

嵌入式系统原理及实验

嵌入式系统概论

Cortex-M3内核

STM32最小系统

嵌入式C语言

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

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

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

通用定时器

预分频器（PSC）

计数器 (CNT)

自动装载寄存器 (ARR)

高级定时器

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

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

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：
- 数据位：
- 校验位：
- 停止位：

- 空闲位：1表示空闲

检验模式：

- 无检验：
- 偶检验：
- 奇检验：

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

USART模块

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

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

工作流程：DMA请求、DMA相应、DMA传输、DMA结束

优势：

- 解放CPU
- 提高传输效率
- 降低系统功耗

应用场景：

•

ADC

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
- 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

总结

:::tabs @tab:active 概论

@tab Cortex-M3内核

@tab STM32最小系统

@tab 嵌入式c语言 extern的作用

static的作用

volatile的作用

什么是回调函数

@tab GPIO

@tab 中断

@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

通用定时器的核心

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

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

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

@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