数字与模拟转换2

程晨闻

东南大学电气工程学院

➢ 内容回顾

- 数字/模拟转换器DAC 的工作原理
  - 译码特点
  - 倒T型网络DAC
  - 电阻串结构DAC
  - DAC指标（转换精度，分辨率，线性度，偏移，转换时间…）
  - MCP4725
- 模拟/数字转换器ADC的工作原理
  - 采样，保持，量化，编码
  - 计数器式，逐次逼近式，双积分式，并行式，Delta-Sigma…
  - ADC指标（分辨率，转换速度，精度，量化误差，偏移误差，满刻度误差，线性度…）

内容概要

- **内容概要**
  - **STM32F401的ADC模块及其使用方法**

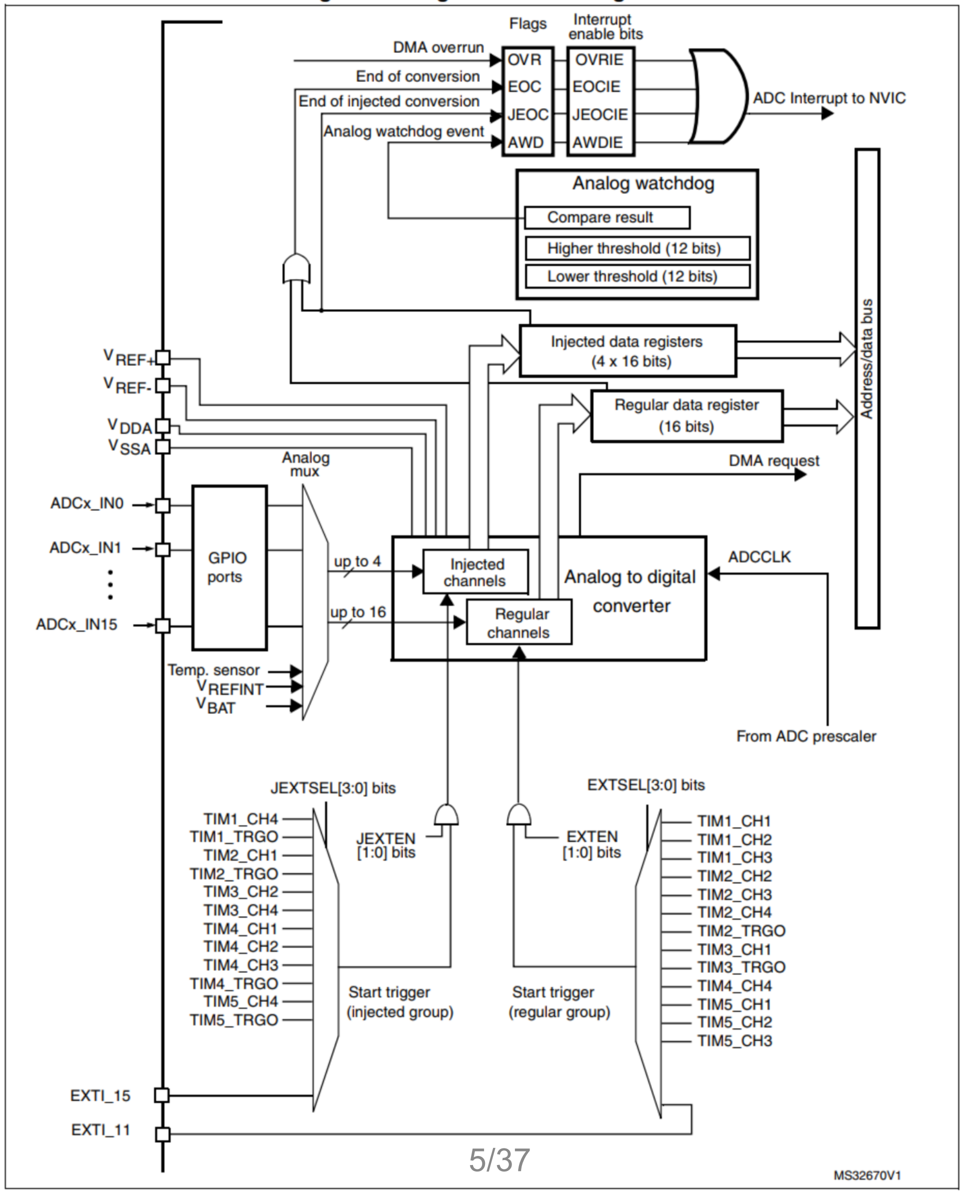
STM32F401的ADC

STM32F401的ADC模块

- 有1个逐次逼近型ADC，ADC1
- 12-bit, 10-bit, 8-bit, 6-bit精度配置
- 单次或连续转换模式
- 在转换结束、注入转换结束以及发生模拟看门狗或溢出事件时产生中断
- 用于从通道0到通道“n”自动转换的扫描模式
- 数据对齐（16位储存），以保持内置数据一致性
- 不连续模式
- ADC供电：204V至3.6V全速运行，1.8V低速运行
- ADC输入电压范围：VREF- ≤ VIN ≤ VREF+
- 16个外部源，2个内部源，VBAT通道

➢ ADC模块的基本结构

Figure 31. Single ADC block diagram

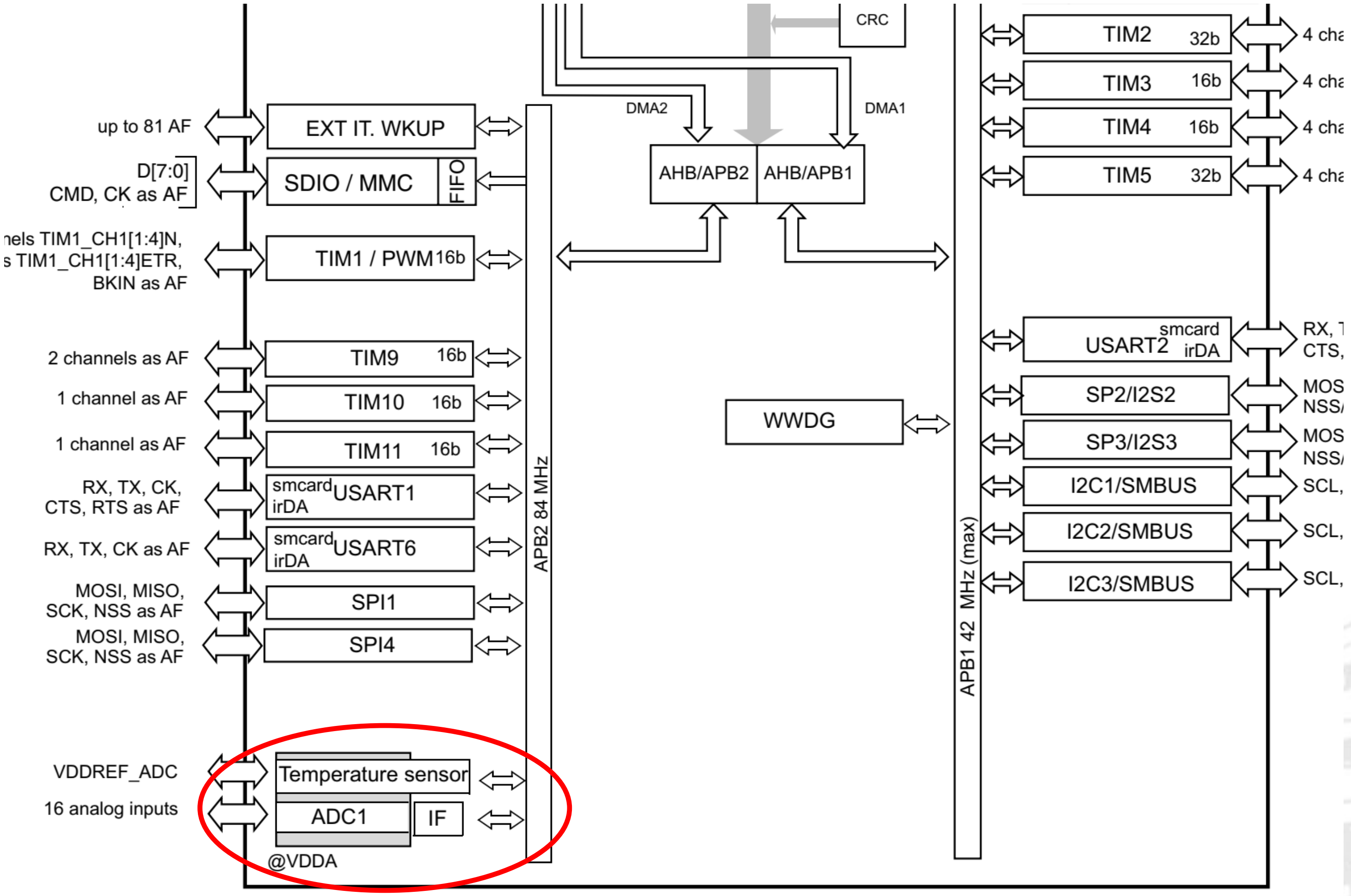


➢ ADC模块的基本结构

- ➢ 1. ADC模块的核心是一个模拟数字转换器（analog to digital converter）
- ➢ 2. MCU的16个引脚，对应ADC的16个输入通道（channel）。每一时刻，可以由模拟复选器（Analog mux）选择1个引脚（输入通道）上的模拟信号输入到模拟数字转换器中进行转换。
- ➢ 3. 转换的速度由ADCCLK控制。
- ➢ 4. 转换的参考电压是VREF- 和 VREF+。在STM32F401CB这个芯片中，VREF-在内部连接到了VSSA，而VREF+在内部连接到了VDDA。
- ➢ 5. 可以由软件启动转换，也可以由其他定时器模块的信号启动转换。
- ➢ 6. 转换的模式为成组转换。
- ➢ 7. ADC模块中，有两个组，一个是常规组（Regular Group），还有一个是插入组（Injected Group）。
- ➢ 8. 两个寄存器ADC_CR1和ADC_CR2来控制ADC模块。有一个ADC_SR寄存器来指示ADC的状态。

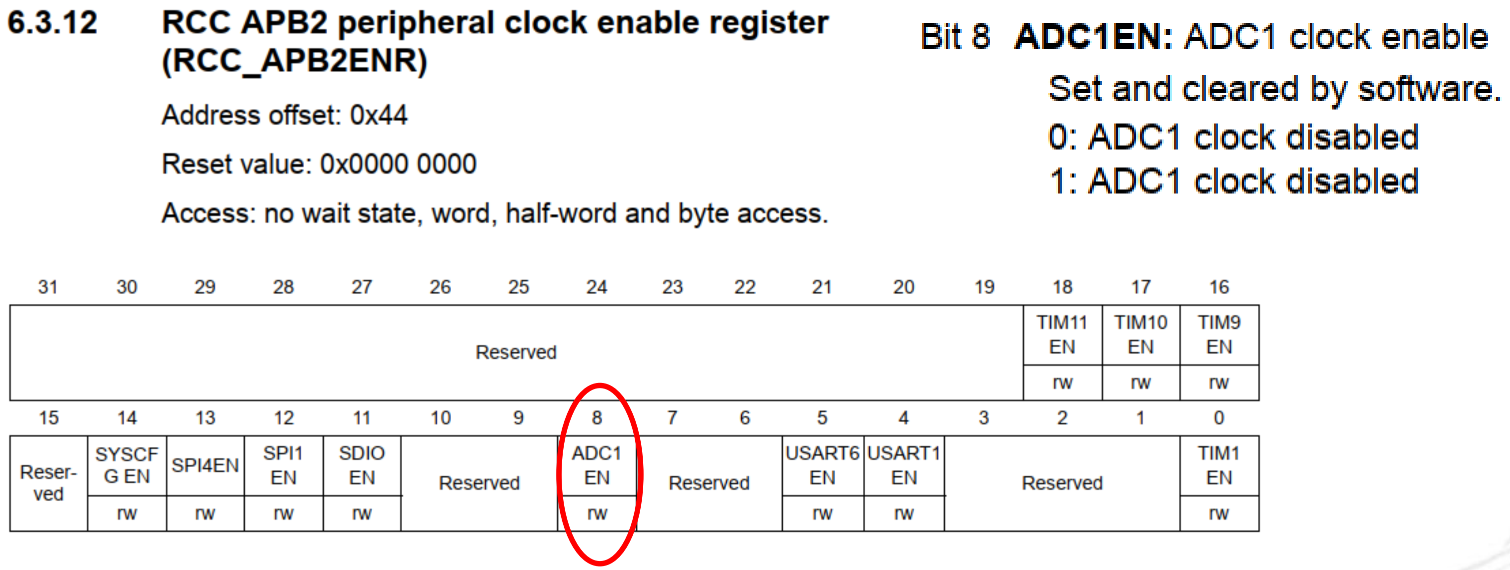
➢ ADC时钟使能

- 看手册的MCU结构框图，确认ADC1模块挂载在APB2总线上



➢ ADC时钟使能

- 将RCC_APB2ENR寄存器的ADC1EN位置1，使能ADC时钟



6.3.12 RCC APB2 peripheral clock enable register (RCC_APB2ENR)

Address offset: 0x44

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

Bit 8 **ADC1EN:** ADC1 clock enable

Set and cleared by software.

0: ADC1 clock disabled

1: ADC1 clock disabled

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM11 EN	TIM10 EN	TIM9 EN
													rw	rw	rw

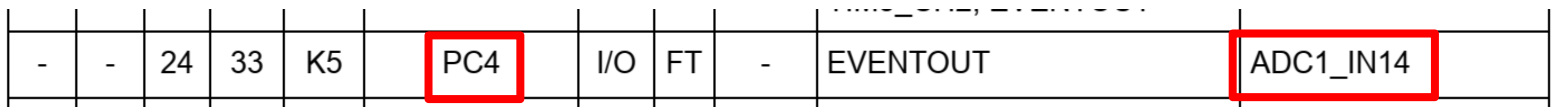
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser-ved	SYSCF G EN	SPI4EN	SPI1 EN	SDIO EN	Reserved		ADC1 EN	Reserved		USART6 EN	USART1 EN	Reserved			TIM1 EN
	rw	rw	rw	rw			rw			rw	rw				rw

```
#define __HAL_RCC_ADC1_CLK_ENABLE() \
do { \
__IO uint32_t tmpreg = 0x00U; \
SET_BIT(RCC->APB2ENR, RCC_APB2ENR_ADC1EN);\
/* Delay after an RCC peripheral clock enabling */ \
tmpreg = READ_BIT(RCC->APB2ENR, RCC_APB2ENR_ADC1EN);\
UNUSED(tmpreg); \
} while(0U)
```

➢ ADC所用GPIO模块时钟使能

– 查看引脚与ADC输入信号的对应关系

使用ADC1模块的第14个通道，即ADC1_IN14。查看芯片手册：



-	-	24	33	K5	PC4	I/O	FT	-	EVENTOUT	ADC1_IN14

```
__HAL_RCC_GPIOC_CLK_ENABLE();
```

```
temp = GPIOC->MODER;
temp &= ~(0x03 << (4 * 2));
temp |= (0x03 << (4 * 2));
GPIOC->MODER = temp;
```

STM32F401的ADC模块的使用

➢ ADC模块使能

11.12.3 ADC control register 2 (ADC_CR2)

Address offset: 0x08

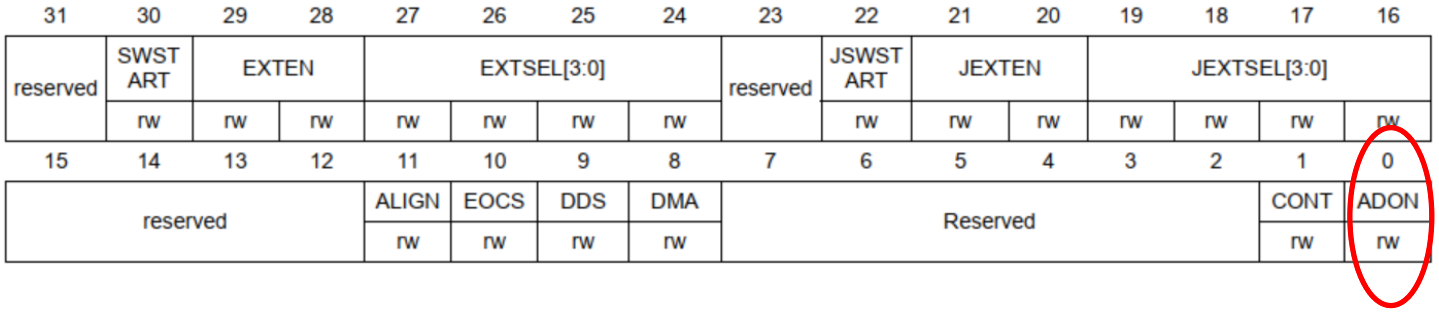
Reset value: 0x0000 0000

ADC on-off control

The ADC is powered on by setting the ADON bit in the ADC_CR2 register. When the ADON bit is set for the first time, it wakes up the ADC from the Power-down mode.

Conversion starts when either the SWSTART or the JSWSTART bit is set.

You can stop conversion and put the ADC in power down mode by clearing the ADON bit. In this mode the ADC consumes almost no power (only a few μA).



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved	SWSTART	EXTEN		EXTSEL[3:0]				reserved	JSWSTART	JEXTEN		JEXTSEL[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
				rw	rw	rw	rw							rw	rw

Bit 0 **ADON:** A/D Converter ON / OFF

This bit is set and cleared by software.

Note: 0: Disable ADC conversion and go to power down mode

1: Enable ADC

```
ADC1->CR2|=0x01; //ADON=1
```

➢ ADC时钟

在使用ADC模块之前，需要确认ADC模块最大的工作频率

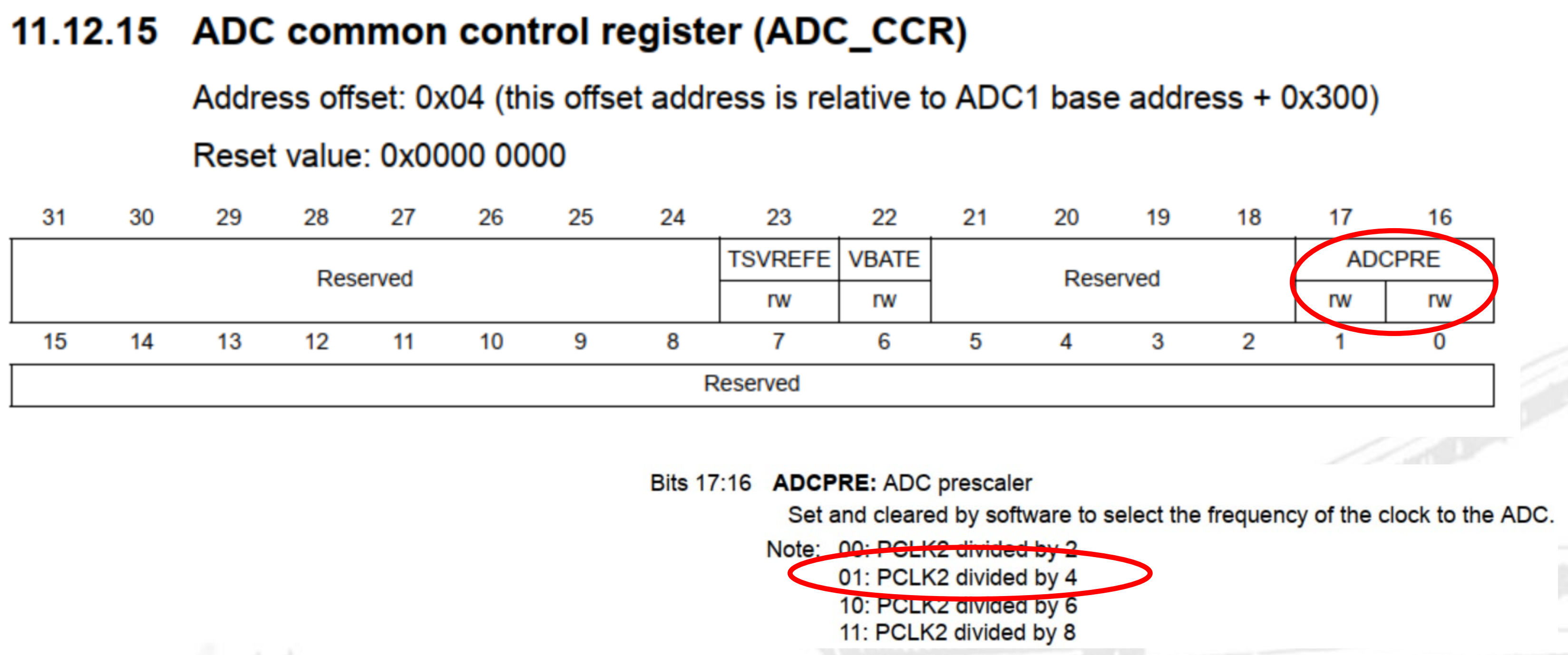
Table 66. ADC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDA}	Power supply	$V_{DDA} - V_{REF+} < 1.2$ V	1.7[1]	-	3.6	V
V_{REF+}	Positive reference voltage		1.7[1]	-	V_{DDA}	
V_{REF-}	Negative reference voltage	-	-	0	-	
f_{ADC}	ADC clock frequency	V_{DDA} = 1.7[1] to 2.4 V	0.6	15	18	MHz
		V_{DDA} = 2.4 to 3.6 V	0.6	30	36	MHz
f_{TRIG}[2]	External trigger frequency	f_{ADC} = 30 MHz, 12-bit resolution	-	-	1764	kHz
		-	-	-	17	$1/f_{ADC}$

ADC模块的工作频率fADC与VDDA的电压有关系，如果VDDA为3.3V，那么fADC的最高频率为36Mhz。

➢ ADC时钟

ADC_CCR寄存器中的第16、17位控制了APB2总线到fADC的分频系数



11.12.15 ADC common control register (ADC_CCR)

Address offset: 0x04 (this offset address is relative to ADC1 base address + 0x300)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TSVREFE	VBATE	Reserved				ADCPRE	
								rw	rw					rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits 17:16 **ADCPRE:** ADC prescaler

Set and cleared by software to select the frequency of the clock to the ADC.

Note: 00: PCLK2 divided by 2

01: PCLK2 divided by 4

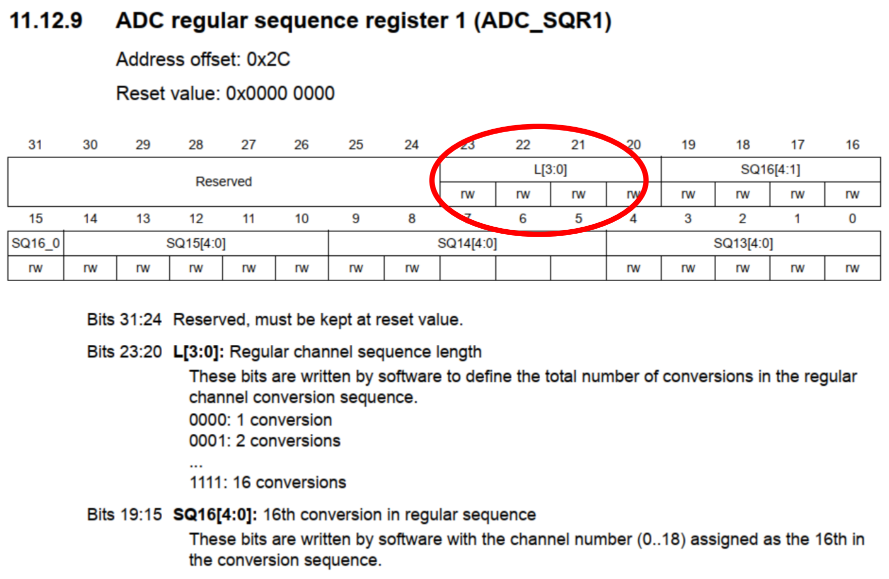
10: PCLK2 divided by 6

11: PCLK2 divided by 8

```
ADC1_COMMON->CCR=(0x01<<16); //ADCPRE=0x01 /4
```

➢ ADC通道选择

- AD转换（conversion）可以分为两个组：常规组（regular）和注入组（injected）
- 组包含了AD转换使用的通道及其顺序
  - 常规组：最多16个转换，在ADC_SQRx寄存器中定义通道和顺序；在ADC_SQR1寄存器中的L[3:0]位定义转换的总数



11.12.9 ADC regular sequence register 1 (ADC_SQR1)

Address offset: 0x2C

Reset value: 0x0000 0000

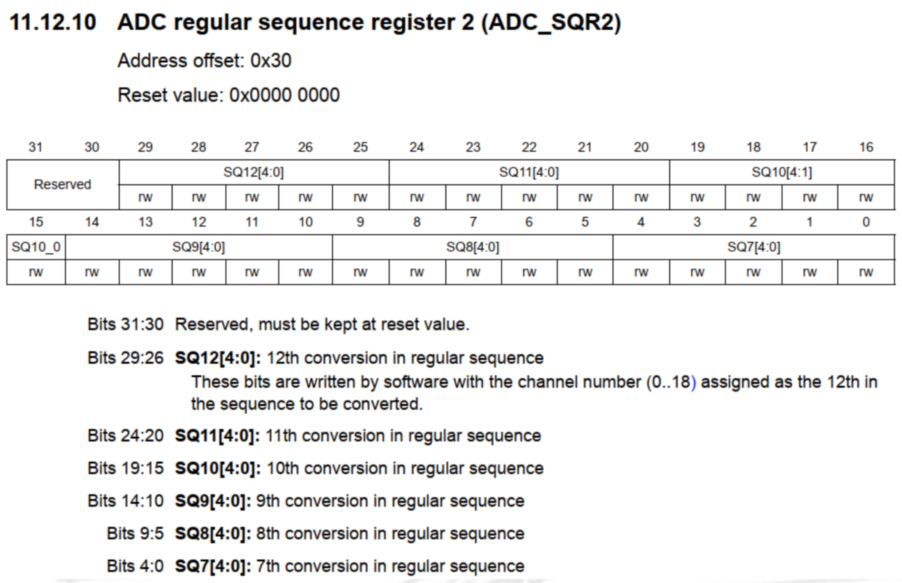
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								L[3:0]				SQ16[4:1]			
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16_0	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits 31:24 Reserved, must be kept at reset value.

Bits 23:20 **L[3:0]:** Regular channel sequence length
These bits are written by software to define the total number of conversions in the regular channel conversion sequence.
0000: 1 conversion
0001: 2 conversions
...
1111: 16 conversions

Bits 19:15 **SQ16[4:0]:** 16th conversion in regular sequence
These bits are written by software with the channel number (0..18) assigned as the 16th in the conversion sequence.



11.12.10 ADC regular sequence register 2 (ADC_SQR2)

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10_0	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 Reserved, must be kept at reset value.

Bits 29:26 **SQ12[4:0]:** 12th conversion in regular sequence
These bits are written by software with the channel number (0..18) assigned as the 12th in the sequence to be converted.

Bits 24:20 **SQ11[4:0]:** 11th conversion in regular sequence

Bits 19:15 **SQ10[4:0]:** 10th conversion in regular sequence

Bits 14:10 **SQ9[4:0]:** 9th conversion in regular sequence

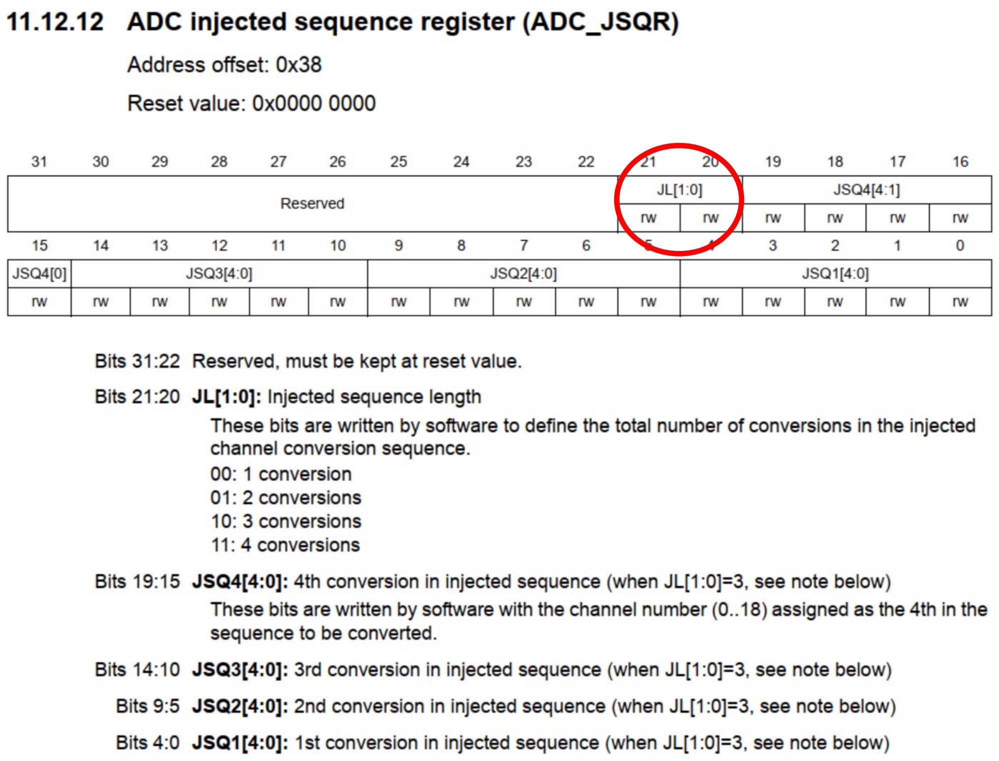
Bits 9:5 **SQ8[4:0]:** 8th conversion in regular sequence

Bits 4:0 **SQ7[4:0]:** 7th conversion in regular sequence

STM32F401的ADC模块的使用

➢ ADC通道选择

- AD转换（conversion）可以分为两个组：常规组（regular）和注入组（injected）
- 组包含了AD转换使用的**通道**及其**顺序**
  - 注入组：最多4个转换，在ADC_JSQR寄存器中定义通道和顺序；在ADC_JSQR寄存器中的L[1:0]位定义转换的总数



11.12.12 ADC injected sequence register (ADC_JSQR)

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										JL[1:0]		JSQ4[4:1]			
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]	JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:22 Reserved, must be kept at reset value.

Bits 21:20 **JL[1:0]:** Injected sequence length
These bits are written by software to define the total number of conversions in the injected channel conversion sequence.
00: 1 conversion
01: 2 conversions
10: 3 conversions
11: 4 conversions

Bits 19:15 **JSQ4[4:0]:** 4th conversion in injected sequence (when JL[1:0]=3, see note below)
These bits are written by software with the channel number (0..18) assigned as the 4th in the sequence to be converted.

Bits 14:10 **JSQ3[4:0]:** 3rd conversion in injected sequence (when JL[1:0]=3, see note below)

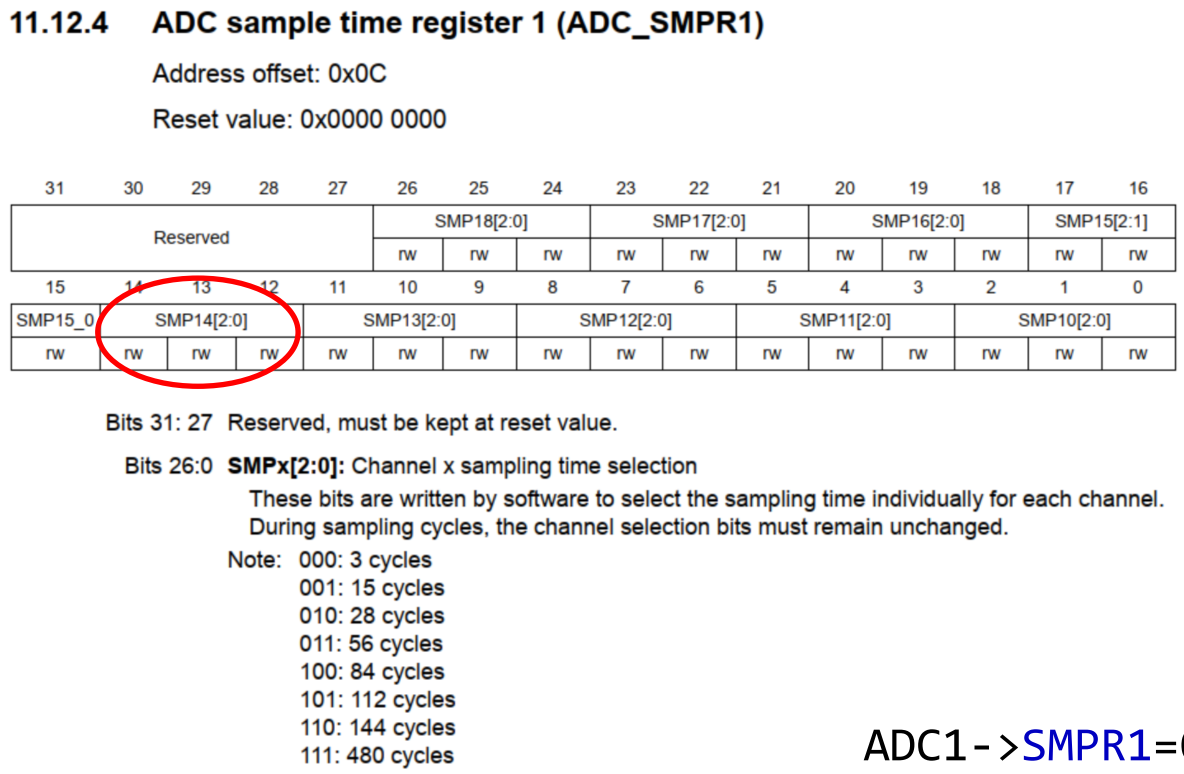
Bits 9:5 **JSQ2[4:0]:** 2nd conversion in injected sequence (when JL[1:0]=3, see note below)

Bits 4:0 **JSQ1[4:0]:** 1st conversion in injected sequence (when JL[1:0]=3, see note below)

```
ADC1->JSQR=((0<<20)+(14<<(3*5)));
```

➢ 采样时间

- ADC采样需要一系列的ADCCLK周期
- 采样周期数由ADC_SMPR1和ADC_SMPR2中的SMPx[2:0]位定义
- 每个通道可以配置为不同的采样时间
- 总的转换时间可由下式计算得到：　Tconv = Sampling time + 12 cycles



11.12.4　ADC sample time register 1 (ADC_SMPR1)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31: 27　Reserved, must be kept at reset value.

Bits 26:0　**SMPx[2:0]:** Channel x sampling time selection

These bits are written by software to select the sampling time individually for each channel.
During sampling cycles, the channel selection bits must remain unchanged.

Note:　000: 3 cycles
001: 15 cycles
010: 28 cycles
011: 56 cycles
100: 84 cycles
101: 112 cycles
110: 144 cycles
111: 480 cycles

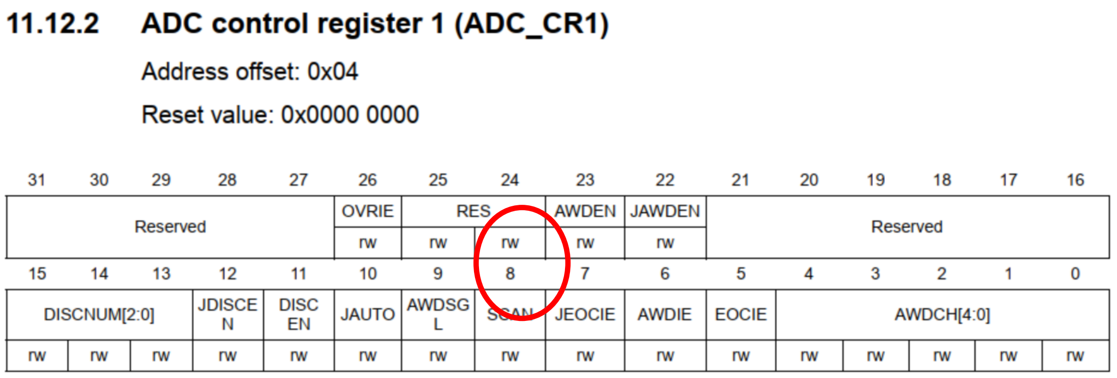
```
ADC1->SMPR1=0x01<<12;
```

➢ ADC转换模式

- **扫描模式**（Scan mode）： 用于扫描一组模拟通道（对**不同channel**采样）
  - 将 ADC_CR1寄存器中的SCAN位**置1**

```
ADC1->CR1|=1<<8;
```

  - ADC扫描ADC_SQRx寄存器（regular channel）或ADC_JSQR寄存器（injected channel）中的所有通道
  - 每个通道都进行一次转换，完成后下一个通道自动开始转换
  - 如果CONT位**置1**，最后一个通道转换完成后会从第一个通道重新开始采样
- 转换结束后，EOC位**置1**：
  - 如果EOCS (ADC_CR2) = 0，一组转换结束后EOC位置1
  - 如果EOCS（ADC_CR2) = 1，一个转换结束后EOC位置1



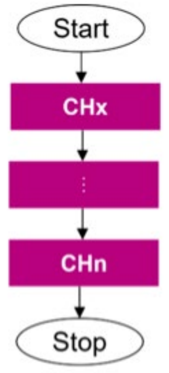
11.12.2 ADC control register 1 (ADC_CR1)

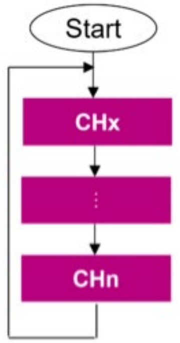
Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OVRIE	RES		AWDEN	JAWDEN	Reserved					
					rw	rw	rw	rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw





➢ 启动ADC转换

ADC_CR2寄存器：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved	SWSTART	EXTEN		EXTSEL[3:0]				reserved	JSWSTART	JEXTEN		JEXTSEL[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
				rw	rw	rw	rw							rw	rw

将ADC_CR2寄存器的第22位，即JSWSTART设置为1，就可以开始插入组的转换。

```
ADC1->CR2|=1<<22;
```

➢ 读取ADC转换数据

ADC_SR寄存器：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										OVR	STRT	JSTRT	JEOC	EOC	AWD
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ADC_SR寄存器的第2位JEOC，指示插入组是否转换完成。如果JEOC为1，表示插入组的转换已经完成。

因此在转换前要先清除JEOC：

```
ADC1->SR&=~(0x04);//JEOC=0;
```

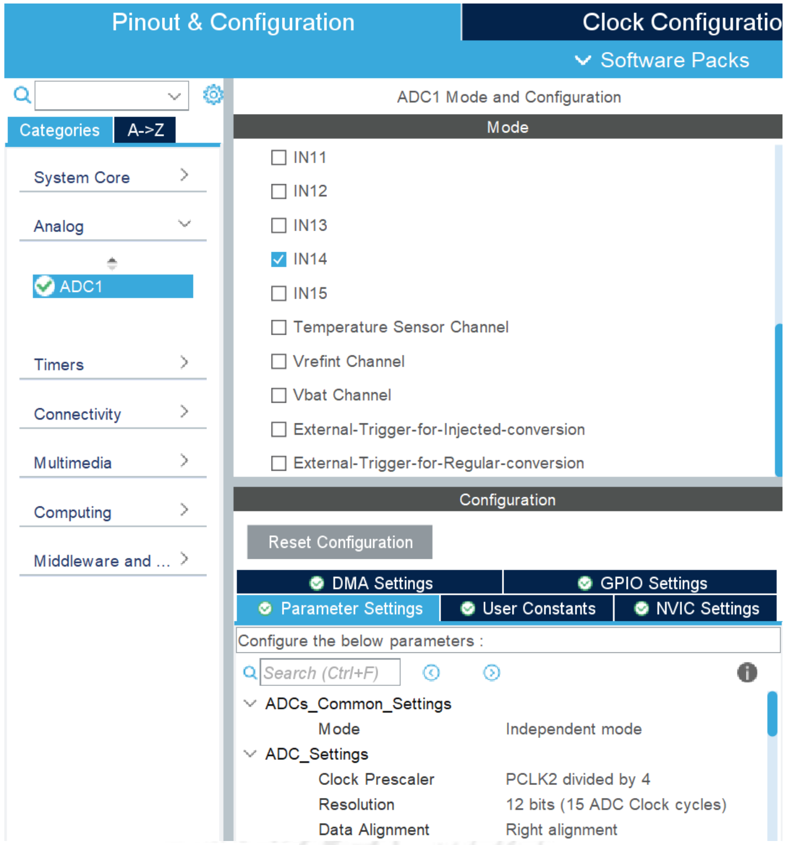
启动ADC转换后，再查看JEOC，如果JEOC为1，就从插入组数据寄存器中读取转换值

```
while((ADC1->SR & 0x04)==0);

ADCdata14=ADC1->JDR1;
```

➢ STM32CubeIDE图形化初始化ADC

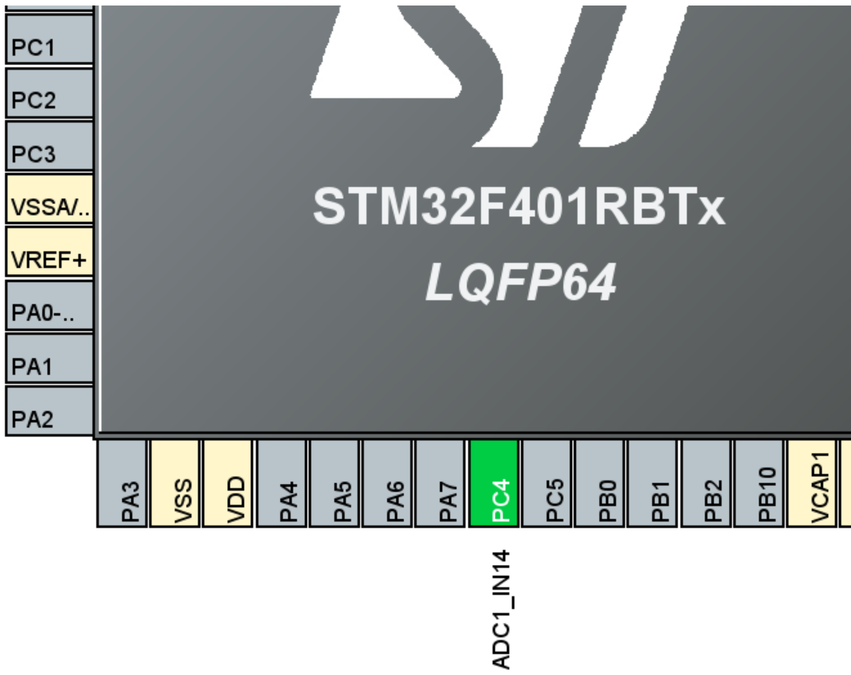
– 从HelloWorld项目复制一个新项目，命名为ADC，并完成相应的修改，打开配置文件ADC.ioc
– 选择ADC1，勾选IN14，可以发现右侧芯片引脚配置图中的PC4被配置为AD采样管脚



☐ IN11
☐ IN12
☐ IN13
☑ IN14
☐ IN15
☐ Temperature Sensor Channel
☐ Vrefint Channel
☐ Vbat Channel
☐ External-Trigger-for-Injected-conversion
☐ External-Trigger-for-Regular-conversion

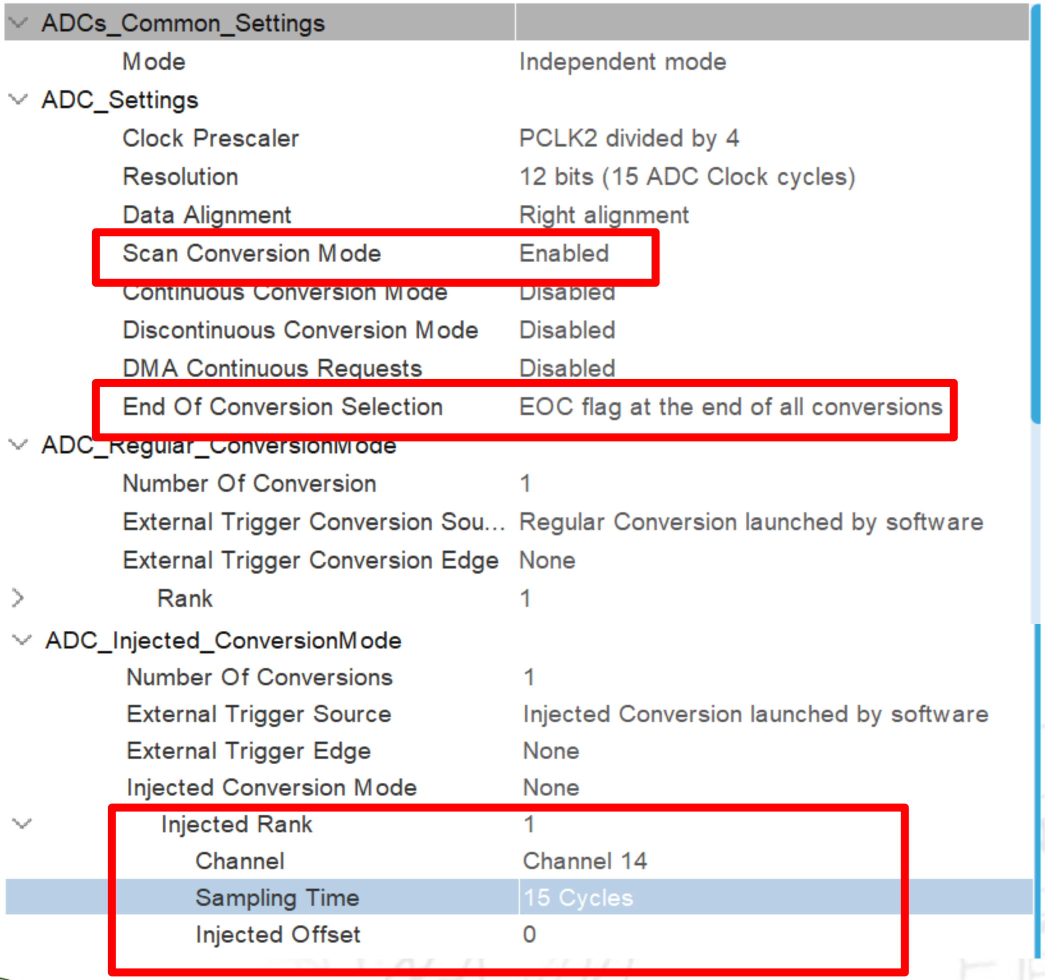
Configure the below parameters :

ADCs_Common_Settings	
Mode	Independent mode
ADC_Settings	
Clock Prescaler	PCLK2 divided by 4
Resolution	12 bits (15 ADC Clock cycles)
Data Alignment	Right alignment



➢ STM32CubeIDE图形化初始化ADC

– 配置ADC模块
– 保存并生成代码



ADCs_Common_Settings	
Mode	Independent mode
ADC_Settings	
Clock Prescaler	PCLK2 divided by 4
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	Disabled
End Of Conversion Selection	EOC flag at the end of all conversions
ADC_Regular_ConversionMode	
Number Of Conversion	1
External Trigger Conversion Sou...	Regular Conversion launched by software
External Trigger Conversion Edge	None
Rank	1
ADC_Injected_ConversionMode	
Number Of Conversions	1
External Trigger Source	Injected Conversion launched by software
External Trigger Edge	None
Injected Conversion Mode	None
Injected Rank	1
Channel	Channel 14
Sampling Time	15 Cycles
Injected Offset	0

➢ 用HAL库函数操作ADC

- 修改main()函数，在while循环中加入以下语句

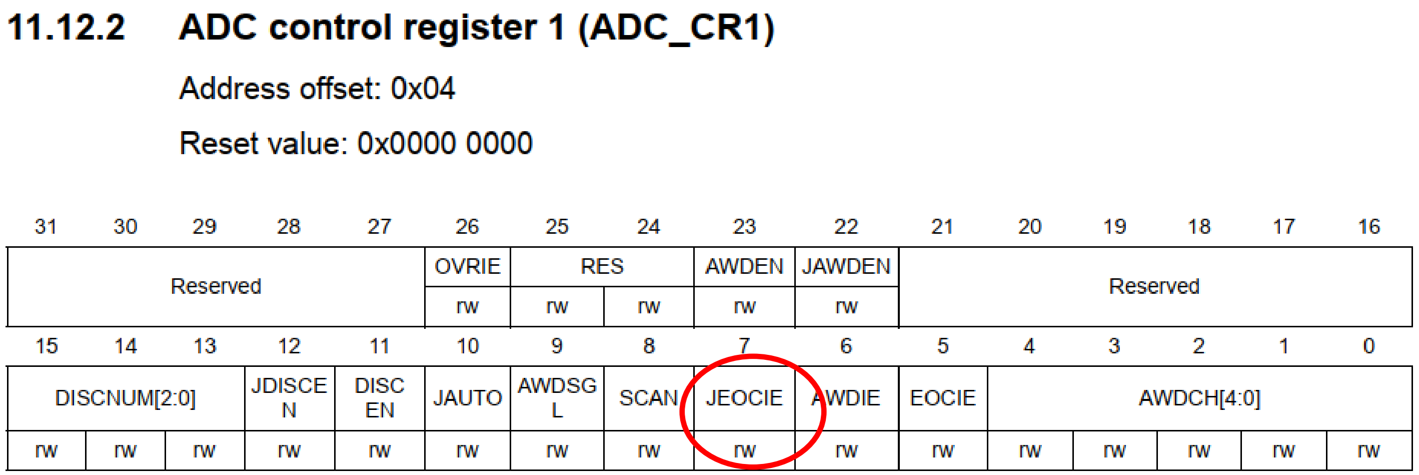
```
HAL_ADCEx_InjectedStart(&hadc1);

if(HAL_ADCEx_InjectedPollForConversion(&hadc1,2)==HAL_OK){
        adcdata14=HAL_ADCEx_InjectedGetValue(&hadc1,1);
}
```

- 转动滑动变阻器，观察变量adcdata14的值

➢ ADC中断

– 如果将第7位JEOCIE设置为1，则插入组转换完成后，ADC模块会发送中断请求给NVIC



11.12.2 ADC control register 1 (ADC_CR1)

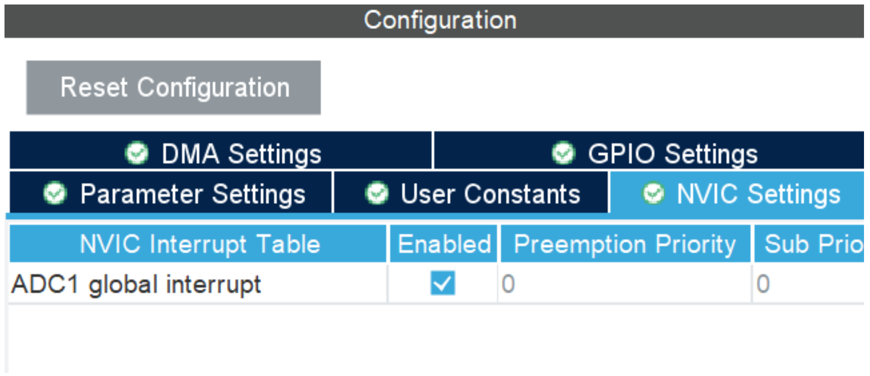
Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OVRIE	RES		AWDEN	JAWDEN	Reserved					
					rw	rw	rw	rw	rw						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

➢ ADC中断

上面工程的基础上， 在ADC1模块的配置界面中， 点击NVIC Settings选项卡， 勾选ADC1 global interrupt。



NVIC Interrupt Table	Enabled	Preemption Priority	Sub Prio
ADC1 global interrupt	✓	0	0

保存并生成代码。

➢ ADC中断

生成代码后，在stm32f4xx_it.c文件中，会出现ADC1的中断服务函数，用于处理ADC1的中断事务。

```
void ADC_IRQHandler(void)
{
  /* USER CODE BEGIN ADC_IRQn 0 */

  /* USER CODE END ADC_IRQn 0 */
  HAL_ADC_IRQHandler(&hadc1);
  /* USER CODE BEGIN ADC_IRQn 1 */

  /* USER CODE END ADC_IRQn 1 */
}
```

➢ ADC中断

➢ 调用HAL_ADCEx_InjectedStart_IT(&hadc1);函数可以启动ADC1的插入组转换

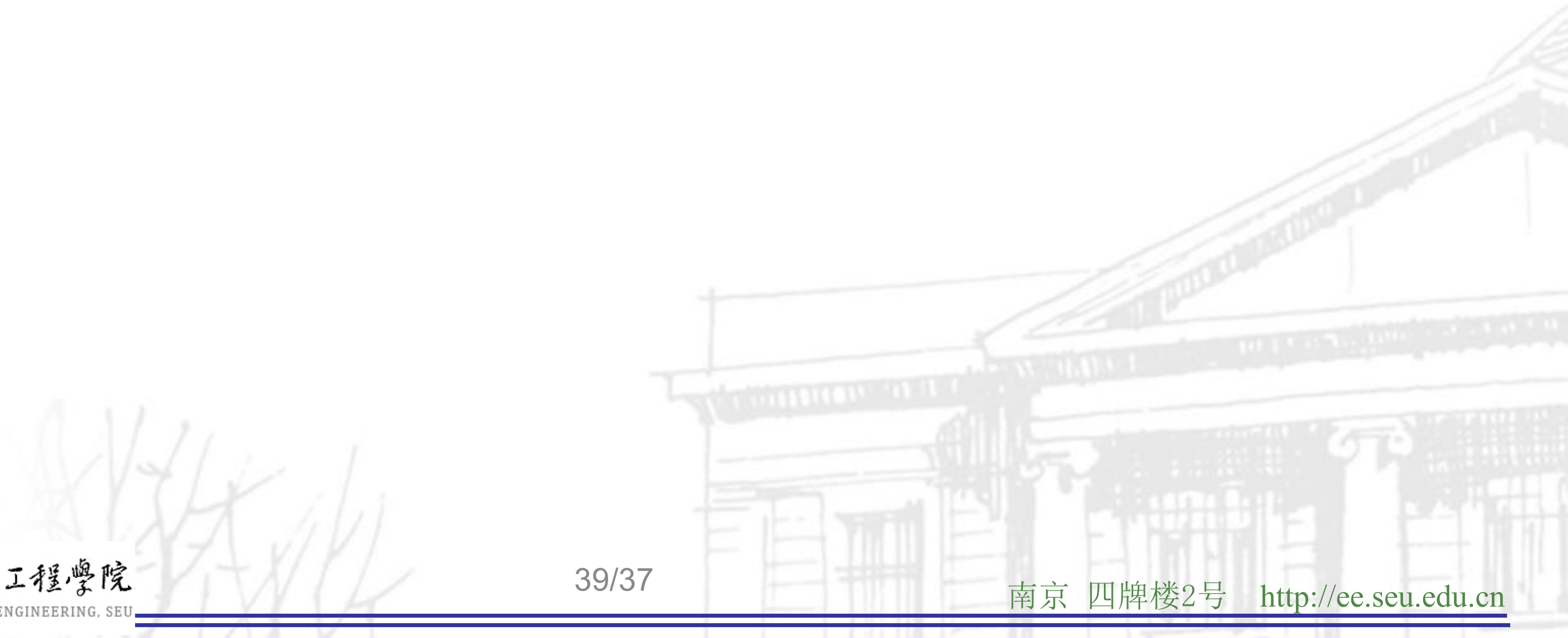
➢ 在转换完成后产生中断，进入中断服务函数

➢ 在中断服务函数中，会调用弱函数HAL_ADCEx_InjectedConvCpltCallback用于处理转换数据。

➢ 因此在main.c中重新定义HAL_ADCEx_InjectedConvCpltCallback函数

```
/* USER CODE BEGIN 0 */
void HAL_ADCEx_InjectedConvCpltCallback(ADC_HandleTypeDef* hadc){
        adcdata14=HAL_ADCEx_InjectedGetValue(hadc,1);

}
/* USER CODE END 0 */
```



谢谢！