[toc]

嵌入式接口技术

硬件基础

- 电源
  - 三类

    > Type A (+5V)
    >
    > Type B (+3.3V)
    >
    > Type C (+1.8V)

    - 高电压抗干扰能力和驱动能力强
      - 0、1滑差大
    - 低电压功耗低

    > 根据使用场景选择种类
    >
    > - 通常是混合在一起用
    >
    >   - 需要解决电源稳定问题
    >
    >   > 如何解决：等待

  - 电源范围
    - $\pm 10 \text{%}$，部分 $\pm 5 \text{%}$

      > 宽电压范围芯片适用性好，可以简化电源设计

  - 相关概念
    - 电源纹波
      - 造成原因：信号反转
      - 如何解决：电源滤波(增加电容)
    - 掉电检测和数据撕裂
      - 如何解决：修改原始数据之前先建立备份

        > 需要考虑的问题
        >
        > - 备份区的使用寿命

    - 运行模式
      - 运行
        - 所有芯片都工作、功耗大

        - 待机
            - 让某一些部件工作，某一些部件不工作
            - 通过中断可回到运行
        - 掉电
            - 所有电路都不工作
            - 只能通过复位回到运行

- 复位
    - 上电复位

        > 基础是**电容充放电**
        >
        > 频繁充放电可能导致复位失效

        - 片内复位：芯片内部有复位电路
    - 片外复位
    - 上电延时时器

        > 避开上电以后电源不稳定期
    - 振荡器起振定时器

        > 确保振荡器起振以后信号稳定cpu再进行工作
        >
        > 例如计数1024个脉冲之后
    - 欠压复位
    - 外部复位电平及复位时长要求

        > **复位电平与端口电平不同**
        >
        > 复位电平：$\ge 80 \text{%}$ or $\le 12 \text{%}$
        >
        > 端口电平：$\ge 70 \text{%}$ or $\le 30 \text{%}$
        >
        > 复位时长以时钟个数来计算
        >
        > - 不是绝对时间
        > - 不同时钟频率可能不同
    - 复位标志
        - 作用

> - 区分冷启动和热启动
> - 对冷启动做数据单元初始化
> - 对热启动做数据单元还原
>   - 一般在出错情况下发生
>   - 如果找到正确备份，需要修复其他错误备份
>   - 备份不应该放在连续的物理地址空间
>   - 存储器容易出现连片错

> - 一个数据单元包括内容和校验码

- 上电复位：冷启动
- 看门狗复位
  - 外部复位
  - 欠压复位

> 复位后需要根据具体的情况进行复制操作

> - 部分寄存器不受任何复位影响
> - 部分在复位时变为确定初值
> - 部分寄存器不受 WDT 复位的影响
>
> 冷、热启动要注意区分
>
> 如果没有复位标志，可以利用 RAM 的特性建立标志区域

- 振荡器
  - CPU 工作最原始的信号源
  - 分类
    - 内部
      - 优点：便宜、体积小
      - 缺点：不准确
    - 外部
      - 种类

> - RC $$ T = RC = \cfrac{1}{2\pi f_c} $$
>   - 优点：**永远会工作**；成本低，不超过1角
>   - 缺点：及其不准
> - 陶瓷
> - 优点：精度高；相比于晶体更便宜，2角左右
> - 缺点：成品率低；启动困难
> - 晶体
>   - 缺点：成本高，6角左右
> - 优点：精度比陶瓷更高；成品率高；启动困难
> - EC

      - 双振荡器

        - RC振荡器保证起振
        - 陶瓷/晶体振荡器工作
    - 时钟占空比 $1:1$
        - 采用模 2 运算或者锁相环
    - 时钟分频与分配
        - 总线时钟
        - 部件时钟
    - 指令周期与时钟周期
        - 应用关心的是指令周期
    - 震荡器控制
        - 频率选择
        - 状态锁定位
        - 状态稳定位

            1. 选择振荡器
            2. 等待振荡器稳定
            3. 切换

- 端口
    - 信号上升时间：10% 上升到 90%
        - 上升时间往往更慢
    - 信号下降时间：90% 下降到 10%

        好信号：上升和下降时间 $\le$ 信号周期 $\times 4 \text{%}$

        如何解决不好信号：增大信号周期(降低时钟频率)；更改硬件
    - 端口复用
        - 为什么需要复用?

            端口资源珍贵

            - 端口需要有通用性
            - 引脚保护电路
                - 需要有足够大的面积散热
    - 端口配置
        - 易用性和灵活性 trade-off
        - **难点**

        - 配置复杂
    - 推挽
        - 优点：工作频率高
        - 缺点：容易坏

            > 集肤效应
            >
            > - 多股导线
            >   - 短路概率高

        - 通常在板内使用

            > 不容易形成短路

    - OD开漏
        - 开漏输出与推挽输出
        - 通常在板间使用

- 功耗控制

    > $$ P = K \times f^2 $$
    >
    > K 对于固定硬件可看成一个常数

    - 分总线和部件控制时钟

        > 关闭不必要的时钟，保留中断时钟
        >
        > 停止时钟还可以**规避干扰**

    - 低电压供电

- 可靠性
    - 看门狗

        > - 实质上是一个定时计数器
        > - 正常情况下可通过程序发出的清除指令清零
        > - 如果出现死机会造成溢出，导致复位
        > - 种类
        >   - 传统 WDT
        >   - 只有上限
        >   - 窗口 WDT
        >   - 有下限和上限

    - 软件 WDT

- 没事时休眠
- 内存可靠性高于端口

- 输入：周期性重复配置与滤波
- 输出：利用端口数据备份周期性刷新内容(先)与配置(后)

通信

UART

- 注意事项

> 1. 接受与死机
>
>   - 超时处理
>   - 重新初始化
>   - 接收缓冲区防止溢出
>
> 2. 停止位位数与可靠通信
>
>   - 停止位(又称为数据保护时间)的作用
>   - 给对方一定的处理数据的时间
>
> 3. 波特率精度与可靠通信
>
>   - 误差不超过 $\pm 2.5 \text{%}$
>   - 误差过大时需要换振荡器或者调整协议
>
> 4. 通信协议、帧格式及校验
>
> 5. 帧间隔、帧长度
>
>   - 处理时间
>
> 6. 波特率与通信距离
>
>   - 频率越高，通信距离越短

- RS-232-C的不足之处
  - 接口电平值较高，易损坏接口芯片
  - 传输速率低
  - 容易产生共模干扰
  - 传输距离有限

- RS-485
  - 采用差分传输
    - 逻辑1以两线的电压差为2~6V表示
    - 逻辑0以-2~-6V表示
  - 保证帧间间隔大于一个字符的时间
    - 可做极性识别
      - 0的个数不超过9(1个起始位+8个数据位，停止位为高)
  - **应该注意的事项**
    - 极性

        - **最远端**的匹配电阻(各110Ω)
        - T型线的长度 $<$ 3.4m
            - 如果超过需要调整主干线
        - 线路保护
            - 利用气体放电管 + TVS 半导体放电管 + PPTC(正温度系数热敏电阻)
                - 气体放电管
                    - 解决大功率问题
                - TVS + PPTC
                    - 解决快速短路

I2C

- UART的缺点
    - 通信双方需要精确的时钟
- I2C的优点
    - 简单、有效
    - 占用空间小，降低互联成本
- 总线
    - 由数据线 SDA 和时钟 SCL 组成
    - 最高传送速率 100kbs
    - 从设备均并联在这条总线上，每个设备都有唯一的地址
    - I2C总线是多主系统，系统可以有多个I2C节点设备组成
- 标准模式的扩展
    - 标准模式
        - 100kbit/s
        - 7位寻址
    - 快速模式
        - 400kbit/s
    - 高速模式
        - 3.4Mbit/s
        - 10位寻址
- 为了规避版权，有些硬件删掉了I2C的部分功能
- 注意事项
    - ACK与死机
    - 帧间隔与正常通信
    - 不是所有芯片都支持广播地址
    - 7位与10位地址
        - 有些器件不支持10位

    - SCL、SDA一定要开路输出
    - 时钟速率与通信距离
        - 满足应用需求，速率尽量低
            > 提高抗干扰能力
    - 结束与复位

SPI

- 4条线
    - 串行时钟线(SCK)
    - 主机输入/从机输出数据线(MISO)
    - 主机输出/从机输入数据线(MOSI)
    - 低电平有效的从机选择线CS(SS)
- 最高速率 5 Mb/s
- 注意事项
    - 对于不同的串行接口外围芯片，时钟时序不同
        - 根据外围芯片时序配置主机时序
    - 本身不支持纠错
        - 需要协议支持

1-WIRE

- 单线供电、通信
- 通信流程

> 1. 初始化
> 2. ROM 功能命令
> 3. PIO 功能命令
>   - 若只有一个设备可以通过 skip 命令跳过 2-3 步骤
> 4. 数据

接口按硬件分

- 基础元素

接口按部件分

- **==控制流程==**

> 传感器
>
> 执行器

> 人机界面
>
> 相互通信
>
> 电源

作业

- 自学 `Modbus` 通讯协议，写出提高串行通信可靠性的注意事项及分析

拓展阅读

- 波特率自适应
- 考虑系统是否会对人作出限制
    - 如有限制可能会遭到人为攻击