

嵌入式系统原理及实验

[本页PDF](#)

嵌入式系统概论

嵌入式系统

- 嵌入式系统是一切非PC和大型机的计算机系统
- 嵌入式系统以**高性能、智能化**发展方向为主，且得益于**微电子技术、通信技术、感知测量技术**的发展
- **以应用为中心**
- **具有实时性、可靠性、功耗约束**

设计目标：从**高性能转向用户需求和资源约束的平衡**

MCU：微控制器单元，也叫做单片机，将内存、计数器等集成在一块芯片上，芯片级计算机

FPGA：现场可编辑门阵列，**半定制电路**，由逻辑功能块、输入输出块、布线资源组成

SoC：片上系统，系统级芯片

计算的概念：数值计算、通用信息处理、智能计算

ARM应用的两种开发方式

- 直接在ARM芯片上进行应用开发
- 在ARM上运行操作系统

计算机架构

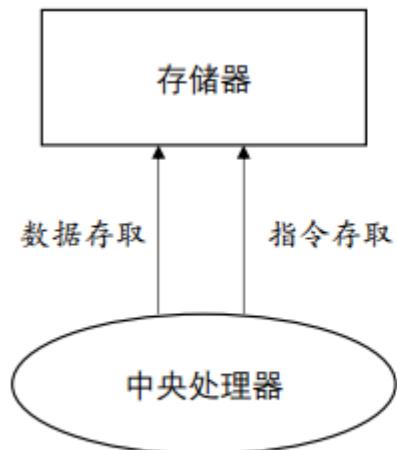
冯-诺伊曼架构

完整的计算机系统应该包含这五部分：存储器、运算器、控制器、输入设备、输出设备

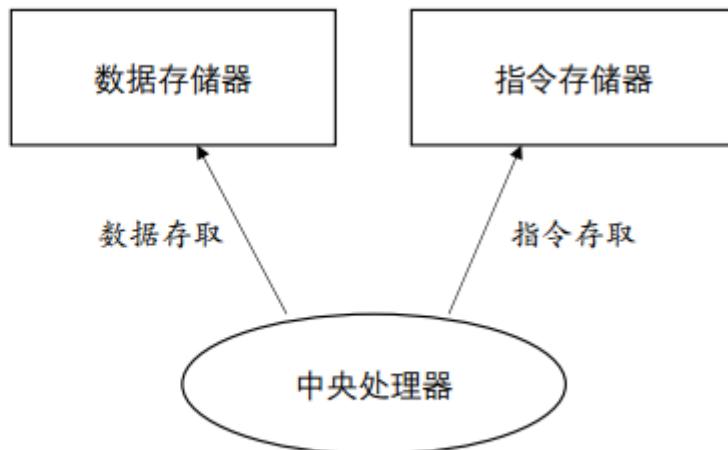
- 且由于**运算器**的数据吞吐能力有限，成为系统的瓶颈
- 后续以**存储器**为中心改进架构
- 内存储器放置正在运行的代码或数据，外存储器存放断电期的数据

哈佛架构

对**指令存储器**和**数据存储器**进行了区分，分别设置**指令总线（I-Code）** 和**数据总线（D-Code）** 进行存取



冯·诺伊曼架构



哈佛架构

指令集体系结构

简称为ISA

- 定义了一台计算机可以执行的所有指令的集合
- 定义处理器上的软件如何构建
- 复杂指令集：CISC，包含x86指令集
- 精简指令集：RISC，包括ARM指令集

RISC技术与CISC技术比较

| 指标 | CISC | RISC |
|-------|---------------------------------------|---|
| 价格 | 硬件结构复杂， 芯片成本高 | 硬件结构较简单， 芯片成本低 |
| 流水线 | 减少代码尺寸，增加指令的执行周期数，注重硬件执行指令的功能性 | 使用流水线 降低指令的执行周期数，增加代码密度 |
| 指令集 | 指令长度不固定 ，大量的混杂型指令集，有专用指令完成特殊功能 | 指令长度固定 ，简单的单周期指令，不常用的功能由多个简单指令组合完成 |
| 功耗与体积 | 含有丰富的电路单元，功能强， 体积大，功耗大 | 处理器结构简单， 体积小，功耗小 |
| 设计周期 | 长 | 短 |
| 应用范围 | 通用桌面机、高性能计算机 | 嵌入式领域、移动设备 |

计算机体系结构

指的是计算机的逻辑结构与功能特性

ARM的体系结构

- 主要包括微处理器所支持的指令集和基于该体系结构下微处理器的编程模型
- 对于开发人员来说，体系结构最重要的部分是指此微处理器提供的指令系统和寄存器组

Cortex-M3微处理器

- 由Cortex-M3内核和调试系统构成，哈佛架构，三级流水线
- 32位处理器，Thumb-2指令集

芯片级计算机=内核+外设

Cortex-M3内核的组成部分：

- Cortex-M3 core
- NVIC
- 系统时钟
- 存储器保护单元
- 总线矩阵
- 调试系统

流水线

主要步骤：取指、译码、执行

吞吐率(单位为指令/s, IPS)

$$\text{吞吐率} = \frac{\text{指令条数}}{\text{指令流水线所需时间}}$$

流水线执行n条指令所需时间

$$\text{所需时间} = \text{完成一条指令所需时间} + (n - 1) \times \text{时间最长的指令段}$$

加速比

完成一批任务不使用流水线时间和使用流水线所用时间之比

总线

各个部件之间传送信息的公共通路，包括：

- 数据总线
- 地址总线
- 控制总线

AMBA：高级微控制器总线架构

比较有代表性的总线接口：

- APB：高级外设总线，支持一主多从，低成本、低功耗、结构简单
 - APB1：连接低速外设

- APB2：连接高速外设，频率更高
- AHB：高级高性能总线，低成本，低功耗，结构简单，支持多主多从

AHB-Lite协议

- 一主多从
- I-Code指令总线和D-Code数据总线使用的协议

总线结构

- I-Code指令总线：基于AHB-Lite
- D-Code数据总线：基于AHB-Lite
- 系统总线：基于AHB-Lite
- 外设总线

存储器

暂存指令、数据和位址

使用模型：堆栈，由一块连续内存和一栈顶指针组成

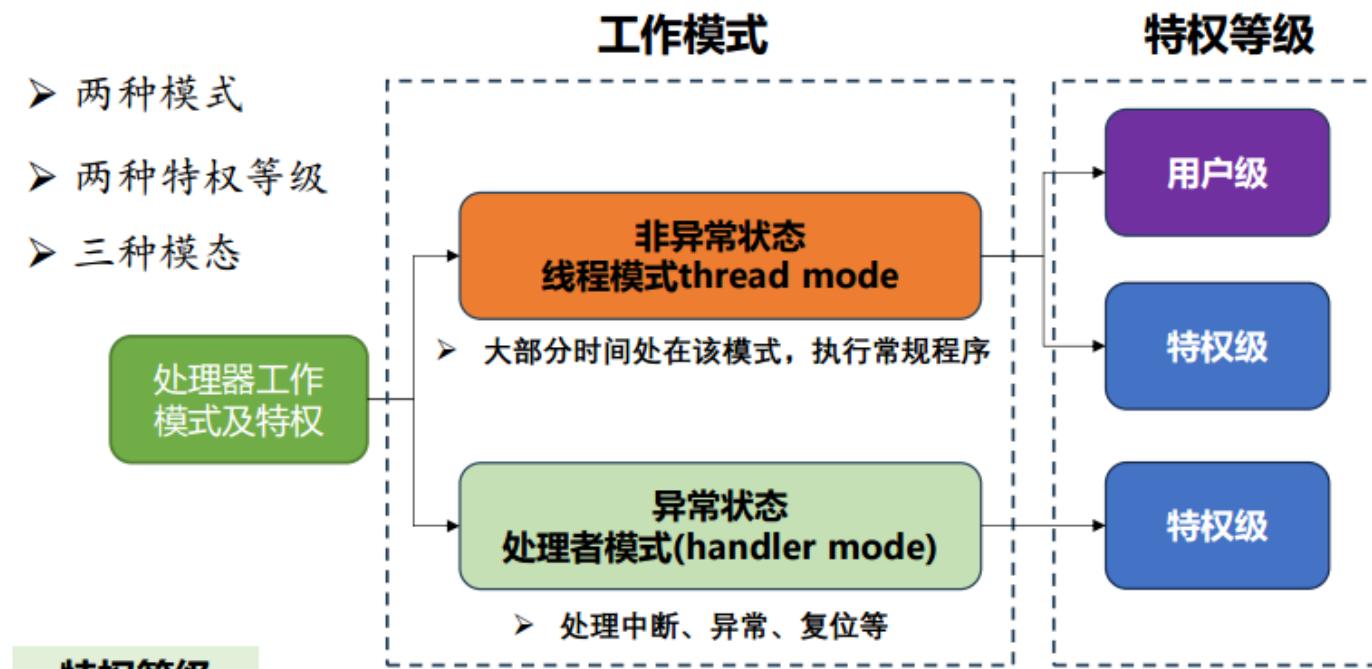
存储结构

Cortex-M3微处理器采用存储器与I/O设备（外设）**统一编址**

存储器映射：设置部分存储器地址范围用于外设的访问

工作模式

- 两种模式：线程模式（非异常状态）、处理器模式（异常状态）
- 两种特权等级：用户级、特权级
- 三种模态



中断

中断响应过程：保存现场、取向量、更新寄存器

中断悬起

中断发生时，正在处理同级或高优先级异常，或者被掩蔽，中断不能立即得到响应

中断请求信号保持

中断源持续不断地产生中断请求，CPU不返回主程序，立即再次进入同一中断程序

中断活跃

中断服务例程开始执行，悬起位被硬件清除

调试访问端口：DAP

两种调试主机接口

- JTAG
- 串行线（SWD）接口

高级跟踪接口：ATB

STM32最小系统

最小系统：用最少的元件组成微控制器可以工作的系统

STM32最小系统组成部分：微控制器、电源电路、时钟电路、复位电路、调试和下载电路

- 复位电路：有上电复位和手动复位
- 时钟电路：产生脉冲信号，晶振包括有源晶振（振荡器）和无源晶振（晶体），晶振频率越大，单片机运行速度越快，功耗越大

CoreSight调试架构包括调试接口协议、调试总线协议、对调试组件的控制、安全特性、跟踪接口

STM32开发

开发环境：

- 嵌入式开发环境
- 集成开发环境

STM32微控制器开发模式：

- 寄存器开发方式
- 标准外设库开发方式
- HAL库开发方式

两个调试接口：

- JTAG
- SWD

Cortex-M3内核有三种启动方式，由BOOT0和BOOT1的电平组合进行选择

针对ARM内核的主流集成开发环境主要有**Keil MDK**和**IAR EWARM**

嵌入式C语言

采用模块化、层次化的设计思想

`const`：定义全局、只读、必须初始化的变量

`static`：定义静态变量，只在定义该变量的源文件内有效

`volatile`：不让编译器优化，从内存中读写数据，不在寄存器读写数据

`extern`：声明此函数或变量定义在别的文件中

`typedef`：定义别名，编译阶段生效

```
typedef /*变量名*/ /*别名*/
```

`define`：宏定义，预处理阶段（编译之前）生效

```
#define /*宏名*/ /*替换文本*/
```

条件编译

- `#undef`：撤销定义的宏名
- `#ifdef`：判断某个宏是否未被定义
- `#elif`：类似 `else if`
- `else`：就是 `else` 的作用
- `#endif`：条件编译结束命令

`union` 共用体：允许在相同的内存位置存储不同的数据类型

- 占用的内存应足够存储共用体中内存最大的成员
- 使用运算符 `.` 访问共用体成员

回调函数：系统函数调用用户定义的函数，被系统调用的函数为回调函数

`volatile` 三个应用场景：

- 多任务环境下各任务间共享的标志
- 存储器映射的硬件寄存器
- 中断服务程序中修改的供其他程序检测的变量

指针占4个字节

GPIO

通用输入输出，是微控制器（MCU）上的一种通用引脚，实现与外部设备的数字信号交互

STM32F103一共有7组GPIO，每一组的编号都为0-15，每组GPIO端口都由7个寄存器组成，负责控制该端口的16个引脚

GPIO端口由五个主要部分构成：

- 引脚缓冲电路

- 方向控制
- 输出驱动
- 输入控制
- 复用功能切换

八种工作模式

输出模式

推挽输出（PP）：两个MOS管受互补信号的控制，即一个导通另一个则截止。输出高电平P-MOS导通，输出低电平N-MOS导通

- 目的：增大输出电流，增加输出引脚的驱动能力，提高电路的负载能力和开关速度

开漏输出（OD）：只有下拉MOS管（N-MOS），漏极直接与I/O引脚相连，不与电源连接，**处于悬空状态**

- 功能：实现逻辑“与”，减少芯片内部驱动，只可**输出低电平**（外接上拉电阻可输出高电平）

复用（AF）推挽/开漏输出：一个引脚可以作为多个外设引脚使用，一个引脚某一时刻只能使用复用功能中的一个

输入模式

上拉输入：引脚内部有上拉电阻，默认输入高电平

下拉输入：引脚内部有下拉电阻，默认输入低电平

浮空输入：不接上拉、下拉电阻，通过施密特触发器输入I/O引脚的信号

模拟输入：施密特触发器关闭，不接上拉、下拉电阻，常用于A/D模拟

中断

中断和异常：本质上是改变处理器执行指令的顺序

中断的处理流程：中断请求、中断响应、中断服务、中断返回

- 中断请求：**会置位中断请求寄存器**，向CPU发起中断

中断向量表：定义在启动文件，是中断服务程序的入口地址，**每个向量（函数地址）占4字节**

Cortex-M3集成了一个外设：**NVIC（嵌套向量中断控制器）**，专门用于处理中断

- 统一管理和配置中断
- 通过优先级来控制中断的嵌套和深度

STM32中的NVIC支持**16位优先级**（只有5种划分方式），因为只用到了Cortex-M3的8位优先级寄存器中的4位来配置中断优先级

Reset, NMI, Hard Fault的优先级为负，且不可修改，优先级高于普通的中断

中断的使用意义：

- 实时控制
- 故障处理
- 数据传输
- 不占用CPU

优先级

数值越小，优先级越高

抢占优先级 > 响应优先级 > 中断向量表顺序

外部中断/事件控制器 EXTI

STM32芯片之外的外设的中断由EXTI和NVIC共同负责

支持中断模式和事件模式：

- 中断模式：软件代码编写，需要CPU
- 事件模式：硬件触发执行，无需CPU

一共有19条外部中断线，其中0-15是对应IO引脚的外部中断

- 对应了每一个GPIO引脚都可以配置成一个外部中断触发源
- GPIO中断是以组为单位的，同组GPIO端口只能共用一条中断控制线
- 0-4有独立的中断服务函数，5-9共用一个中断服务函数，10-15公用一个中断服务函数

定时器

TIM2-7靠APB1控制，TIM1, 8靠APB2控制

功能：定时、计数、输入捕获、输出比较

分类：

- 基本定时器：TIM6、7，支持16位向上计数
- 通用定时器：TIM2、3、4、5，支持16位向上、向下计数
- 高级定时器：TIM1、8，支持16位向上/向下计数

基本定时器

预分频系数为1-65535，可产生中断，DAC触发信号

通用定时器

核心：16位自动装载计数器

组成部分：时钟源、时钟单元、捕获和比较通道

预分频器 (PSC)

取值范围为1-65535

对时钟源进行分频，供计数器计数

$$f_{CK_{CNT}} = \frac{f_{CK_{PSC}}}{(PSC + 1)}$$

本质也是一个分频计数器

计数器 (CNT)

取值范围为1-65535，可实现向上、向下、向上/向下双向计数

达到预定值会产生中断/DMA请求，属于软件中断

自动装载寄存器（ARR）

定义CNT计数器的终点，CNT达到后会进行复位

$$\text{定时时间} = (ARR + 1) \times \frac{PSC + 1}{f_{CK_{PSC}}}$$

高级定时器

可进行互补PWM输出，进行死区控制

PWM输出

PWM：脉冲宽度调制

$$\text{占空比} = \frac{\text{高电平时间}}{\text{周期时间}} \times 100 /$$

通用、高级定时器可以产生PWM输出

- 通用定时器可产生4路PWM输出
- 高级定时器可产生7路PWM输出

注意： - 频率由PSC（预分频器）决定 - 占空比由CCR（捕获/比较寄存器）决定

SysTick定时器

24位从重载值向下计数到0的计数器，是NVIC的一部分，由AHB **8分频得到**

功能：精确延时，任务切换

一共有4个寄存器，分别是：

- SysTick控制及状态寄存器CTRL
- SysTick重装载数值寄存器LOAD
- SysTick当前数值寄存器VAL
- SysTick校准数值寄存器CALIB

USART

USART：全双工通用同步/异步串行收发器 UART：全双工通用异步串行收发器

通信按数据传输格式区分：

- 并行通信：多位数据用多条数据线传输
- 串行通信：通过**单条或者两条数据线**传输数据

按同步方式区分：

- 同步通信：发送方和接受方有统一时钟线
- 异步通信：发送方和接受方无统一时钟线

按数据传输方向区分：

- 单工：数据只能单向传输
- 半双工：数据可以双向传输，但不能同时进行
- 全双工：数据可以同时进行双向传输

波特率：每秒传输的二进制位数，单位为bit/s, bps

字符速率：每秒传输的字符数

$$\text{波特率} = \text{字符速率} \times \text{字符位数}$$

UART模块

点对点通信

由TxD, RxD, GND三条线组成

- TxD：发送
- RxD：接收
- GND：信号地线

通讯协议：可分为**通信传输速率和数据帧格式**

异步串行通信协议

异步串行通信数据帧：

- 起始位：0表示传输数据的开始
- 数据位
- 校验位：用于检测数据是否有效
- 停止位：可以是1, 1.5, 2位
- 空闲位：1表示空闲

检验模式：

- 无检验：数据位直接跟停止位
- 偶检验：让数据位和校验位的1的出现次数为偶数，校验位根据数据位中1出现次数是奇数还是偶数来填0还是1
- 奇检验：让数据位和校验位的1的出现次数为奇数

RS-232是串行通信物理接口标准，为异步串行通信接口，电气标准为负逻辑，需要进行**电平转换**后和微控制器相连，微控制器标准为**TTL电平标准**

USART模块

组成部分：

- 数据收发寄存单元
- 波特率发生器
- 发送器/接收器控制单元

在UART基础上支持同步

编程模式：

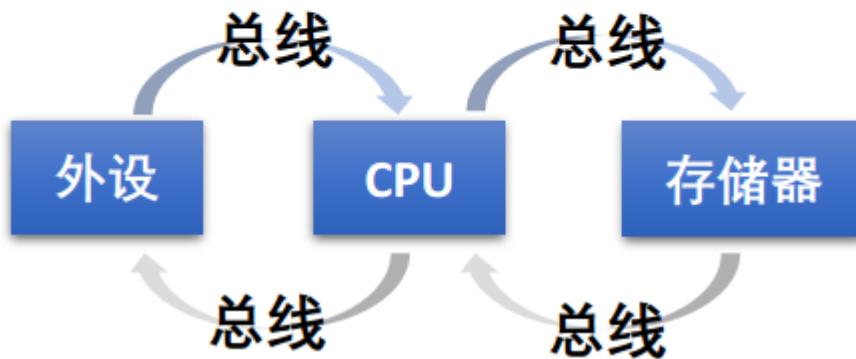
- 轮询

- 中断
- DMA（直接内存访问）

DMA

嵌入式数据处理过程：

嵌入式芯片中的数据处理过程



直接内存访问，允许外部设备和存储器进行直接的数据交换，无需CPU的介入，每次只传输一个字节，一个DMA同一时刻只有1个请求有效

工作流程：DMA请求、DMA响应、DMA传输、DMA结束

运行机制：通过专用DMA控制器（DMAC）接管总线控制权，实现外设与存储器、存储器不同区域之间的高速数据搬运

优势：

- 解放CPU
- 提高传输效率
- 降低系统功耗

应用场景：

- 存储设备I/O
- 高速通信通道
- 数据采集系统
- 图像处理
- 多处理器系统

DMAC的组成部分

- 地址寄存器
- 状态寄存器
- 控制寄存器
- 字节计数器

STM32F103包含2个DMA控制器：DMA1， DMA2

- DMA1有7个传输通道
 - 包括外设：TIMx ($x=1,2,3,4$)、ADC1、SPI1、IICx ($x=1,2$)、USARTx ($x=1,2,3$)
- DMA2有5个传输通道
 - 包括外设：(TIMx ($x=5,6,7,8$)、ADC3、SPI/I2S3、USART4、DAC通道1、2和SDIO)

DMA控制器提供2个AHB主端口

- AHB存储器端口
- AHB外设端口

DMA关键配置参数：

- 传输方向
- 数据密度
- 地址递增
- 传输模式
- 中断能力

DMA传输模式：

- 单次模式
- 循环模式

DMA使用仲裁器对多个DMA请求进行优先级管理

- 分为4个优先级：最高优先级、高优先级、中等优先级、低优先级
- 如果2个请求有相同的软件优先级，则较低编号的通道比较高编号的通道有较高的优先权

DMA中断事件，分别对应三个中断标志：HTIF、TCIF、TEIF

- HT：传输一半
- TC：传输完成
- TE：传输错误

SPI

串行外围设备接口

- 全双工
- 串行
- 同步
- 主从架构（一主多从）
- 使用单主设备通信协议
- 主设备：提供时钟
- 从设备：接收时钟

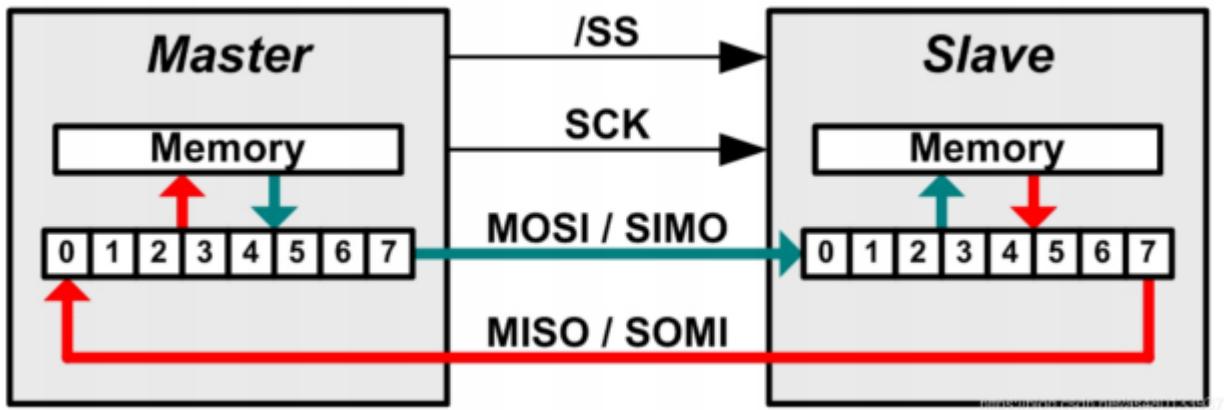
SPI接口使用四条信号线通信：

- MISO：串行输入数据线，主设备输入/从设备输出引脚
- MOSI：串行输出数据线，主设备输出/从设备输入引脚
- CS/SS：设备选择线，从设备片选信号，由主设备控制，有多个从机的话则不止一条，**低电平有效**
- SCLK：串行时钟信号，由主设备产生

工作原理

主机和从机都有一个串行移位寄存器

- 发送数据的时候，主机移位寄存器的最高位被推到MOSI，从机通过MOSI接收数据并放到自己移位寄存器的最低位
- 接收数据的时候，从机移位寄存器的最高位被推到MISO，主机通过MISO接收数据并放到自己移位寄存器的最低位
- 一个字节发送完后，主机和从机的移位寄存器的内容被交换



数据传输的时序模式

主从设备要调整成相同的时序模式！！！

CPOL (时钟极性)：用来配置SCLK的空闲电平状态

- 0为低电平
- 1为高电平

CPHA (时钟相位)：定义数据的采集时间

- 0代表在时钟的第一个跳变沿（上升或下降）采样数据，在第二个跳变沿发送数据
- 1代表在时钟的第二个跳变沿（上升或下降）采样数据，在第一个跳变沿发送数据
- 第一个跳变沿是从空闲到非空闲的边沿
- 第二个跳变沿是从非空闲到空闲的边沿

一共有4种标准的SPI模式，由CPOL和CPHA的两个位的0/1组成

I2C

串行、半双工、同步总线，是一种多主机总线，具备冲突检测和仲裁，抗干扰能力强

I2C总线由两根双向的信号线组成：

- SDA：用于收发数据
- SCL：用于通信双方时钟同步

每个连接到I2C总线上的器件都有一个唯一的地址（7bit），且每个器件都可以作为主机也可以作为从机（但同一时刻只能有一个主机）

具体阐述

通信步骤：
1. 主机发送起始信号启用总线
2. 主机发送一个字节数据指明从机地址和后续字节的传送方向
3. 被寻址的从机发送应答信号回应主机
4. 发送器发送一个字节数据
5. 接收器发送应答信号回应发送器
6.
（循环步骤4、5）
7. 通信完成后主机发

送停止信号释放总线

起始信号：SCL为高电平时，SDA由高变低

停止信号：SCL为高电平时，SDA由低变高

起始信号和停止信号都是由主机发出，从机只能被动应答

空闲时，SCL和SDA都是高电平

传输数据的最小单位是1个字节，主机发送起始信号后要发送1个字节的数据，该数据的高7位表示从机地址，最低位表示后续字节的传送方向

- 0：主机发送数据给从机
- 1：从机发送数据给主机

两种停止情况：

- 主机不想通信，发送停止信号
- 从机不想通信，不进行应答

I2C通信时一个数据帧一共有9位

- 发送时先发高位再发低位，每次发1个字节数据
- 发送方发送完数据后接收方要传回1位应答位回应发送方

应答位：0 非应答位：1

同步数据信号

- SCL为高电平时，数据线SDA要保持稳定，不允许发生变化，接收方从数据线上读取1位数据
- SCL为低电平时，数据线SDA可以发生变化，此时发送方向数据线发送1位数据

时钟同步

是通过I2C总线上SCL的线与功能实现的

- 如果有多个主机同时产生时钟，那么只有所有主机都发送高电平时，SCL上才表现为高电平
- 否则SCL都表现为低电平

仲裁

- 总线仲裁：当所有主机在SDA上都写1时，SDA的数据才是1，否则是0
- 主机每发送1bit数据，在**SCL为高电平时**，就检查SDA的电平和自己发送的数据是否一致，若不一致说明自己输掉了仲裁，然后停止向SDA写数据

典型时序

(1) 主机向从机发送数据

阴影：由主机向从机传送数据
无阴影：从机向主机发送数据。



(2) 从机向主机发送数据



(3) 主机先向从机发送数据，从机再向主机发送数据



S: 起始信号

A: 应答信号, 表示成功接收数据

P: 终止信号

Ā: 非应答信号, 表示接收失败或不再接收数据

ADC

模数转换器，将模拟信号转换为数字信号

主要过程包括：采样、保持、量化、编码

参考电压：进行模拟到数字转换时，ADC所能测量的最大模拟输入电压

香农采样定理

采样频率应该不小于模拟信号频谱中最高频率的2倍

常用的四种ADC输入模式

- 单端输入
- 差分输入
- 伪差分输入
- 全差分输入

常用的4种ADC

| 架构类型 | 特点 |
|---|----------------------|
| 逐次逼近型 SAR ADC | 中速、中高精度、低功耗 |
| $\Delta-\Sigma$ 型 Delta-Sigma ADC | 低速、 超高精度 、低功耗 |
| 流水线型 Pipelined ADC | 高速 、中高精度、中高功耗 |
| 闪速型 Flash ADC | 超高速 、低精度、高功耗 |

- $\Delta - \Sigma$ 使用了过采样和噪声整形技术

主要技术参数

分辨率

A/D转换器对输入模拟量微小变化的分辨能力，通常用二进制数的有效位表示

$$\text{分辨率} = \frac{\text{参考电压}}{2^{\text{有效位}}}$$

量化误差

有限分辨率而引起的误差，反映A/D转换器实际输出数字量和理想输出之间的差异

通常为1个或半个最小数字量的模拟变化量，表示为1LSB、1/2LSB

转换时间

指从转换控制信号到ADC输出端得到稳定的数字量所需要的时间

转换时间=采样保持时间+量化编码时间 = 1 / 转换速率

采样频率应小于等于ADC芯片的最高转换速率

STM32的ADC

ADC由五部分组成：

- ADC引脚
- 模拟多路开关

- ADC时钟
- 注入或规则通道
- 中断

集成有三个12位逐次逼近型的A/D转换器，ADC1和ADC3能产生DMA请求

转换结果是12位二进制数，可以左对齐/右对齐存储在16位的数据寄存器中

有18个通道，可以实现16个外部模拟输入通道和2个内部信号源的AD转换

各个通道的AD转换可以采用**单次、连续、扫描、间断模式**执行

- 单次：ADC只执行一次转换
- 连续：当前ADC转换结束立刻启动下一个转换
- 扫描：扫描一组模拟通道
 - 单次：扫描完本组后停止
 - 连续：扫描完本组后从第一个通道继续扫描
- 间断：用于多个通道的规则组和注入组

流程

- 信号输入
- 触发信号
 - 软件触发或EXTI外部触发或定时器触发
 - 硬件触发源分为规则通道和注入通道
- AD转换
- 数据保存

两种组转换模式：**规则组和注入组**

- 规则组：允许16个规则通道进行转换
 - EOC标志置位，产生中断或DMA
- 注入组：允许4个通道进行转换，4个注入通道寄存器存放转换结果
 - 无DMA，JEOC置位，产生相应中断

ADC触发信号产生方式：

- 软件触发
- 外部触发
 - 触发源：中断线、定时器

三种中断事件：

- EOC中断
- JEOC中断
- AWD中断

ADC总的转换时间 = 采样时间 + 12.5个周期，单位是周期

数据对齐：

- 左对齐：数据传输从高位开始
- 右对齐：数据传输从低位开始

校准的目的：减少因内部电容变化导致的误差

总结

:::tabs @tab:active 概论 嵌入式系统是什么 发展方向是什么 得益于什么的发展 以什么为中心 有什么特性

嵌入式系统的设计目标

计算的概念

MCU FPGA SoC的定义

ARM应用的两种开发方式

冯诺依曼架构的组成部分 系统的瓶颈是什么 为什么 后续如何改进的 内外存储器存放什么

哈佛架构对什么进行区分 设置了什么进行存取

ISA是什么 有什么用 有哪两种指令集

计算机体系架构是什么

ARM的体系架构 对于开发人员来说体系架构的重要部分是什么

@tab Cortex-M3微处理器 Cortex-M3微处理器由什么构成 是什么架构 什么流水线 多少位处理器 什么指令集

芯片级计算机由什么组成

Cortex-M3内核的组成部分

流水线的主要步骤

吞吐率怎么算

指令流水线执行时间怎么算

加速比怎么算

AMBA是什么 总线是什么 包括哪些总线

有代表性的总线接口和他们的特性 APB特点

Cortex-M3总线结构

APB1和APB2的区别

AHB-Lite协议的特点 什么总线使用了这个协议

存储器的作用 它的模型 它的组成部分

存储结构是什么 存储器映射是什么

有几种模式 分别是什么 几种特权等级 分别是什么 几种模态 分别是什么

中断的响应过程

中断悬起是什么 中断活跃是什么 中断请求信号保持是什么

DAP是什么 有哪两种调试主机端口

CoreSight调试架构包括什么

ATB是什么

@tab STM32最小系统

最小系统是什么

STM32最小系统的组成部分 复位有几种类型 晶振包括什么 晶振频率和单片机的关系

STM32有什么开发环境

STM32微控制器有什么开发模式

两个调试接口

Cortex-M3内核有几种启动方式 由谁决定

针对ARM内核的主流集成开发环境有哪些

@tab 嵌入式c语言 嵌入式开发采用什么设计思想

const static volatile extern typedef define有什么作用

define和typedef的区别

什么是回调函数

共用体是什么 如何访问共用体成员

volatile的三个应用场景

指针占几个字节

@tab GPIO

GPIO的全称是什么

STM32F103有几组GPIO，每一组的编号是多少到多少，每组GPIO端口由几个寄存器组成

GPIO端口由哪五个主要部分构成

GPIO由几种输入输出模式

推挽输出的特性，输出高电平谁导通，输出低电平呢，目的是什么

开漏输出的特性，功能是什么

复用输出的定义

上拉输入默认输入什么电平，下拉输入呢

浮空输入接电阻吗，会通过什么器件

模拟输入常用于什么情况，有通过什么器件吗

@tab 中断 中断和异常的本质

中断的处理过程

中断请求如何执行

中断向量表是什么 每个向量占几个字节

NVIC的名称是什么 作用

有哪些中断的优先级为负

STM32中的NVIC支持几位优先级，为什么

中断的使用意义

如何比较优先级

优先级的类型和他们的优先顺序

EXTI的名称是什么

外设的中断由谁共同负责

EXTI支持哪两种模式 具体有什么区别

EXTI有几条外部中断线 IO引脚的外部中断线有几条，编号是多少

GPIO中断是以什么为单位的

EXTI中断服务函数是怎么分配给IO引脚的

@tab 定时器 定时器的基本功能（四个）

基本定时器、通用定时器、高级定时器是TIM几，他们支持几位，如何计数

通用定时器的核心 通用定时器的组成部分

使用通用定时器计数的周期的公式

通用定时器实现计数功能的三个组件是什么 说出名字和缩写

计数器计数达到预定值会发生什么 产生的中断属于什么中断

高级定时器的功能

谁接在APB1 谁接在APB2

通用定时器和高级定时器可以有几路的PWM输出

什么是占空比

在PWM中，谁控制频率，谁控制占空比

CCR的中文名称是什么

SysTick定时器是几位的 向上还是向下计数 功能是什么 怎么得到的 四个寄存器分别是什么

@tab USART USART和UART的定义

可以有哪三种方式划分通信方式，他们分别可以分成什么类，每个类的定义

波特率是什么，单位是什么 字符速率是什么 波特率和字符速率的转换公式

通讯协议由什么组成

异步串行通信数据帧的组成部分

数据检验模式的分类和他们的定义

UART由哪三根线组成，它们定义分别是什么

USART模块的三个组成部分

波特率时钟频率和波特率的转换公式

USART的编程模式有哪三种

RS-232是什么，电气标准是什么，微控制器的标准是什么，它们俩之间是否要进行电平转换

@tab DMA

DMA的名称 DMAC的名称，作用

DMA的工作流程 优势 应用场景 运行机制

DMAC的组成部分

STM32F103包含几个DMA控制器 各自有几个通道

DMAC提供几个AHB主端口 分别是什么

DMA的关键配置参数

DMA的传输模式

DMA使用什么进行优先级管理 有几个优先级 优先级相同时怎么办

DMA有几个中断事件 分别是什么意思 对应什么中断标志

@tab SPI SPI的名称 特性 使用了什么协议

SPI使用几条信号线通信 分别是什么 有什么功能

SPI的工作原理是什么

有几种标准的SPI模式

主从设备的时序模式要相同吗

用什么来配置SCLK的空闲电平状态 具体是什么

用什么来定义数据的采集时间，具体是什么

@tab I2C I2C的特性 具备什么功能 是什么总线

I2C总线由什么组成 有几条信号线 分别是什么 有什么用

同一时刻只能有一个主机吗

从机的地址是几位的

通信步骤是什么

起始信号和停止信号是什么

空闲的时候SCL和SDA是什么电平

传输数据的最小的单位是什么 主机发送起始信号后发送的1个字节的数据的组成部分 最低位具体代表什么意思

有哪两种通信的停止情况

I2C通信时一个数据帧的组成情况

应答位和非应答位用什么电平表示

同步数据信号的做法

时钟同步是什么

仲裁是什么

@tab ADC

ADC是什么 主要过程有什么 参考电压是什么

香农采样定理是什么

常用的4种ADC输入模式

有哪四种常用的ADC $\Delta - \Sigma$ 用了什么技术

分辨率的计算公式 量化误差的最小单位

转换时间的计算公式 转换速率怎么算 采样频率和ADC芯片的最高转换速率的关系

ADC由哪五部分组成 STM32集成了几个什么类型的AD转换器 转换结果是几位二进制数 可以存在什么样的数据寄存器中

AD转换有什么模式 分别有什么功能

ADC的流程是什么

ADC有几种组转换模式 分别是什么 分别会产生什么样的中断

ADC触发信号产生的方法

ADC有几种中断事件 分别是什么

ADC总的转换时间怎么算

ADC有什么数据对齐的方式 分别对应什么

ADC校准的目的是什么