

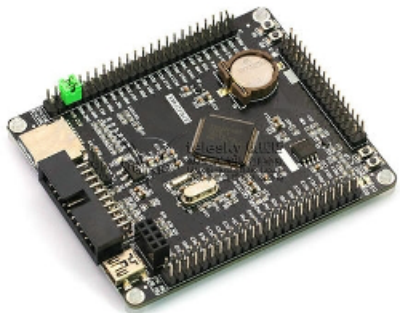


# 单片机产品设计与制作

( stm32机型 )

袁秀英 石梅香





# 项目8、利用ADC实现土壤湿度控制

## 任务8.1 认识STM32单片机的ADC



# 总目录

- 1 项目1、利用GPIO和位带操作实现温度报警
- 2 项目2、利用SysTick实现温度报警与控制
- 3 项目3、利用按键查询实现参数设定及显示
- 4 项目4、利用外部中断实现工件计数显示
- 5 项目5、利用定时器实现直流电动机PWM调速
- 6 项目6、利用计数器实现工件计数显示和打包控制
- 7 项目7、利用ADC实现土壤湿度控制



# 具体工作任务

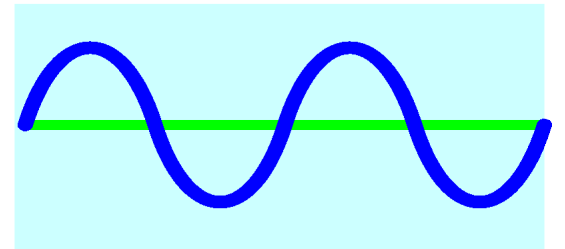
利用STM32内部ADC采集土壤湿度，当湿度值低于下限时，打开水泵开始喷灌；湿度值大于上限时，关闭水泵，停止喷灌。

- 1、进行电路设计与调试
- 2、进行程序设计与调试
- 3、进行软硬件联合调试



# 项目总目标

- 1、理解ADC ( Analog to Digital Converter ) 模/数转换器的基本概念与术语；
- 2、了解STM32内部ADC的结构与特性。
- 3、掌握基于STM32F103ZET6 ADC的电路设计方法。
- 4、掌握STM32基于HAL库函数的ADC操作编程方法；
- 5、会查找相关资料、阅读相关文献。



# 子目录



## 项目8、利用ADC实现土壤湿度控制

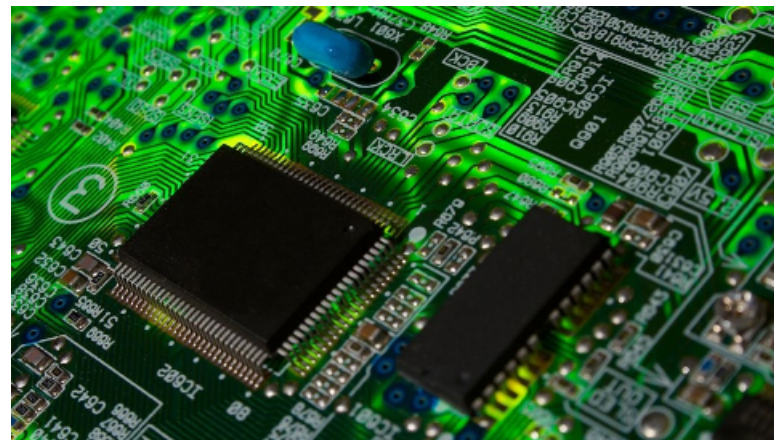
8.1 认识STM32单片机的ADC

8.2 方案设计及器件选型

8.3 电路设计与调试

8.4 程序设计与调试

8.5 STM32软硬件深入（七）



# 任务8.1 认识STM32单片机的ADC

## 本任务目标：

了解STM32F103ZET6的ADC的基本特性与原理。

## 具体工作任务：

阅读相关资料并展开讨论。



# 讨论与发言

举一些例子说明自动化系统中哪些是模拟量，哪些是数字量。

总结什么是模拟量，什么是数字量？二者有何区别？

什么是ADC？什么是DAC？

衡量ADC性能的指标有哪些？

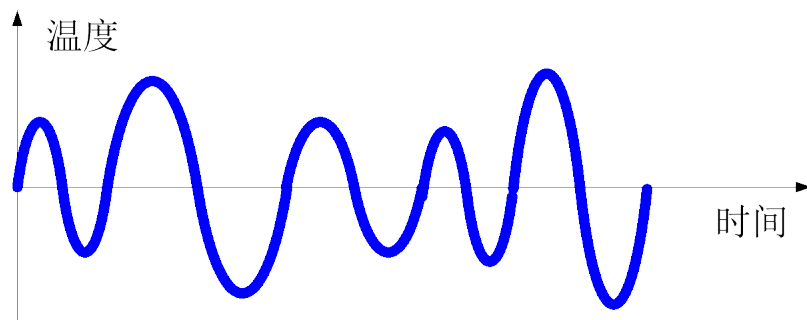




# 认识ADC

## 1、什么是模拟量，什么是数字量？

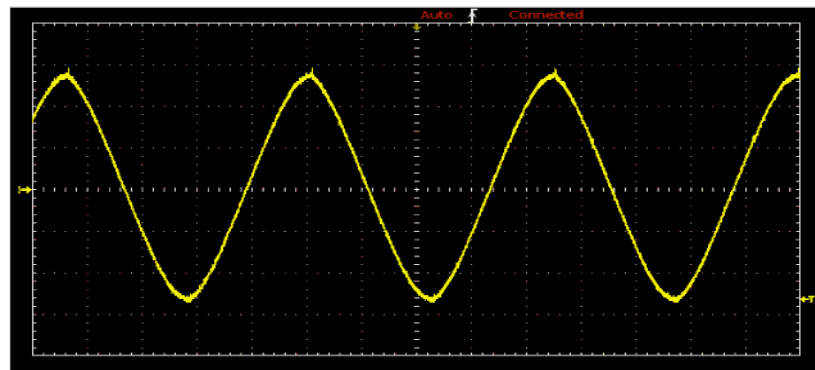
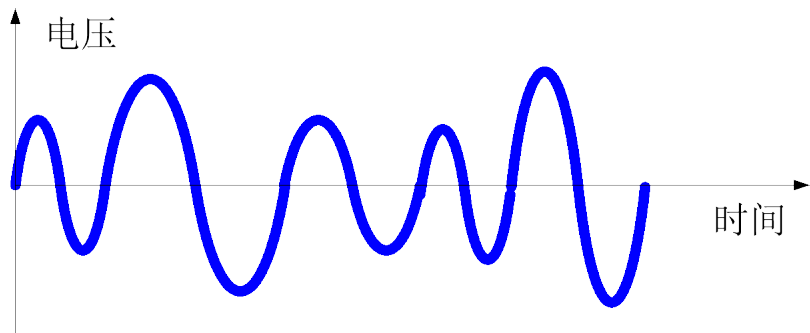
在实际生产生活中，我们大量接触在幅值和时间上都连续变化的物理量，例如0~100℃的温度、-500~+500mm的位移、0~10MPa的压力、0~100%的湿度等等。



# 认识ADC

## 1、什么是模拟量，什么是数字量？

在自动化系统中，我们也会接触大量在幅值和时间上都连续变化的电信号，例如0~3V的电压、0~5V的电压、0~10V的电压、0~10mA的电流、4~20mA的电流、0~1000Ω的电阻等等。



这种在幅值和时间上都连续变化的信号，称为模拟信号或称模拟量。

# 认识ADC

## 1、什么是模拟量，什么是数字量？

在之前几个项目中，当GPIO引脚做输入时，

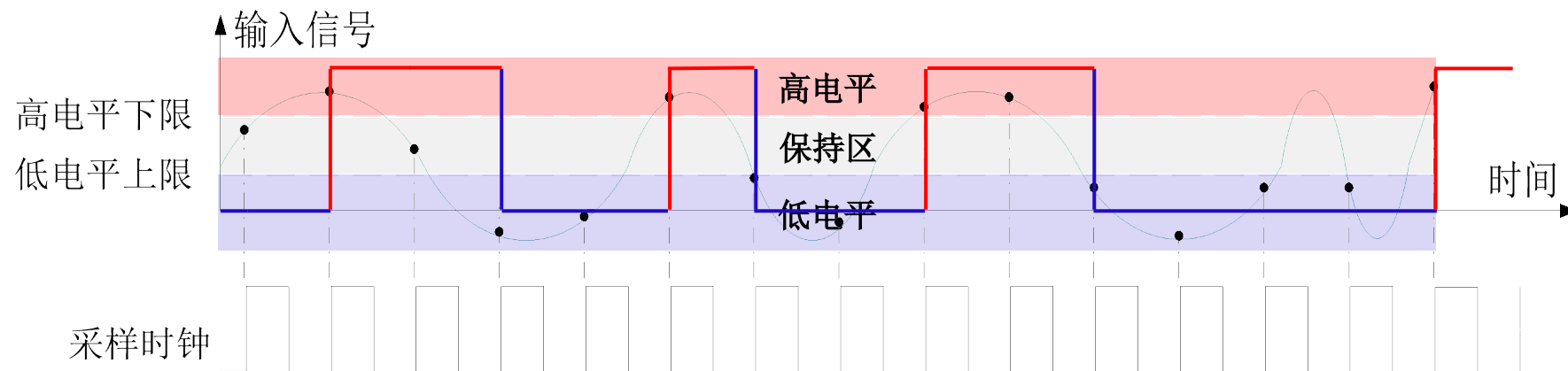
只能识别高电平或低电平信号，并不能细致区分电压的幅度，

并且只能分时采样输入

也就是说，GPIO输入输出信号在幅值和时间上是断续的。

幅值上：只有高电平和低电平

时间上：只有采样时刻的输入



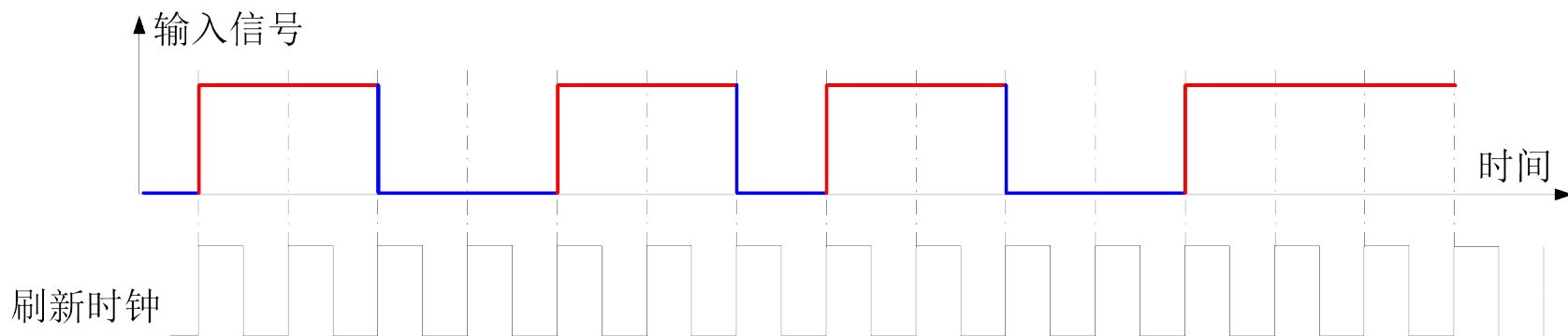
# 认识ADC

## 1、什么是模拟量，什么是数字量？

当GPIO引脚做输出时，只能输出高电平或低电平信号。

输出信号的幅值是断续的。

输出刷新必须在GPIO时钟的控制下。

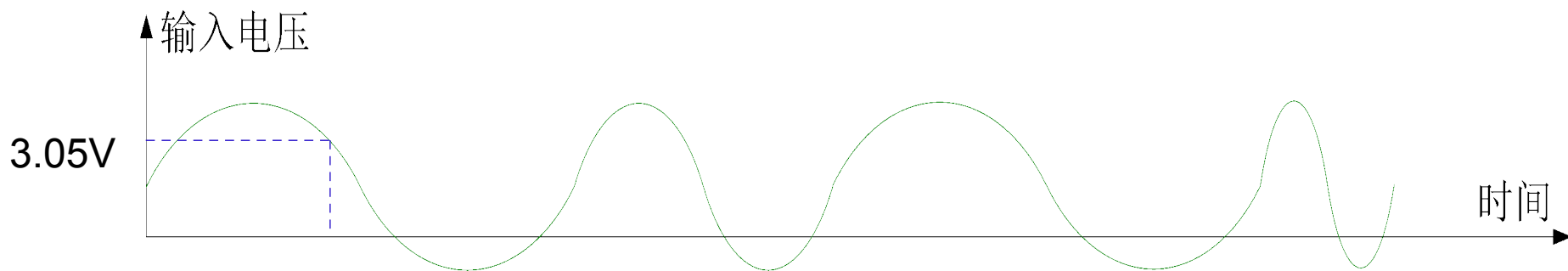


这种在幅值和时间上都不连续的信号，称为数字信号或称数字量。

# 认识ADC

## 2、什么是ADC，什么是DAC？

很多时候，我们希望STM32不仅能够接收0~3V这样的电压输入，更能够区分到底输入的是0.1V还是1.3V，还是其它，而不是简单地将输入信号分辨成3V高电平或者0V低电平。

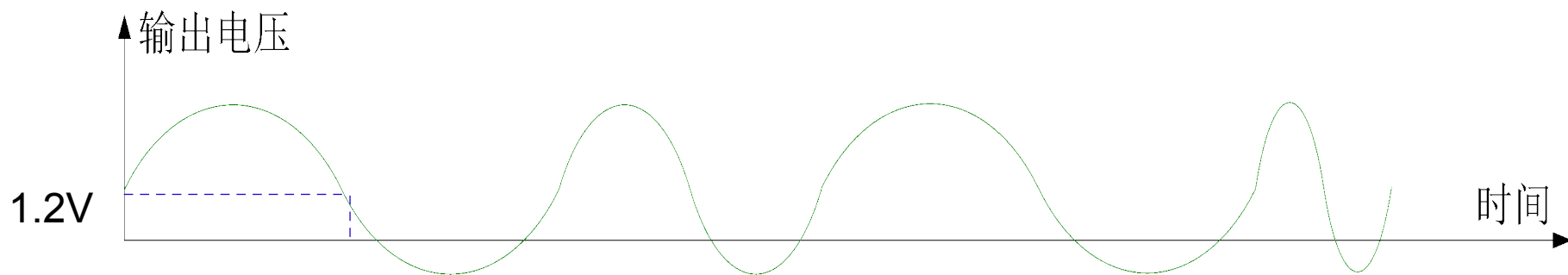


# 认识ADC

## 2、什么是ADC，什么是DAC？

很多时候，我们也希望单片机能够精确控制其输出电压的幅度为0.5V或者1.2V或者其它，而不是只输出3V高电平或者0V低电平。

也就是说我们希望STM32具有模拟量的处理能力



# 认识ADC

## 2、什么是ADC，什么是DAC？

我们希望STM32具有模拟量的处理能力

但实际上，STM32单片机内部的绝大多数器件特别是它的核心器件例如CPU都是数字器件。

如何让它们具有模拟量输入/输出信号的处理能力呢？

很显然，输入时需要一个电路，**将模拟量输入信号转换成数字量**，  
这样，CPU才能够处理模拟量输入。

这里用到的电路就叫**ADC（Analog-to-Digital Converter）**，即**模/数转换器**。

同理，输出时也需要一个电路，**将CPU输出的数字量转换成模拟量输出**。

这样的电路就叫**DAC（Digital-to-Analog Converter）**，即**数/模转换器**。



# 认识ADC

## 3、ADC的输入信号

常见的ADC有电压输入、电流输入、电阻输入等。

输入信号的范围也有多种。

STM32F103ZET6的ADC是电压输入，

要求输入电压的范围是： $V_{REF-} \leq V_{IN} \leq V_{REF+}$ 。

由于 $V_{REF-}$ 与 $V_{SSA}$ 接在一起，作为0V，如果 $V_{REF+}=3.3V$ ，

则模拟电压输入的范围就是0~3.3V。

当需要测量负电压或测量的电压信号超出范围时，要先设法将输入转换成0~3.3V。

一定记住，不要直接将高于3.3V的电压接到ADC管脚上，那样将可能烧坏芯片

# 认识ADC

## 4、ADC的输出信号及其位数

ADC的输出也就是A/D转换的结果，是n位数字量。  
根据位数的不同又分为8、12、14、16、24位等。

例如8位的ADC，其结果有8位，

对应二进制数最小是0000 0000，最大是1111 1111，对应十进制数是0~255。

而16位的ADC，其输出数字量为16位，

对应二进制数是0000 0000 0000 0000~1111 1111 1111 1111，即十进制数0~65535。

位数	电压	数字		转换公式及分辨率分辨率
		二进制	十进制	
n=8	<b>0V</b>	<b>0000 0000</b>	<b>0</b>	转换公式： $D = V_{IN} \times 256 / 3.3$ 分辨率： $3.3 / 2^8 = 12.890625\text{mV} \approx 13\text{mV}$
	12.890625mV	0000 0001	1	
	25.78125mV	0000 0010	2	
	.....	.....	.....	
	3.287109375V	1111 1111	255	
	<b>3.3V</b>	<b>0000 0000 (溢出)</b>	<b>256</b>	
n=12	<b>0V</b>	<b>0000 0000 0000</b>	<b>0</b>	转换公式： $D = V_{IN} \times 4096 / 3.3$ 分辨率： $3.3 / 2^{12} = 0.8056640625\text{mV} \approx 0.8\text{mV}$
	0.8056640625mV	0000 0000 0001	1	
	1.611328125mV	0000 0000 0010	2	
	3.2991943359375V	1111 1111 1111	4095	
	<b>3.3V</b>	<b>0000 0000 0000 (溢出)</b>	<b>4096</b>	
n=16	<b>0V</b>	<b>0000 0000 0000 0000</b>	<b>0</b>	转换公式： $D = V_{IN} \times 65536 / 3.3$ 分辨率： $3.3 / 2^{16} = 50.35400390625\mu\text{V} \approx 50\mu\text{V}$
	50.35400390625μV	0000 0000 0000 0001	1	
	3.29994964599609375V	1111 1111 1111 1111	65535	
	<b>3.3V</b>	<b>0000 0000 0000 0000 (溢出)</b>	<b>65536</b>	

# 认识ADC

## 4、ADC的输出信号及其位数

假设ADC输入模拟量为电压，位数为n，  
转换后的数字量D与输入电压 $V_{IN}$ 之间关系是：

$$D = V_{IN} \times 2^n / (V_{MAX} - V_{MIN})$$

$$V_{IN} = D \times (V_{MAX} - V_{MIN}) / 2^n$$

对于0~3.3V的输入电压， $V_{MAX} - V_{MIN} = 3.3 - 0 = 3.3$ 。

能够分辨的最小电压即分辨率是： $(V_{MAX} - V_{MIN}) / 2^n$

STM32F103ZET6的ADC是12位，分辨率大约为0.8mV。

$$V_{IN} = D \times 3.3 / 4096$$

$$D = V_{IN} \times 4096 / 3.3$$

# 认识ADC

## 5、A/D转换时间和转换速率

A/D转换时间是指完成一次A/D转换需要的时间。

A/D转换速率是转换时间的倒数，即每秒钟完成A/D转换的次数。

STM32的A/D转换所需要的时间包括采样时间和转换时间。 $T_{CNV}=T_{\text{采样}}+T_{\text{转换}}$

$T_{\text{采样}}$ 可以编程设置为1.5、7.5、13.5、27.5、41.5、55.5、71.5、239.5个ADCCLK周期。

$T_{\text{转换}}$ 是12.5个ADCCLK周期。

因此转换一次**最少需要**：1.5+12.5=14个ADCCLK周期，**最多需要**239.5+12.5=252个ADCCLK周期。

ADCCLK是ADC工作时钟，允许的最大频率为14MHz。因此：A/D转换最小时间为

14/（14MHz）=1μs，最大速率=14MHz/14=1MHz。

# 认识ADC

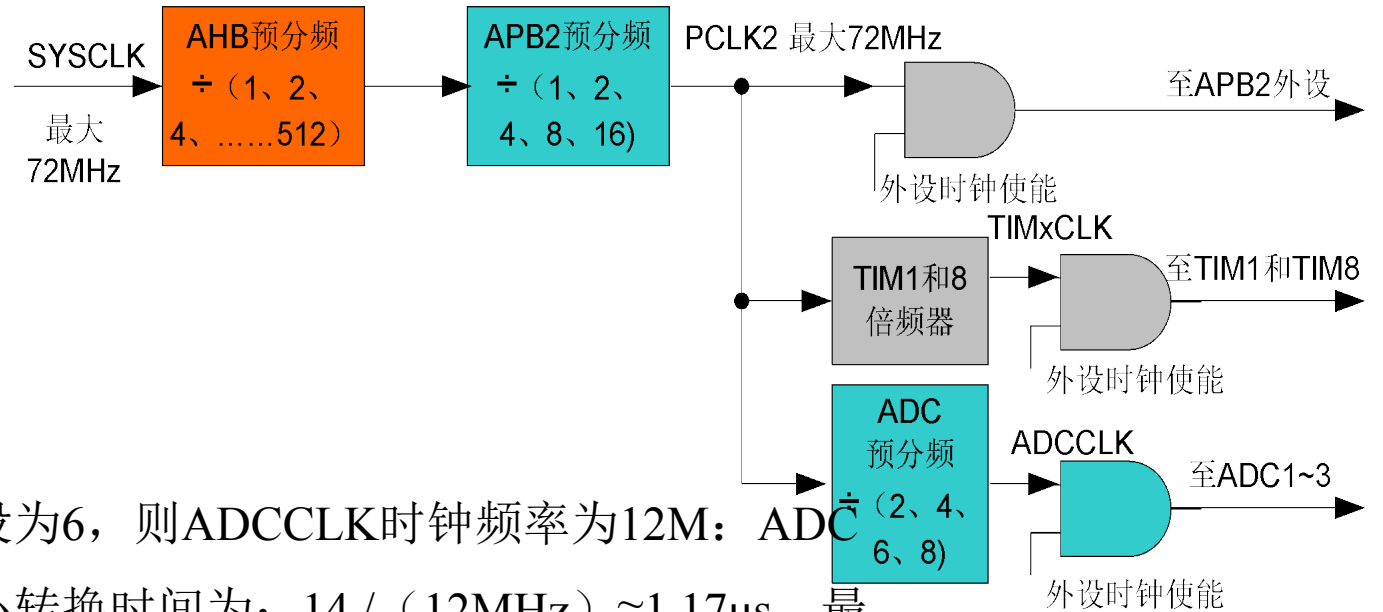
## 5、A/D转换时间和转换速率

ADCCLK由RCC提供

ADCCLK由PCLK2 经ADC预分频产生，  
分频系数可以设置为2、4、6、8。

由于ADCCLK允许的最大值是14MHz，  
因此当APB2总线时钟为 72M时，  
只能将ADC预分频系数设为6或8。

如果设为6，则ADCCLK时钟频率为12M：ADC  
的最小转换时间为： $14 / (12\text{MHz}) \approx 1.17\mu\text{s}$ ，最  
大转换速率 $\approx 0.86\text{MHz}$ 。ADC的最大转换时间为：  
 $252 / (12\text{MHz}) = 21\mu\text{s}$ ，最大转换速率 $\approx 0.14\text{MHz}$ 。  
当然，你也可以算出ADC预分频系数设为8时的  
最小和最大转换时间。



# 认识ADC

## 6、A/D转换原理

常用ADC电路有逐次比较型和双积分型等。

前者转换速度快，后者精度更高。

STM32F103ZET6ADC采用逐次比较型。

关于二者的具体转换原理，这里不再介绍。



# 认识STM32 A

## 1、ADC结构框图

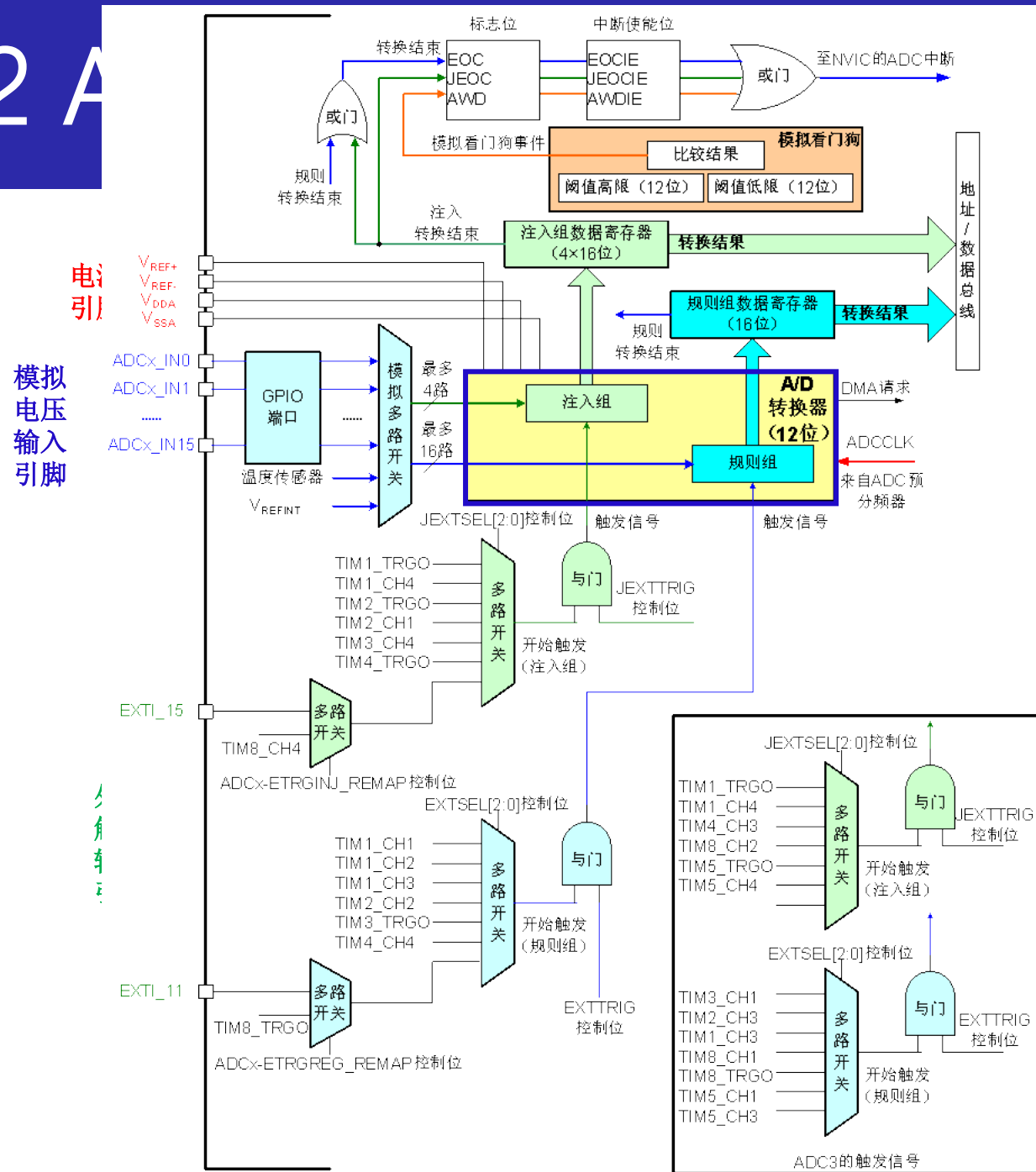
STM32F103ZET6内部有3个ADC，  
分别名为ADC1、ADC2、ADC3。

它们可以独立使用，

也可以双重使用以提高采样率。

图示为一路ADC结构。

核心部件是12位A/D转换器

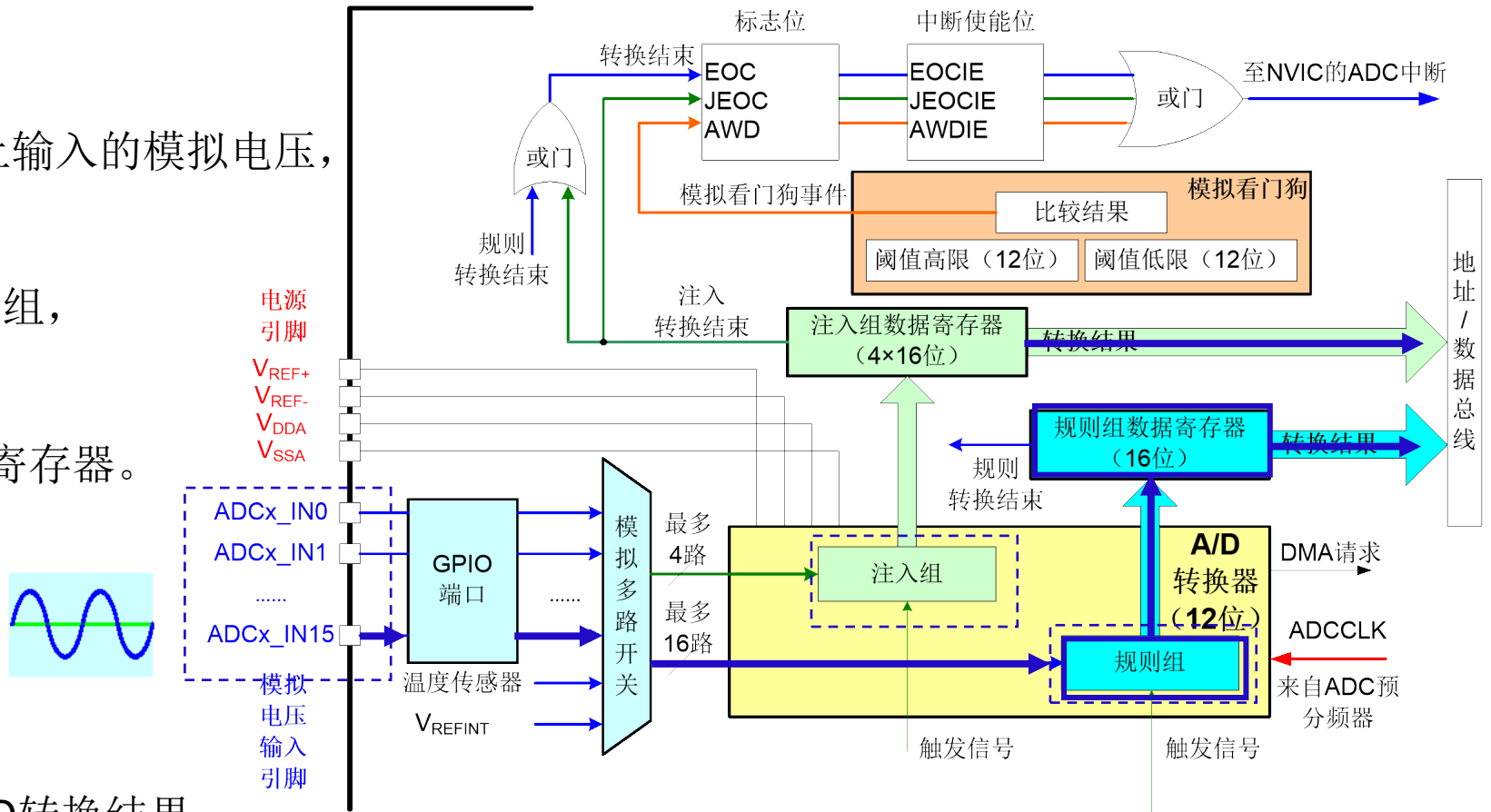


# 认识STM32 ADC的结构

## 2、ADC转换的过程

(1) ADCx\_IN0~ADCx\_IN15引脚上输入的模拟电压，经GPIO和模拟多路开关被送入AD转换器的注入组或规则组，之后进行AD转换。转换后的结果被存入相应的数据寄存器。

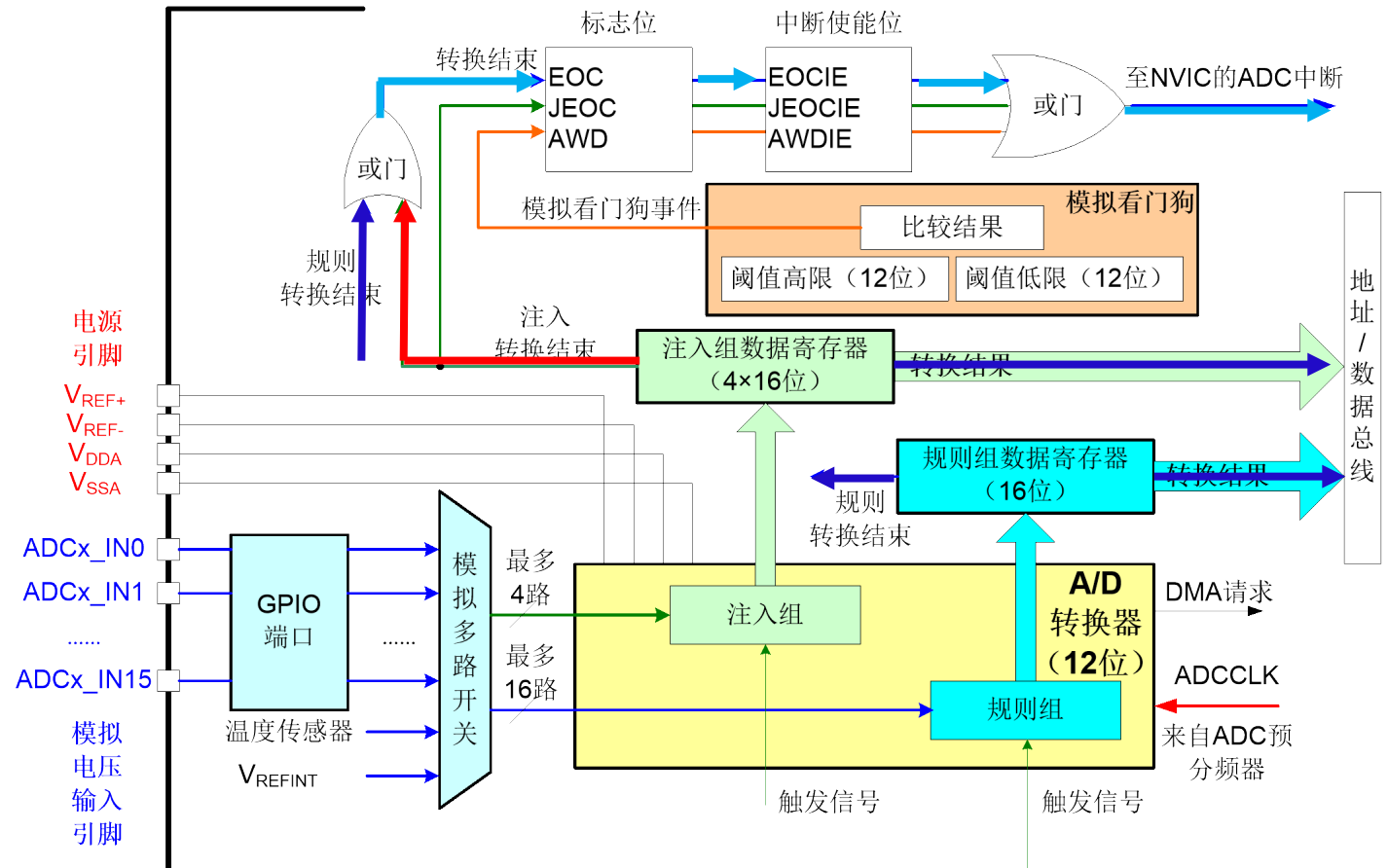
可以编程指定采样哪个引脚、送入哪个组，并从哪里读取数据寄存器中的A/D转换结果。



# 认识STM32 ADC的结构

## 2、ADC转换的过程

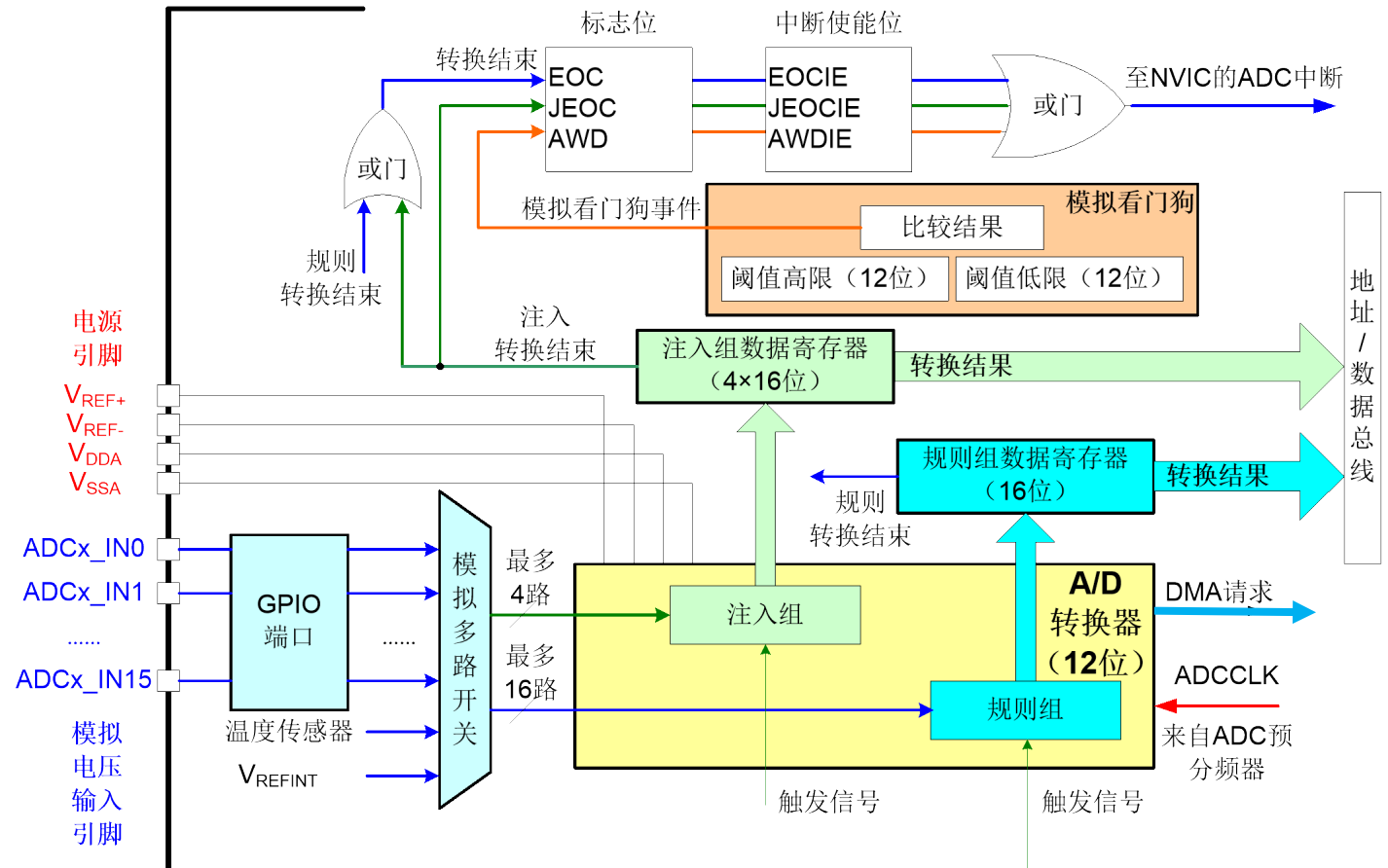
- (2) 无论是注入组还是规则组，在AD转换结束后，都会发出转换完成信号EOC。可以编程查询EOC，从而判断转换是否完成，然后读取转换结果。从而确保得到正确的转换结果。
- 如果允许了中断，则可以向NVIC发中断请求。
- 如果NVIC允许对该中断做出响应，则可以在中断服务程序中读取转换结果。



# 认识STM32 ADC的结构

## 2、ADC转换的过程

- (3) A/D转换结束后，除了产生中断外，还可以产生 DMA 请求，把转换好的数据直接存储在内存里面。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/045201032010012011>