	CH2	PA2	GND	[2]	0x00000001	unsigned int
	CH3	PA3	VDD	[3]	0x00000FFF	unsigned int
ADC3	CH10	PC0	GND	[4]	0x00000FFF	unsigned int
	CH11	PC1	VDD	[5]	0x00000FFF	unsigned int

看到这里，是不是觉得很简单呢。只要你愿意对 STM32 手册做认真阅读把把握相关原理，你也可以轻松实现。不过，或许有人对结果产生了疑问，DMA 搬到数组的数据怎么是前面连续 3 个 00，后面连续 3 个 0xfff 而不是 00, fff, 00, fff, 00, fff 呢？不妨结合上面的介绍和参考手册自行思考下，相信你可以找到答案。