Embedded System and Microcomputer Principle

LAB8 Analog-to-digital Converter (ADC)

2021 Fall

wangq9@mail.sustech.edu.cn

CONTENTS

1 ADC Description

2 ADC Registers

3 How to Program

4 Practice

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

01

ADC Description

1. ADC Description

-- What is ADC

- ADC（Analog-to-digital converter）即模拟/数字转换器
- 用于将模拟形式的连续信号转换为数字形式的离散信号的一类设备。
- 在计算机控制系统中，需经各种检测装置，以连续变化的电压或电流作为模拟量，随时提供被控制对象的有关参数（如速度、压力、温度等）而进行控制。计算机的输入必须是**数字量**，故需用模数转换器达到控制目的。
- 典型的模拟/数字转换器将模拟信号转换为表示一定比例电压值的数字信号。
- 与之相对的设备称为DAC，即数字/模拟转换器。

1. ADC Description

-- Main features

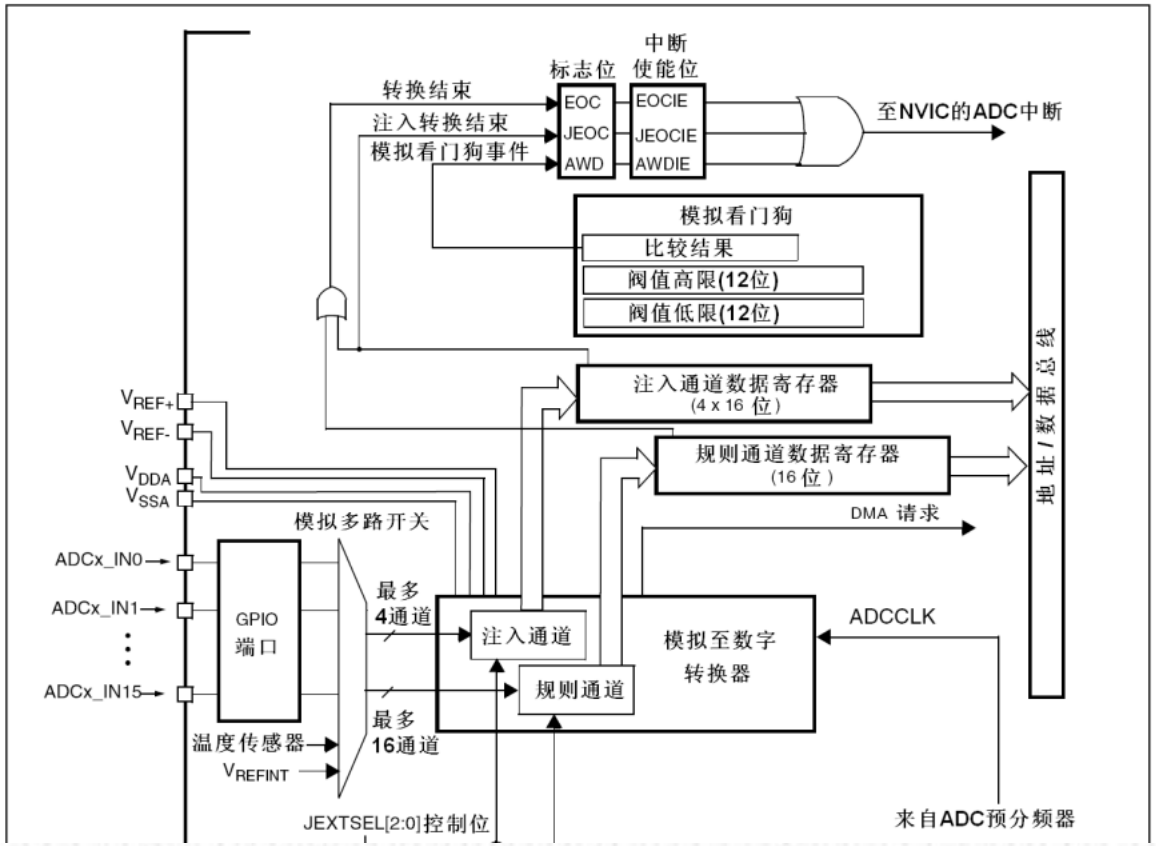
- 12位逐次逼近型的模拟数字转换器。
- 最多带3个ADC控制器
- 最多支持18个通道，可最多测量16个外部和2个内部信号源。
- 支持单次和连续转换模式
- 转换结束，注入转换结束，和发生模拟看门狗事件时产生中断。
- 从通道0到通道n的自动扫描模式
- 自校准
- 带内嵌数据一致性的数据对齐
- 采样间隔可以按通道分别编程
- 规则转换和注入转换均有外部触发选项

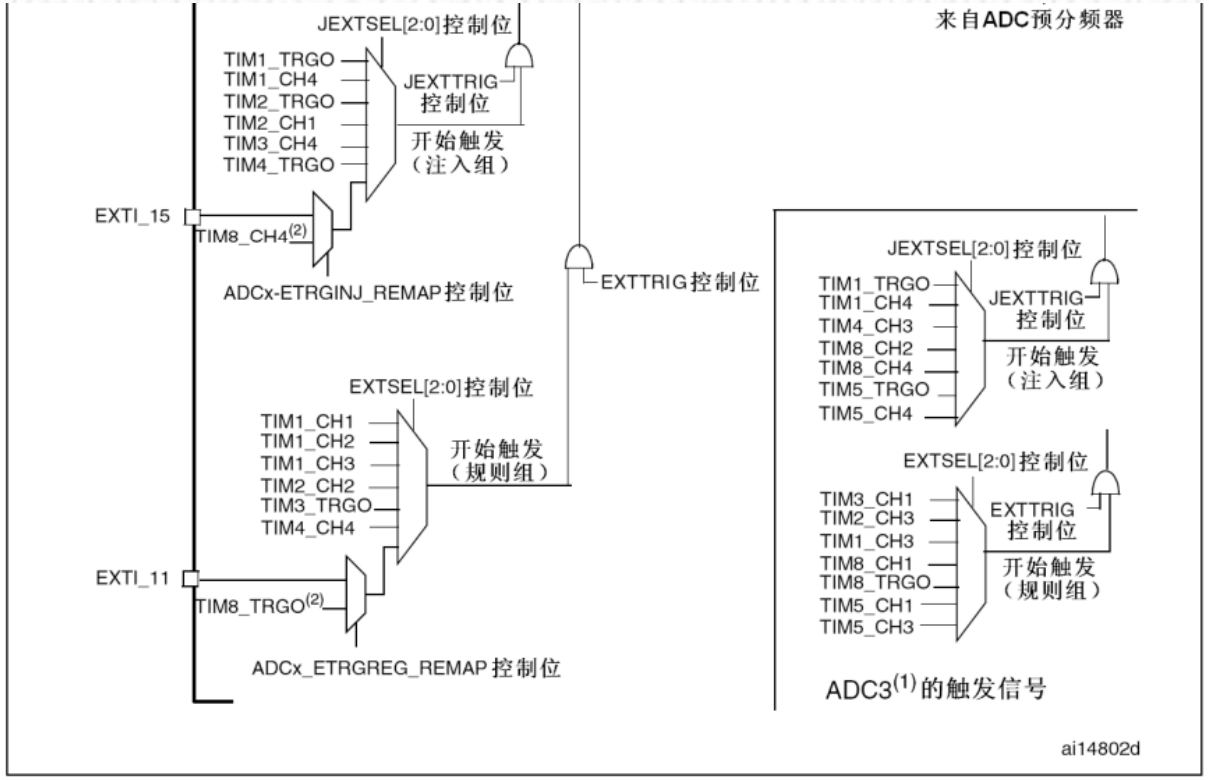
1. ADC Description

-- Main features(continued)

- 间断模式
- 双重模式(带2个或以上ADC的器件)
- ADC转换时间(STM32F103xx增强型产品)：
  - 最大转换速度为1MHz，在**ADCCLK=14MHz**，采样周期为1.5个ADC时钟下得到，此时时钟为56MHz
  - 时钟为72MHz为1.17μs
- ADC供电要求： 2.4V到3.6V
- ADC输入范围： $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- 规则通道转换期间有DMA请求产生
- 由时钟控制器提供的ADCCLK时钟和PCLK2(APB2时钟)同步

1. ADC Description
-- ADC functional description





1. ADC3的规则转换和注入转换触发与ADC1和ADC2的不同。
2. TIM8_CH4和TIM8_TRGO及它们的重映射位只存在于大容量产品中。

Single ADC block diagram

1. ADC Description
-- ADC functional description(continued)

ADC pins

名称	信号类型	注解
V_{REF+}	输入，模拟参考正极	ADC使用的高端/正极参考电压，$2.4V \le V_{REF+} \le V_{DDA}$
V_{DDA}(1)	输入，模拟电源	等效于V_{DD}的模拟电源且：$2.4V \le V_{DDA} \le V_{DD}(3.6V)$
V_{REF-}	输入，模拟参考负极	ADC使用的低端/负极参考电压，$V_{REF-} = V_{SSA}$
V_{SSA}(1)	输入，模拟电源地	等效于V_{SS}的模拟电源地
ADCx_IN[15:0]	模拟输入信号	16个模拟输入通道

1. V_{DDA}和V_{SSA}应该分别连接到V_{DD}和V_{SS}。

1. ADC Description

-- Channel selection

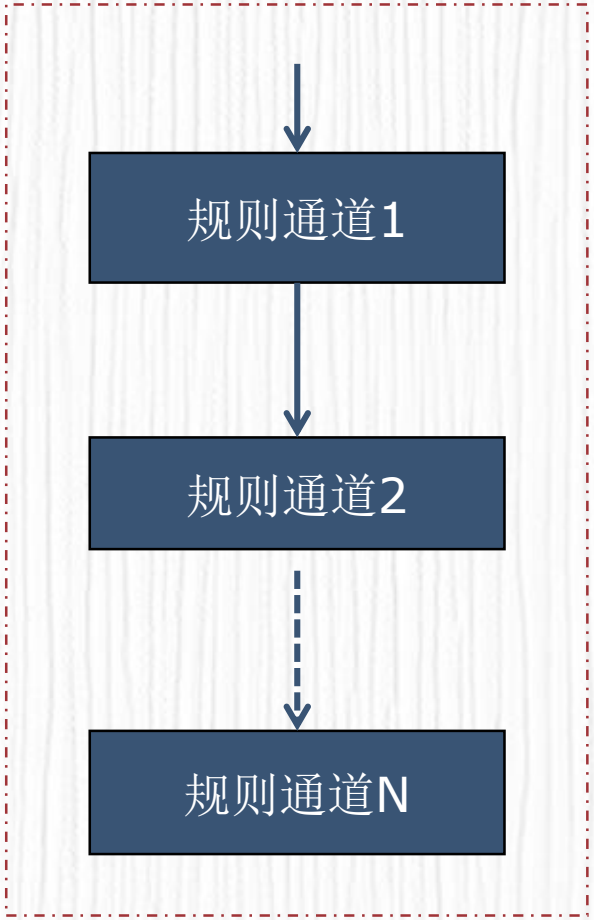
- 有16个多路通道
- 可以把转换组织成两组：规则组和注入组。
- 在任意多个通道上以任意顺序进行的一系列转换构成成组转换。例如，可以如下顺序完成转换：通道3、通道8、通道2、通道2、通道0、通道2、通道2、通道15。
- 规则组由多达16个转换组成。
- 注入组由多达4个转换组成。
- 温度传感器和通道ADC1_IN16相连接，内部参照电压V_{REFINT}和ADC1_IN17相连接。可以按注入或规则通道对这两个内部通道进行转换。

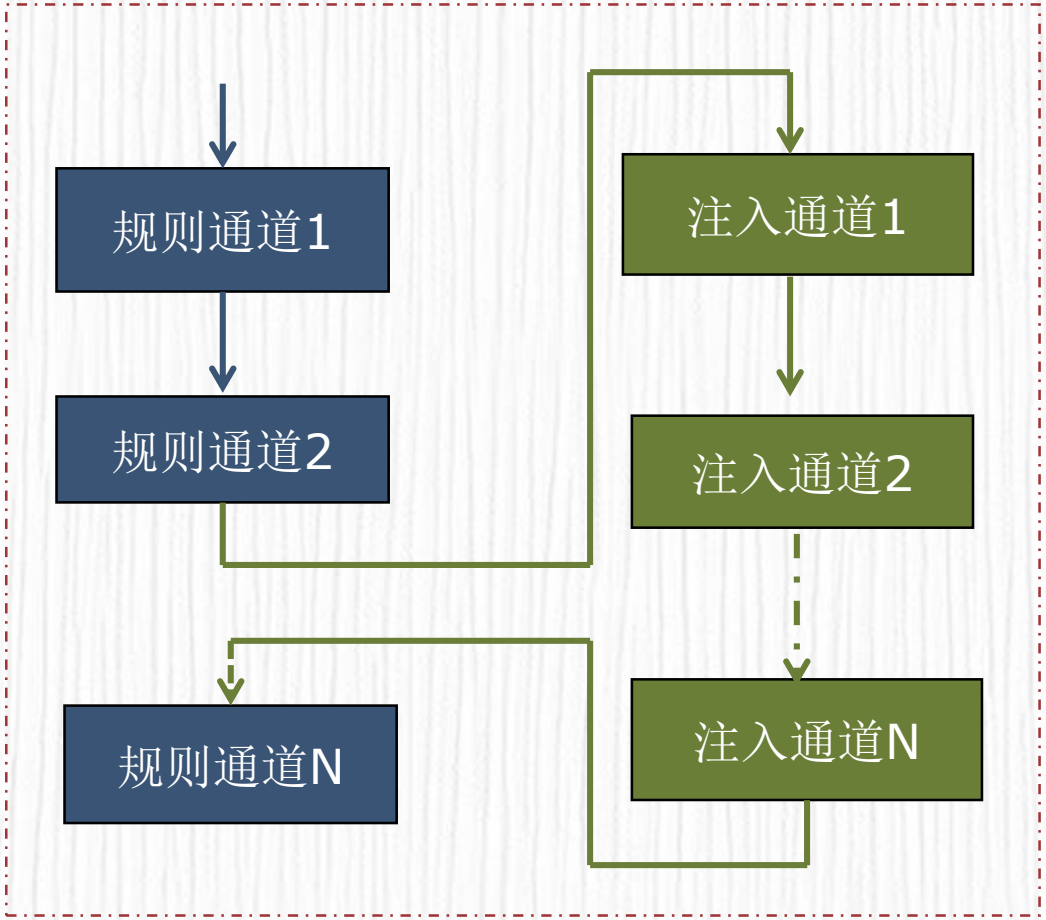
1. ADC Description

-- Channel selection(continued)

- 规则通道组：相当正常运行的程序。
  - 最多16个通道。
  - 规则通道和它的转换顺序在ADC_SQRx寄存器中选择
  - 规则组转换的总数应写入ADC_SQR1寄存器的L[3:0]中
- 注入通道组：相当于中断。
  - 最多4个通道。
  - 注入组和它的转换顺序在ADC_JSQR寄存器中选择。
  - 注入组里转化的总数应写入ADC_JSQR寄存器的L[1:0]中

1. ADC Description
-- Channel selection(continued)





1. ADC Description
-- Channel relationship of STM32F103

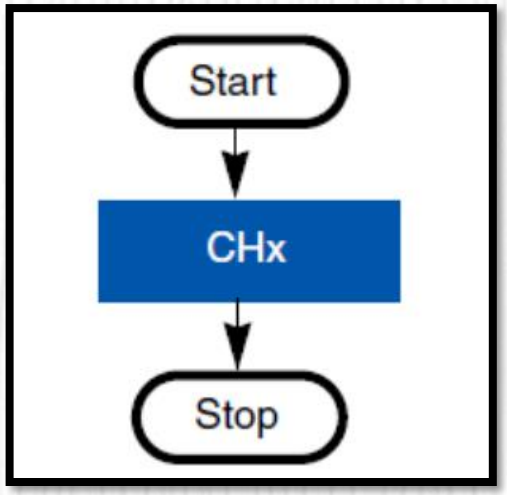
Relationship between ADC channel and GPIO

Channel	ADC1	ADC2	ADC3		Channel	ADC1	ADC2	ADC3
0	PA0	PA0	PA0		9	PB1	PB1	
1	PA1	PA1	PA1		10	PC0	PC0	PC0
2	PA2	PA2	PA2		11	PC1	PC1	PC1
3	PA3	PA3	PA3		12	PC2	PC2	PC2
4	PA4	PA4			13	PC3	PC3	PC3
5	PA5	PA5			14	PC4	PC4	
6	PA6	PA6			15	PC5	PC5	
7	PA7	PA7			16	Temperature Sensor Channel		
8	PB0	PB0			17	V_{REFINT} Channel		

1. ADC Description

-- Single conversion mode

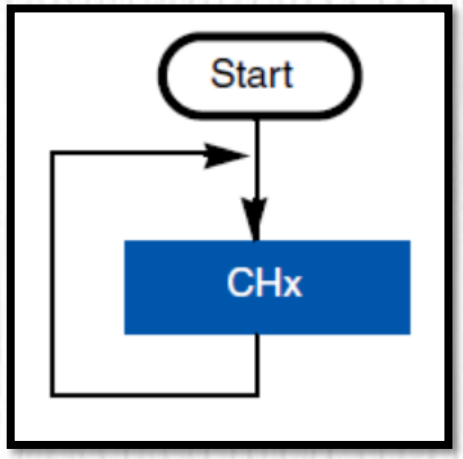
- 单次转换模式下，ADC只执行一次转换。
- 如果一个规则通道被转换完成：
  - 转换数据被储存在16位ADC_DR寄存器中
  - EOC(转换结束)标志被设置
  - 如果设置了EOCIE，则产生中断。
- 如果一个注入通道被转换完成：
  - 转换数据被储存在16位的ADC_DRJ1寄存器中
  - JEOC(注入转换结束)标志被设置
  - 如果设置了JEOCIE位，则产生中断。
- 然后ADC停止



1. ADC Description

-- Continuous conversion mode

- 在连续转换模式中，前面ADC转换一结束马上启动另一次转换。
- 如果一个规则通道被转换：
  - 转换数据被储存在16位的ADC_DR寄存器中
  - EOC(转换结束)标志被设置
  - 如果设置了EOCIE，则产生中断。
- 如果一个注入通道被转换：
  - 转换数据被储存在16位的ADC_DRJ1寄存器中
  - JEOC(注入转换结束)标志被设置
  - 如果设置了JEOCIE位，则产生中断。



1. ADC Description

-- Scan mode

- 扫描模式用来扫描一组模拟通道。
- 扫描模式可通过设置ADC_CR1寄存器的SCAN位来选择。
- 一旦被设置， ADC扫描所有被ADC_SQRX寄存器(对规则通道)或ADC_JSQR(对注入通道)选中的所有通道。
- 在每个组的每个通道上执行单次转换。
- 在每个转换结束时，同一组的下一个通道被自动转换。
- 如果设置了CONT位，转换不会在选择组的最后一个通道上停止，而是再次从选择组的第一个通道继续转换。
- 如果设置了DMA位，在每次EOC后，DMA控制器把规则组通道的转换数据传输到SRAM中。而注入通道转换的数据总是存储在ADC_JDRx寄存器中。

1. ADC Description

-- Discontinuous mode(regular group)

- 间断模式通过设置ADC_CR1寄存器上的DISCEN位激活。
- 它可以用来执行一个短序列的n次转换(n<=8) 。
- 一个外部触发信号可以启动ADC_SQRx寄存器中描述的下一轮n次转换，直到此序列所有的转换完成为止。
- 举例：n=3，被转换的通道 = 0、 1、 2、 3、 6、 7、 9、 10
  第一次触发：转换的序列为 0、 1、 2
  第二次触发：转换的序列为 3、 6、 7
  第三次触发：转换的序列为 9、 10，并产生EOC事件
  第四次触发：转换的序列 0、 1、 2
- 注意：以间断模式转换一个规则组时，转换序列结束后不自动从头开始。当所有子组转换完成，下一次触发启动第一个子组转换。

1. ADC Description

-- Discontinuous mode(injected group)

- 此模式通过设置ADC_CR1寄存器的JDISCEN位激活。
- 在一个外部触发事件后，该模式按通道顺序逐个转换ADC_JSQR寄存器中选择的序列。
- 一个外部触发信号可以启动ADC_JSQR寄存器选择的下一个通道序列的转换，直到序列中所有的转换完成为止。
- 例子：n=1，被转换的通道 = 1、 2、 3
  第一次触发：通道1被转换
  第二次触发：通道2被转换
  第三次触发：通道3被转换，并且产生EOC和JEOC事件
  第四次触发：通道1被转换
- 注意：避免同时为规则和注入组设置间断模式。

1. ADC Description

-- Data alignment

- 数据可以左对齐或右对齐
- 注入组通道转换的数据值已经减去了在ADC_JOFRx寄存器中定义的偏移量，因此结果可以是一个负值。
- SEXT位是扩展的符号值。
- 对于规则组通道，不需减去偏移值，因此只有12个位有效。

数据右对齐

注入组

SEXT	SEXT	SEXT	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

规则组

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

数据左对齐

注入组

SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

规则组

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

1. ADC Description

-- ADC sample time

- 可编程的通道采样时间
- ADC使用若干个ADC_CLK周期对输入电压采样，采样周期数目可以通过ADC_SMPR1和ADC_SMPR2中的SMP[2:0]位更改。
- 每个通道可以分别用不同的时间采样。
- 总转换时间如下计算：

  T_{CONV} = 采样时间 + 12.5个周期
- 例如：

  当ADCCLK = 14MHz，采样时间为1.5周期

  T_{CONV} = 1.5 + 12.5 = 14周期 = 1μs

02

ADC Registers

2. ADC Registers
-- ADC_CR1

- ADC控制寄存器1，ADC control register1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								AWDEN	JAWDEN	保留		DUALMOD[3:0]			
								rw	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

位31:24	保留。必须保持为0。
位23	**AWDEN**：在规则通道上开启模拟看门狗 (Analog watchdog enable on regular channels) 该位由软件设置和清除。 0：在规则通道上禁用模拟看门狗； 1：在规则通道上使用模拟看门狗。
位22	**JAWDEN**：在注入通道上开启模拟看门狗 (Analog watchdog enable on injected channels) 该位由软件设置和清除。 0：在注入通道上禁用模拟看门狗； 1：在注入通道上使用模拟看门狗。
位21:20	保留。必须保持为0。

2. ADC Registers
-- ADC_CR1(continued)

位19:16	**DUALMOD[3:0]**：双模式选择 (Dual mode selection) 软件使用这些位选择操作模式。 0000：独立模式 0001：混合的同步规则+注入同步模式 0010：混合的同步规则+交替触发模式 0011：混合同步注入+快速交叉模式 0100：混合同步注入+慢速交叉模式 0101：注入同步模式 0110：规则同步模式 0111：快速交叉模式 1000：慢速交叉模式 1001：交替触发模式 注： 在ADC2和ADC3中这些位为保留位 在双模式中，改变通道的配置会产生一个重新开始的条件，这将导致同步丢失。建议在进行任何配置改变前关闭双模式。
位15:13	**DISCNUM[2:0]**：间断模式通道计数 (Discontinuous mode channel count) 软件通过这些位定义在间断模式下，收到外部触发后转换规则通道的数目 000：1个通道 001：2个通道 …… 111：8个通道
位12	**JDISCEN**：在注入通道上的间断模式 (Discontinuous mode on injected channels) 该位由软件设置和清除，用于开启或关闭注入通道组上的间断模式 0：注入通道组上禁用间断模式； 1：注入通道组上使用间断模式。

位11	**DISCEN**：在规则通道上的间断模式 (Discontinuous mode on regular channels) 该位由软件设置和清除，用于开启或关闭规则通道组上的间断模式 0：规则通道组上禁用间断模式； 1：规则通道组上使用间断模式。
位10	**JAUTO**：自动的注入通道组转换 (Automatic Injected Group conversion) 该位由软件设置和清除，用于开启或关闭规则通道组转换结束后自动的注入通道组转换 0：关闭自动的注入通道组转换； 1：开启自动的注入通道组转换。
位9	**AWDSGL**：扫描模式中在一个单一的通道上使用看门狗 (Enable the watchdog on a single channel in scan mode) 该位由软件设置和清除，用于开启或关闭由AWDCH[4:0]位指定的通道上的模拟看门狗功能 0：在所有的通道上使用模拟看门狗； 1：在单一通道上使用模拟看门狗。
位8	**SCAN**：扫描模式 (Scan mode) 该位由软件设置和清除，用于开启或关闭扫描模式。在扫描模式中，转换由ADC_SQRx或ADC_JSQRx寄存器选中的通道。 0：关闭扫描模式； 1：使用扫描模式。 注：如果分别设置了EOCIE或JEOCIE位，只在最后一个通道转换完毕后才会产生EOC或JEOC中断。
位7	**JEOCIE**：允许产生注入通道转换结束中断 (Interrupt enable for injected channels) 该位由软件设置和清除，用于禁止或允许所有注入通道转换结束后产生中断。 0：禁止JEOC中断； 1：允许JEOC中断。当硬件设置JEOC位时产生中断。

2. ADC Registers
-- ADC_CR1(continued)

位6	**AWDIE**：允许产生模拟看门狗中断 (Analog watchdog interrupt enable) 该位由软件设置和清除，用于禁止或允许模拟看门狗产生中断。在扫描模式下，如果看门狗检测到超范围的数值时，只有在设置了该位时扫描才会中止。 0：禁止模拟看门狗中断； 1：允许模拟看门狗中断。
位5	**EOCIE**：允许产生EOC中断 (Interrupt enable for EOC) 该位由软件设置和清除，用于禁止或允许转换结束后产生中断。 0：禁止EOC中断； 1：允许EOC中断。当硬件设置EOC位时产生中断。
位4:0	**AWDCH[4:0]**：模拟看门狗通道选择位 (Analog watchdog channel select bits) 这些位由软件设置和清除，用于选择模拟看门狗保护的输入通道。 00000：ADC模拟输入通道0 00001：ADC模拟输入通道1 …… 01111：ADC模拟输入通道15 10000：ADC模拟输入通道16 10001：ADC模拟输入通道17 保留所有其他数值。 注：ADC1的模拟输入通道16和通道17在芯片内部分别连到了温度传感器和V_{REFINT}。 ADC2的模拟输入通道16和通道17在芯片内部连到了VSS。 ADC3模拟输入通道9、14、15、16、17与Vss相连。

2. ADC Registers
-- ADC_CR2

- ADC控制寄存器2，ADC control register2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								TSVREFE	SWSTART	JSWSTART	EXTTRIG	EXTSEL[2:0]			保留
								rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXTTRIG	JEXTSEL[2:0]			ALIGN	保留		DMA	保留				RSTCAL	CAL	CONT	ADON
rw	rw	rw	rw	rw			rw					rw	rw	rw	rw

位31:24	保留。必须保持为0。
位23	**TSVREFE**：温度传感器和V_{REFINT}使能 (Temperature sensor and V_{REFINT} enable) 该位由软件设置和清除，用于开启或禁止温度传感器和V_{REFINT}通道。在多于1个ADC的器件中，该位仅出现在ADC1中。 0：禁止温度传感器和V_{REFINT}； 1：启用温度传感器和V_{REFINT}。
位22	**SWSTART**：开始转换规则通道 (Start conversion of regular channels) 由软件设置该位以启动转换，转换开始后硬件马上清除此位。如果在EXTSEL[2:0]位中选择了SWSTART为触发事件，该位用于启动一组规则通道的转换， 0：复位状态； 1：开始转换规则通道。
位21	**JSWSTART**：开始转换注入通道 (Start conversion of injected channels) 由软件设置该位以启动转换，软件可清除此位或在转换开始后硬件马上清除此位。如果在JEXTSEL[2:0]位中选择了JSWSTART为触发事件，该位用于启动一组注入通道的转换， 0：复位状态； 1：开始转换注入通道。

2. ADC Registers
-- ADC_CR2(continued)

位20	**EXTTRIG**：规则通道的外部触发转换模式 (External trigger conversion mode for regular channels) 该位由软件设置和清除，用于开启或禁止可以启动规则通道组转换的外部触发事件。 0：不用外部事件启动转换； 1：使用外部事件启动转换。
位19:17	**EXTSEL[2:0]**：选择启动规则通道组转换的外部事件 (External event select for regular group) 这些位选择用于启动规则通道组转换的外部事件 ADC1和ADC2的触发配置如下 000：定时器1的CC1事件　100：定时器3的TRGO事件 001：定时器1的CC2事件　101：定时器4的CC4事件 010：定时器1的CC3事件　110：EXTI线11/ TIM8_TRGO事件，仅大容量产品具有TIM8_TRGO功能 011：定时器2的CC2事件　111：SWSTART ADC3的触发配置如下 000：定时器3的CC1事件　100：定时器8的TRGO事件 001：定时器2的CC3事件　101：定时器5的CC1事件 010：定时器1的CC3事件　110：定时器5的CC3事件 011：定时器8的CC1事件　111：SWSTART
位16	保留。必须保持为0。
位15	**JEXTTRIG**：注入通道的外部触发转换模式 (External trigger conversion mode for injected channels) 该位由软件设置和清除，用于开启或禁止可以启动注入通道组转换的外部触发事件。 0：不用外部事件启动转换； 1：使用外部事件启动转换。

位14:12	**JEXTSEL[2:0]**：选择启动注入通道组转换的外部事件 (External event select for injected group) 这些位选择用于启动注入通道组转换的外部事件。 ADC1和ADC2的触发配置如下 000：定时器1的TRGO事件　100：定时器3的CC4事件 001：定时器1的CC4事件　101：定时器4的TRGO事件 010：定时器2的TRGO事件　110：EXTI线15/TIM8_CC4事件(仅大容量产品具有TIM8_CC4) 011：定时器2的CC1事件　111：JSWSTART ADC3的触发配置如下 000：定时器1的TRGO事件　100：定时器8的CC4事件 001：定时器1的CC4事件　101：定时器5的TRGO事件 010：定时器4的CC3事件　110：定时器5的CC4事件 011：定时器8的CC2事件　111：JSWSTART
位11	**ALIGN**：数据对齐 (Data alignment) 该位由软件设置和清除。参考图29和图30。 0：右对齐； 1：左对齐。
位10:9	保留。必须保持为0。
位8	**DMA**：直接存储器访问模式 (Direct memory access mode) 该位由软件设置和清除。详见DMA控制器章节。 0：不使用DMA模式； 1：使用DMA模式。 注：只有ADC1和ADC3能产生DMA请求。

2. ADC Registers
-- ADC_CR2(continued)

位7:4	保留。必须保持为0。
位3	**RSTCAL**：复位校准 (Reset calibration) 该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除。 0：校准寄存器已初始化； 1：初始化校准寄存器。 注：如果正在进行转换时设置RSTCAL，清除校准寄存器需要额外的周期。
位2	**CAL**：A/D校准 (A/D Calibration) 该位由软件设置以开始校准，并在校准结束时由硬件清除。 0：校准完成； 1：开始校准。
位1	**CONT**：连续转换 (Continuous conversion) 该位由软件设置和清除。如果设置了此位，则转换将连续进行直到该位被清除。 0：单次转换模式； 1：连续转换模式。

2. ADC Registers
-- ADC_SMPR1

- ADC采样时间寄存器1，ADC sample time register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								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
SMP 15_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

位31:24	保留。必须保持为0。
位23:0	**SMPx[2:0]**：选择通道x的采样时间 (Channel x Sample time selection) 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。 000：1.5周期　　100：41.5周期 001：7.5周期　　101：55.5周期 010：13.5周期　　110：71.5周期 011：28.5周期　　111：239.5周期 注：ADC1的模拟输入通道16和通道17在芯片内部分别连到了温度传感器和V_{REFINT}。 ADC2的模拟输入通道16和通道17在芯片内部连到了Vss。 ADC3模拟输入通道14、15、16、17与Vss相连

2. ADC Registers
-- ADC_SMPR2

- ADC采样时间寄存器2，ADC sample time register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		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
SMP5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位31:30	保留。必须保持为0。
位29:0	**SMPx[2:0]**：选择通道x的采样时间 (Channel x Sample time selection) 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。 000：1.5周期　　100：41.5周期 001：7.5周期　　101：55.5周期 010：13.5周期　　110：71.5周期 011：28.5周期　　111：239.5周期 注：ADC3模拟输入通道9与Vss相连

2. ADC Registers
-- ADC_SQR1

- ADC规则序列寄存器1，ADC regular sequence register 1

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

位31:24	保留。必须保持为0。
位23:20	**L[3:0]**：规则通道序列长度 (Regular channel sequence length) 这些位由软件定义在规则通道转换序列中的通道数目。 0000：1个转换 0001：2个转换 …… 1111：16个转换
位19:15	**SQ16[4:0]**：规则序列中的第16个转换 (16th conversion in regular sequence) 这些位由软件定义转换序列中的第16个转换通道的编号(0~17)。
位14:10	**SQ15[4:0]**：规则序列中的第15个转换 (15th conversion in regular sequence)
位9:5	**SQ14[4:0]**：规则序列中的第14个转换 (14th conversion in regular sequence)
位4:0	**SQ13[4:0]**：规则序列中的第13个转换 (13th conversion in regular sequence)

2. ADC Registers
-- ADC_SQR2

- ADC规则序列寄存器2，ADC regular sequence register 2

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

位31:30	保留。必须保持为0。
位29:25	**SQ12[4:0]**：规则序列中的第12个转换 (12th conversion in regular sequence) 这些位由软件定义转换序列中的第12个转换通道的编号(0~17)。
位24:20	**SQ11[4:0]**：规则序列中的第11个转换 (11th conversion in regular sequence)
位19:15	**SQ10[4:0]**：规则序列中的第10个转换 (10th conversion in regular sequence)
位14:10	**SQ9[4:0]**：规则序列中的第9个转换 (9th conversion in regular sequence)
位9:5	**SQ8[4:0]**：规则序列中的第8个转换 (82th conversion in regular sequence)
位4:0	**SQ7[4:0]**：规则序列中的第7个转换 (7th conversion in regular sequence)

2. ADC Registers
-- ADC_SQR3

- ADC规则序列寄存器3，ADC regular sequence register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SQ6[4:0]					SQ5[4:0]					SQ4[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
SQ4_0	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位31:30	保留。必须保持为0。
位29:25	**SQ6[4:0]**：规则序列中的第6个转换 (6th conversion in regular sequence) 这些位由软件定义转换序列中的第6个转换通道的编号(0~17)。
位24:20	**SQ5[4:0]**：规则序列中的第5个转换 (5th conversion in regular sequence)
位19:15	**SQ4[4:0]**：规则序列中的第4个转换 (4th conversion in regular sequence)
位14:10	**SQ3[4:0]**：规则序列中的第3个转换 (3rd conversion in regular sequence)
位9:5	**SQ2[4:0]**：规则序列中的第2个转换 (2nd conversion in regular sequence)
位4:0	**SQ1[4:0]**：规则序列中的第1个转换 (1st conversion in regular sequence)

2. ADC Registers
-- ADC_DR

- ADC规则数据寄存器，ADC regular data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC2DATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位31:16	**ADC2DATA[15:0]**：ADC2转换的数据 (ADC2 data) - 在ADC1中：双模式下，这些位包含了ADC2转换的规则通道数据。见11.9：双ADC模式 - 在ADC2和ADC3中：不使用这些位。
位15:0	**DATA[15:0]**：规则转换的数据 (Regular data) 这些位为只读，包含了规则通道的转换结果。数据是左对齐或右对齐，如图29和图30所示。

2. ADC Registers
-- ADC_JSQR

- ADC注入序列寄存器，ADC injected sequence register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										JL[3:0]		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

位31:22	保留。必须保持为0。
位21:20	**JL[1:0]**：注入通道序列长度 (Injected sequence length) 这些位由软件定义在规则通道转换序列中的通道数目。 00：1个转换 01：2个转换 10：3个转换 11：4个转换
位19:15	**JSQ4[4:0]**：注入序列中的第4个转换 (4th conversion in injected sequence) 这些位由软件定义转换序列中的第4个转换通道的编号(0~17)。 注：不同于规则转换序列，如果JL[1:0]的长度小于4，则转换的序列顺序是从(4-JL)开始。例如：ADC_JSQR[21:0] = 10 00011 00011 00111 00010，意味着扫描转换将按下列通道顺序转换：7、3、3，而不是2、7、3。
位14:10	**JSQ3[4:0]**：注入序列中的第3个转换 (3rd conversion in injected sequence)
位9:5	**JSQ2[4:0]**：注入序列中的第2个转换 (2nd conversion in injected sequence)
位4:0	**JSQ1[4:0]**：注入序列中的第1个转换 (1st conversion in injected sequence)

2. ADC Registers
-- ADC_JDRx

- ADC注入数据寄存器x，ADC injected data register x

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位31:16	保留。必须保持为0。
位21:20	**JDATA[15:0]**：注入转换的数据 (Injected data) 这些位为只读，包含了注入通道的转换结果。数据是左对齐或右对齐，如图29和图30所示。

2. ADC Registers
-- ADC_SR

- ADC状态寄存器，ADC status register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											STRT	JSTRT	JEOC	EOC	AWD
											rc w0	rc w0	rc w0	rc w0	rc w0

2. ADC Registers
-- ADC_SR(continued)

位31:15	保留。必须保持为0。
位4	**STRT**：规则通道开始位 (Regular channel Start flag) 该位由硬件在规则通道转换开始时设置，由软件清除。 0：规则通道转换未开始； 1：规则通道转换已开始。
位3	**JSTRT**：注入通道开始位 (Injected channel Start flag) 该位由硬件在注入通道组转换开始时设置，由软件清除。 0：注入通道组转换未开始； 1：注入通道组转换已开始。
位2	**JEOC**：注入通道转换结束位 (Injected channel end of conversion) 该位由硬件在所有注入通道组转换结束时设置，由软件清除 0：转换未完成； 1：转换完成。
位1	**EOC**：转换结束位 (End of conversion) 该位由硬件在(规则或注入)通道组转换结束时设置，由软件清除或由读取ADC_DR时清除 0：转换未完成； 1：转换完成。
位0	**AWD**：模拟看门狗标志位 (Analog watchdog flag) 该位由硬件在转换的电压值超出了ADC_LTR和ADC_HTR寄存器定义的范围时设置，由软件清除 0：没有发生模拟看门狗事件； 1：发生模拟看门狗事件。

03

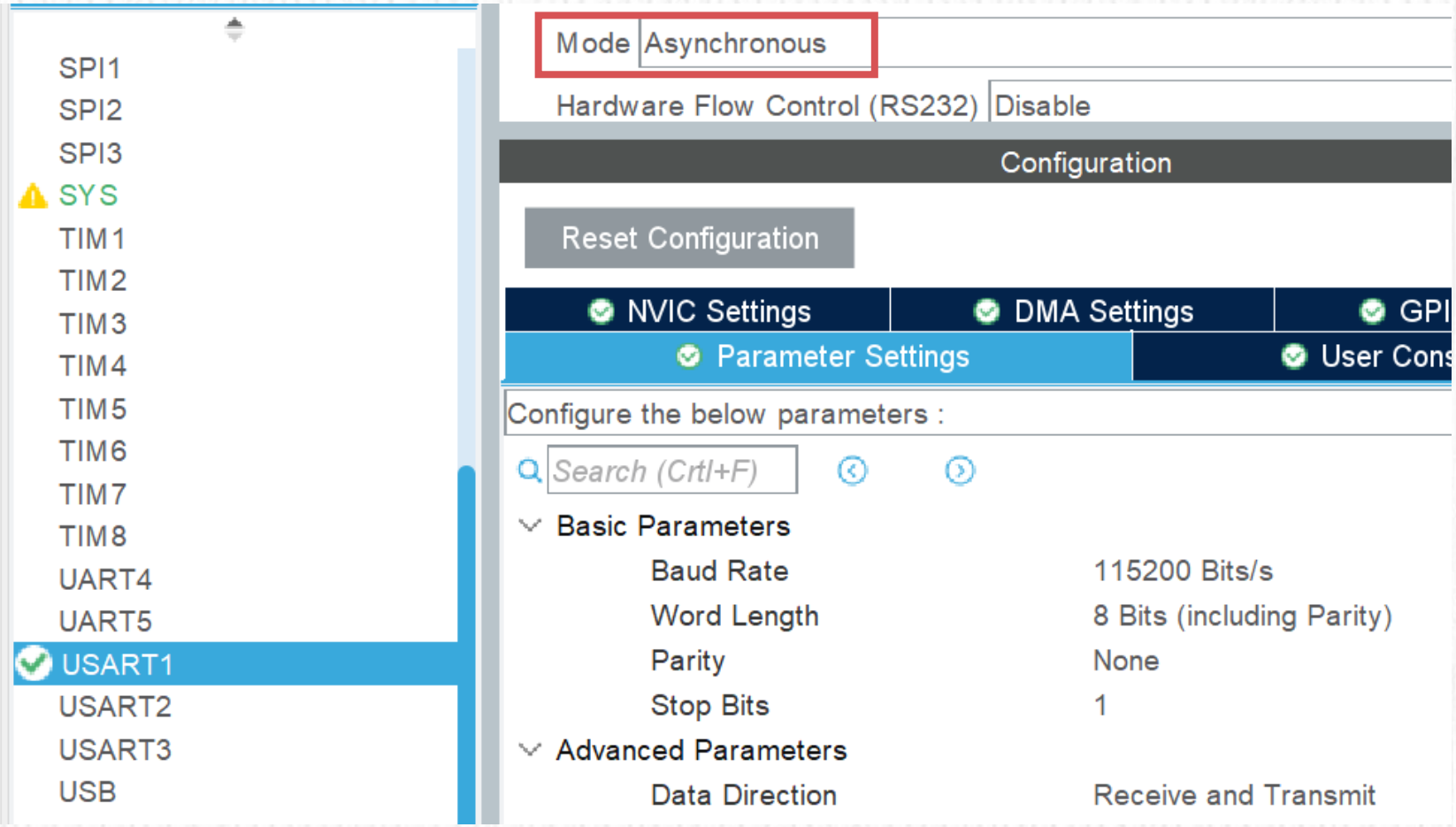
How to Program

3. How to Program

- Our Goal
  - Use ADC1 to get the voltage of GPIO pin，and show it with USART1

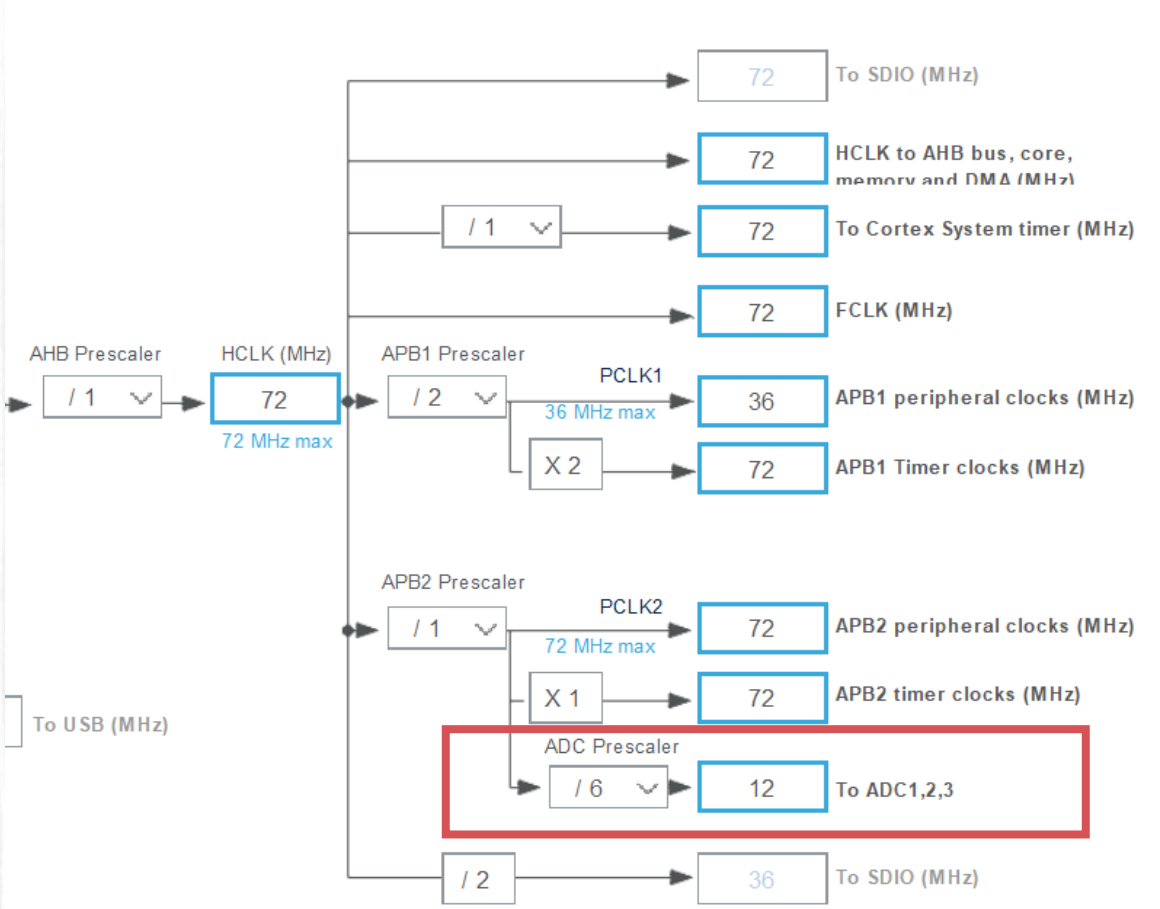
3. How to Program

- Configure USART1
  - Set the USART1 as asynchronous mode



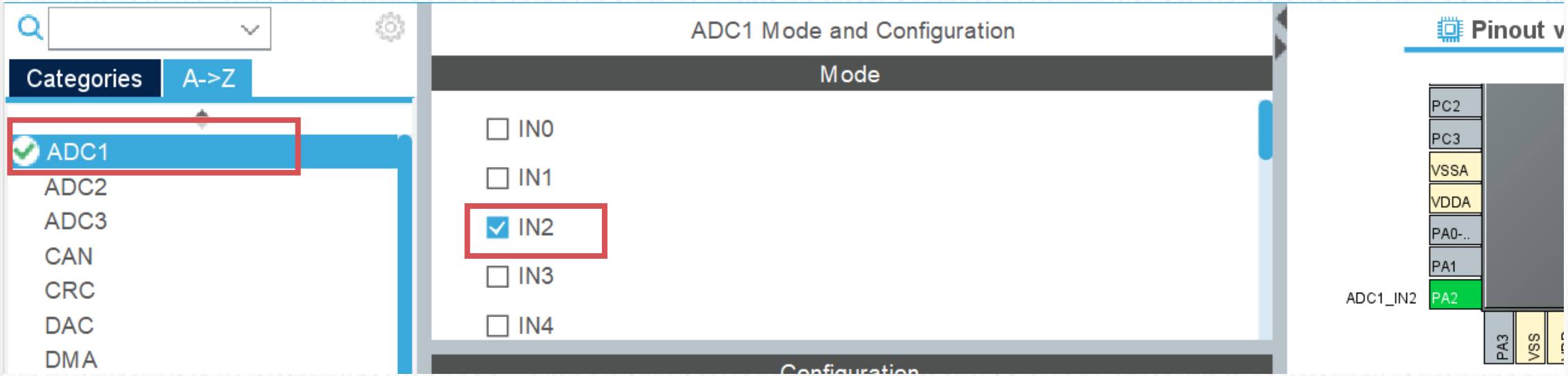
3. How to Program

- Configure Clock
  - ADC Clock frequency should be less than or equal to 14MHz



3. How to Program

- Configure ADC1
  - Go to the Analog categories, click ADC1 and select IN2, which means we enable the channel 2 of ADC1 and we are able to measure the voltage of PA2

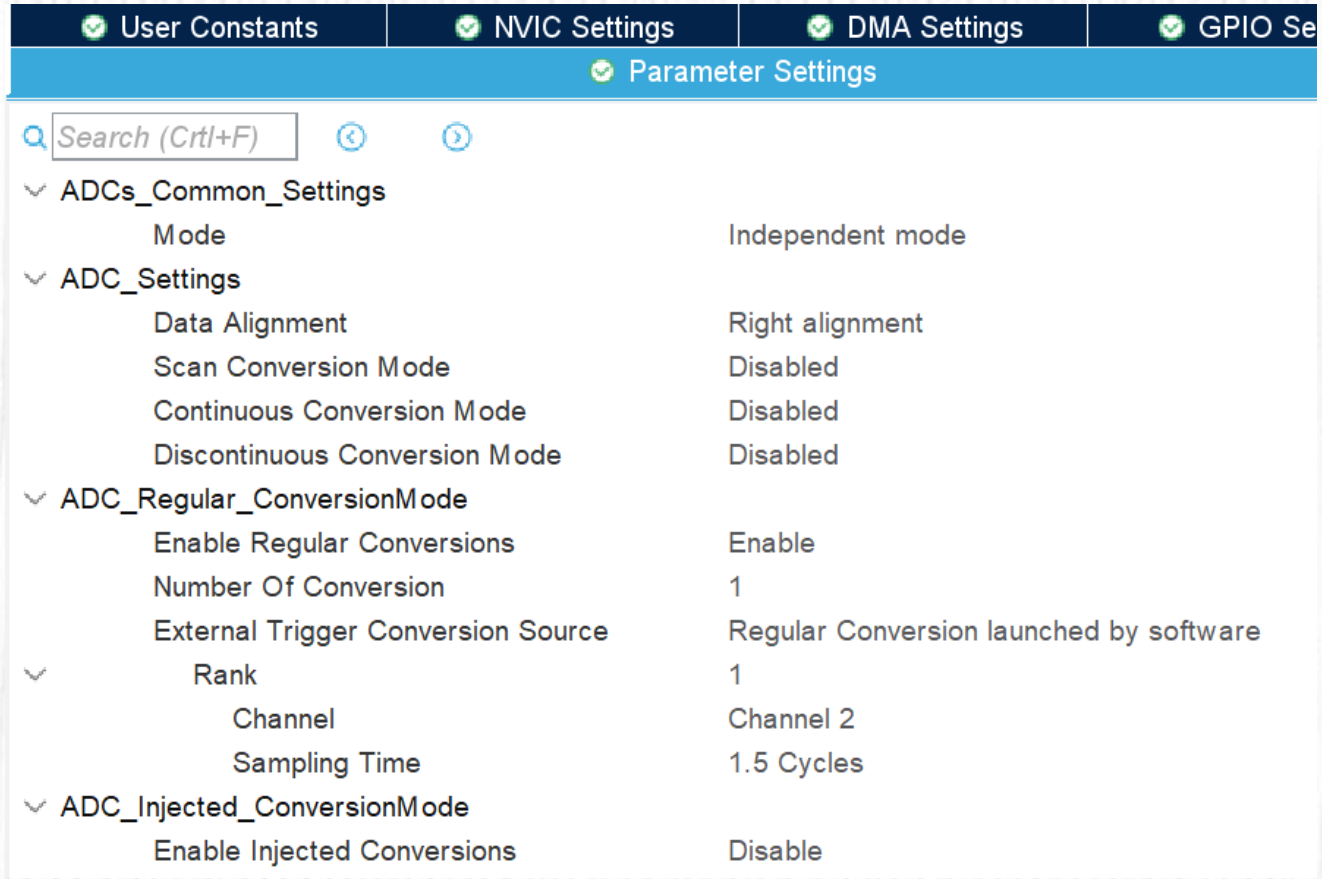


3. How to Program

- Some API of HAL we used
- ADC conversion by polling(轮询):
- Activate the ADC peripheral and start conversions using function **HAL_ADC_Start()**
- Wait for ADC conversion completion using function **HAL_ADC_PollForConversion()** (or for injected group: HAL_ADCEx_InjectedPollForConversion())
- Retrieve conversion results using function **HAL_ADC_GetValue()** (or for injected group: HAL_ADCEx_InjectedGetValue())
- Stop conversion and disable the ADC peripheral using function **HAL_ADC_Stop()**

3. How to Program

- Configure ADC1
  - Single channel, single conversion mode
  - In single conversion mode, ADC only does one conversion, so we need to restart conversion if we want to update the measurements



3. How to Program

- Single channel, single conversion mode

```
// in main.c file
int main(void){
        //……
        uint16_t raw;
        char msg[20];
        double vol;
        HAL_UART_Transmit(&huart1, "Start\r\n", 7, HAL_MAX_DELAY);
        while (1){
                HAL_ADC_Start(&hadc1);
                // Wait for regular group conversion to be completed
                HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
                raw = HAL_ADC_GetValue(&hadc1);  // Get ADC value
                vol = (double)raw * (3.3/4096);  // the voltage should be raw * (3.3/4096)(12 bits)
                sprintf(msg, “Voltage = %f\r\n", vol);  // Convert to string and print
                HAL_UART_Transmit(&huart1, (uint8_t*)msg, strlen(msg), HAL_MAX_DELAY);
        }
}
```

3. How to Program

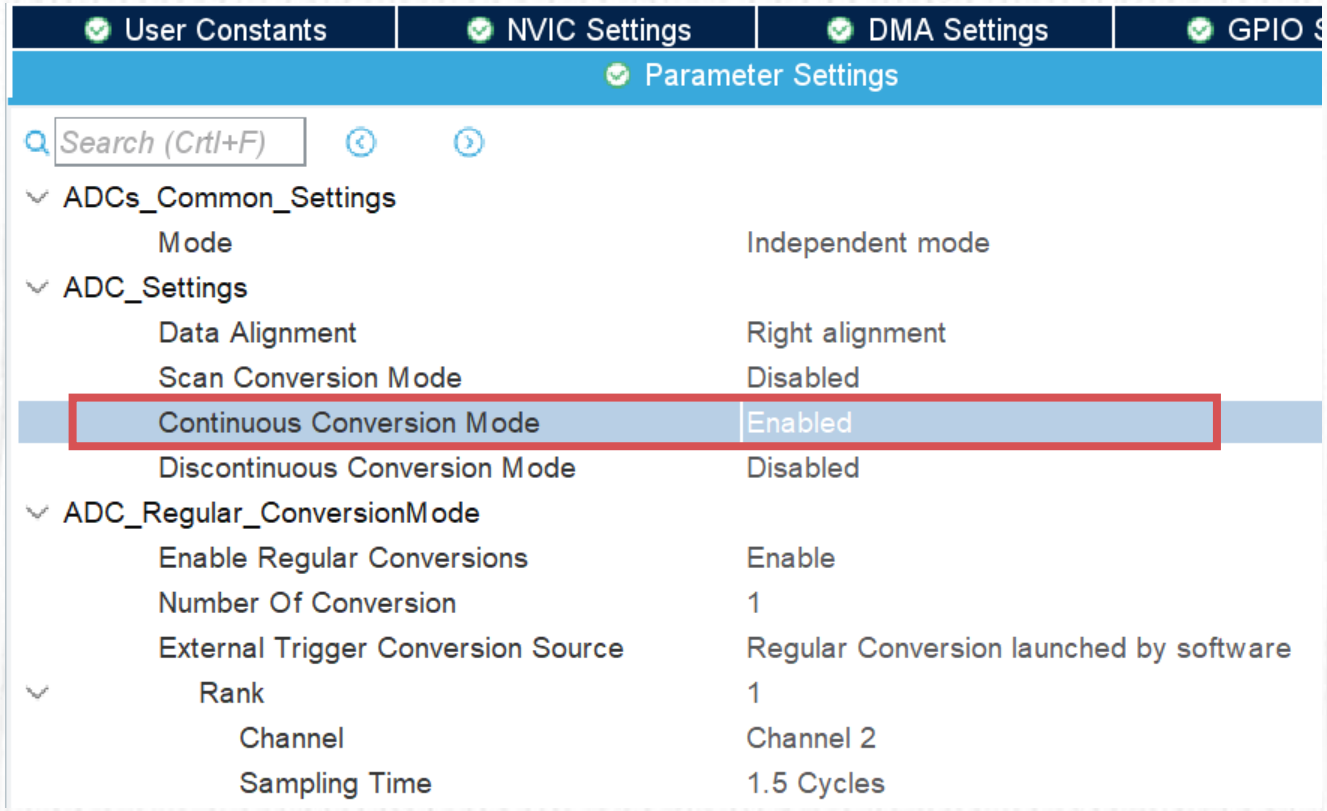
- Result

ATK XCOM XCOM V2.6

```
Voltage = 24.957831
Start
Voltage = 3.250854
Voltage = 3.254077
Voltage = 3.255688
Voltage = 3.254077
Voltage = 3.254883
Voltage = 3.255688
Voltage = 3.254077
Voltage = 3.254077
```

3. How to Program

- Configure ADC1
  - Single channel, continuous conversion mode
  - In continuous conversion mode, ADC starts another conversion as soon as it finishes one, so we don't need to restart the conversion.



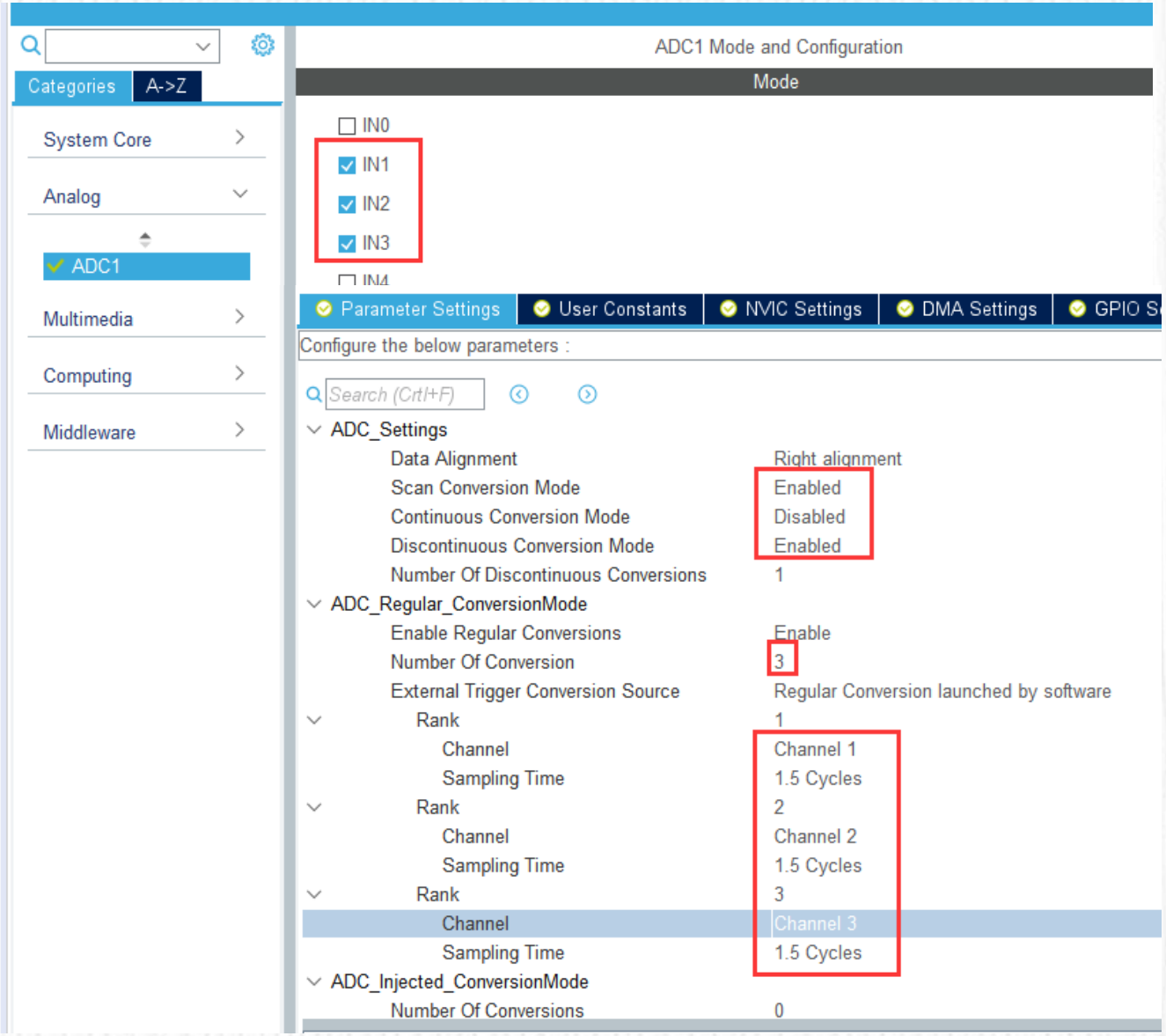
3. How to Program

- Single channel, continuous conversion mode

```
// in main.c file
int main(void){
        //……
        uint16_t raw;
        char msg[20];
        double vol;
        HAL_UART_Transmit(&huart1, "Start\r\n", 7, HAL_MAX_DELAY);
        HAL_ADC_Start(&hadc1);
        while (1){
                // Wait for regular group conversion to be completed
                HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
                raw = HAL_ADC_GetValue(&hadc1);  // Get ADC value
                vol = (double)raw * (3.3/4096);  // the voltage should be raw * (3.3/4096)(12 bits)
                sprintf(msg, “Voltage = %f\r\n", vol);  // Convert to string and print
                HAL_UART_Transmit(&huart1, (uint8_t*)msg, strlen(msg), HAL_MAX_DELAY);
        }
}
```

3. How to Program

- Configure ADC1
  - Multiple channels, single conversion mode
  - If we want to get the measurements of multiple channels and don’t want to use DMA (Direct Memory Access) and interrupt, only single conversion mode can be used.



3. How to Program

- Multiple channels, single conversion mode

```
// in main.c file
while (1)
{
        /* USER CODE END WHILE */
        /* USER CODE BEGIN 3 */
        for (int i = 0; i < 3; i++)
        {
                HAL_ADC_Start(&hadc1);
                HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
                adcBuf[i]=HAL_ADC_GetValue(&hadc1);
                sprintf(msg, "ch:%d %d\r\n", i, adcBuf[i]);
                HAL_UART_Transmit(&huart1, (uint8_t*)msg, strlen(msg), HAL_MAX_DELAY);
        }
        HAL_Delay(500);
}
```

3. How to Program

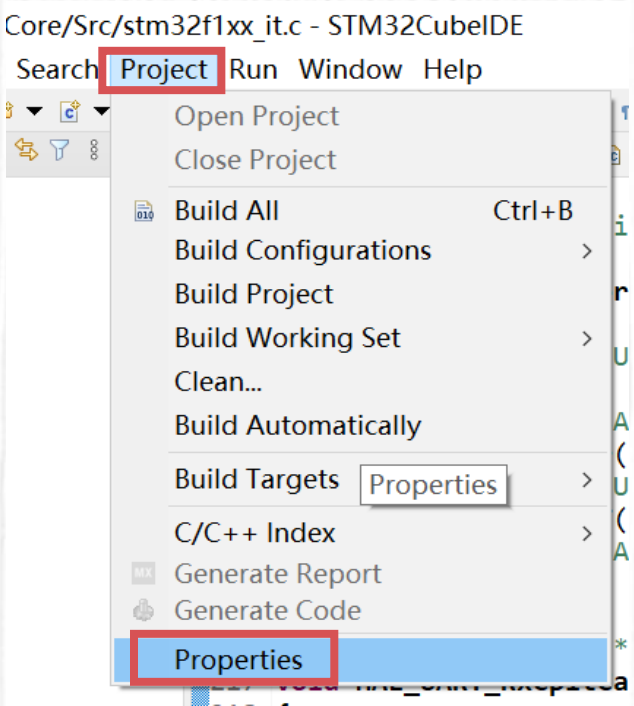
- Result

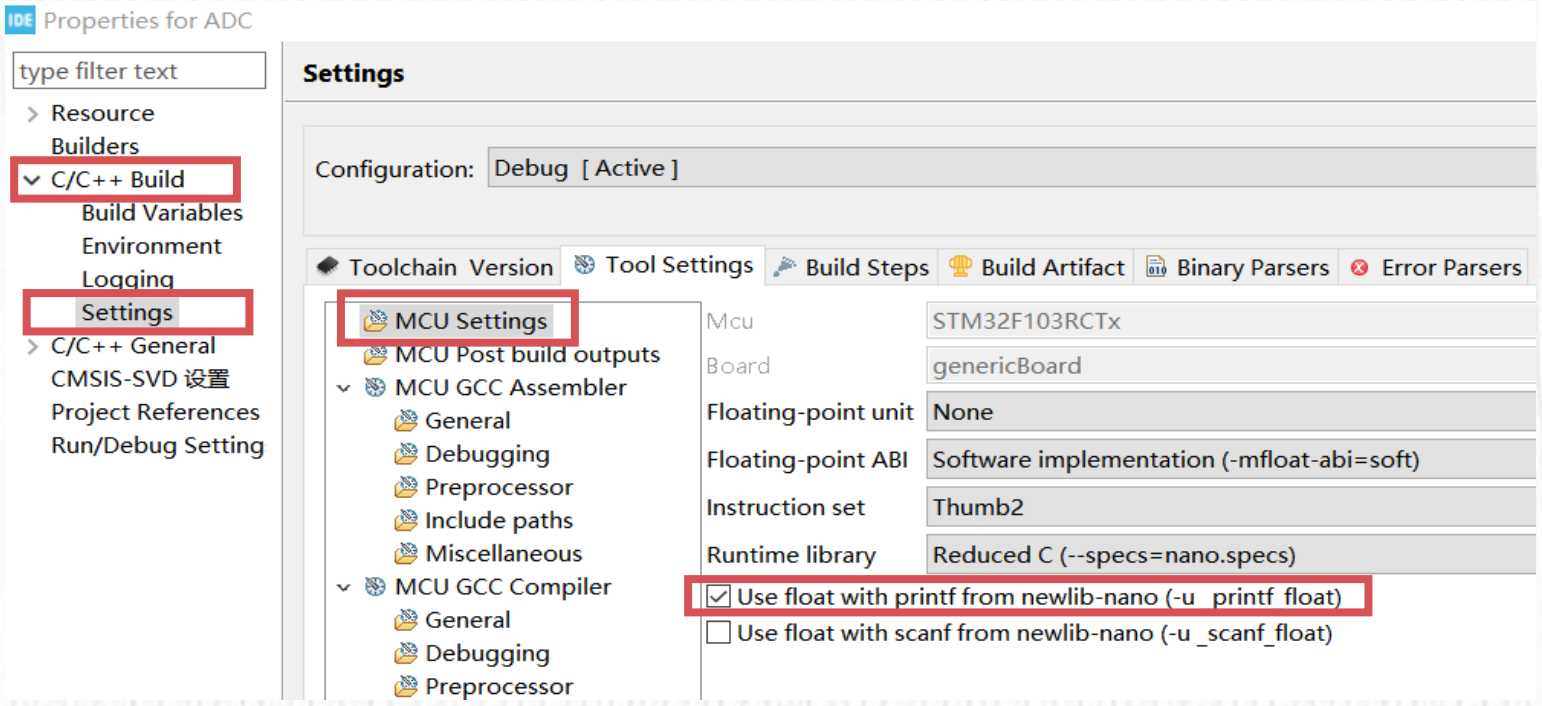
ATK XCOM XCOM V2.6

```
ch:2 561
ch:0 4039
ch:1 3571
ch:2 563
ch:0 4040
ch:1 3570
ch:2 563
ch:0 4039
ch:1 3570
ch:2 558
ch:0 4040
ch:1 3572
ch:2 549
```

3. How to Program

- To print double or float value, you should change the settings as the following figure
- [Project] -> [Properties] -> [C/C++ Build] -> [Settings] -> [MCU Settings] -> [Use float with printf from newlib-nano(-u printf float)] -> [Apply and Close]





04

Practice

4. Practice

- Configure ADC1 in different modes and run the demos on MiniSTM32 board
- Use CH4, CH5 and CH6 of ADC1 in multiple channels, single conversion mode to show the voltage of them continuously