一、概念

1、全双工、半双工、单工

单工：只能一个人传输，只能向一个方向传输

半双工：只能一个人传输，可以多个方向传输

全双工：多方传输，多个方向传输

2、同步通信、一步通信

异步通信：双方时钟可以不同步，发送的信息封装（加上起始位、停止位）实现同步，效率低，用在低速传输中

同步通信：使用频率一致的时钟，数据帧通过独特的bit串作为启停标志，效率高，适合高速传输

3、波特率

数据传输速率使用波特率来表示，单位bps（bits per second）



二、

1. UART（异步、串、慢、全双工、2线、最远1200m）（Universal Asynchronous Receiver/Transmitter)

流程：设置波特率、设置数据格式、设置首发模式



①空闲位：

UART协议规定，当总线处于空闲状态时信号线的状态为‘1’即高电平

②起始位：

开始进行数据传输时发送方要先发出一个低电平’0’来表示传输字符的开始。因为空闲位一直是高电平所以开始第一次通讯时先发送一个明显区别于空闲状态的信号即为低电平。

③数据位：

起始位之后就是要传输的数据，数据可以是5，6，7，8，9位，构成一个字符，一般都是8位。先发送最低位最后发送最高位。

④奇偶校验位：

数据位传送完成后，要进行奇偶校验，校验位其实是调整个数，串口校验分几种方式：

1.无校验（no parity）

2.奇校验（odd parity）：如果数据位中’1’的数目是偶数，则校验位为’1’，如果’1’的数目是奇数，校验位为’0’。

3.偶校验（even parity）：如果数据为中’1’的数目是偶数，则校验位为’0’，如果为奇数，校验位为’1’。

4.mark parity：校验位始终为1

5.space parity：校验位始终为0

⑤停止位：

数据结束标志，可以是1位，1.5位，2位的高电平。

1. USART（同步、异步）(Universal Synchronous Asynchronous Receiver Transmitter)
2. IIC（半双工、串行，同步、两线SDA/SCL，传输慢，距离近，几百K）

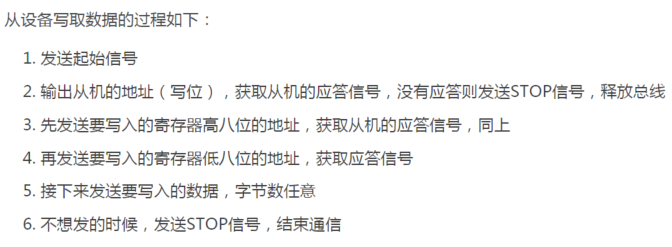
初始化：SDA和SCL都为高

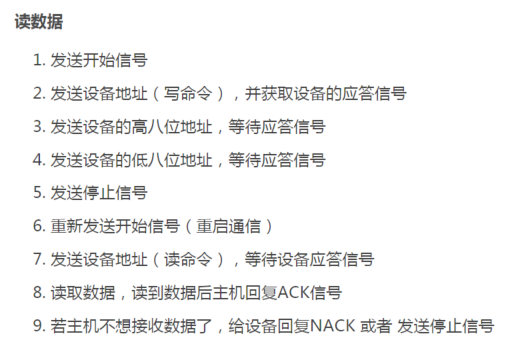
开始信号：SCL 为高电平时，SDA 由高电平向低电平跳变，开始传送数据。

结束信号：SCL 为高电平时，SDA 由低电平向高电平跳变，结束传送数据。

应答信号：接收数据的 IC 在接收到 8bit 数据后，向发送数据的 IC 发出特定的低电平脉冲，表示已收到数据。CPU 向受控单元发出一个信号后，等待受控单元发出一个应答信号，CPU 接收到应答信号后，根据实际情况作出是否继续传递信号的判断。

数据传输：SDA上的数据只能在SCL为低电平期间翻转变化，在SCL为高电平期间必须保持稳定，IIC设备只在SCL为高电平期间采集SDA数据。





IIC总线连接多个器件：

每个器件都有自己固有的地址，只要发相应的地址就可以与之通讯

器件的地址请查器件手册 也可以找找例程。

对于不同地址的模块就不用多说了，直接分别对其地址进行通信即可。那么若拿到相同地址的模块，或者直接是相同的多个模块怎么办

方法一：（内置了两种地址的模块）

对于内置了两种地址的模块，可以通过对某个引脚置高或置低来选择其中一个地址，现假设置高为A，置低为B。

假设你有三个模块要同时通信，首先将模块1置高，模块2、模块3的地址选择口置低，这样仅有模块1在地址A，然后对地址A进行通信即可防止其他模块干扰。接下来将模块1置低，模块2置高，即可对模块2通信。循环下去即可实现同时对三个模块通信。

方法二：（具有使能端的模块）

若你手中的模块某个引脚必须拉低或拉高才能正常使用，那么仿照方法一，对其中一个使能，其他均处于非使能状态，如此便可以仅对其中一个模块通信咯。

万能方法三：（什么都没有，仅有一种地址的模块）

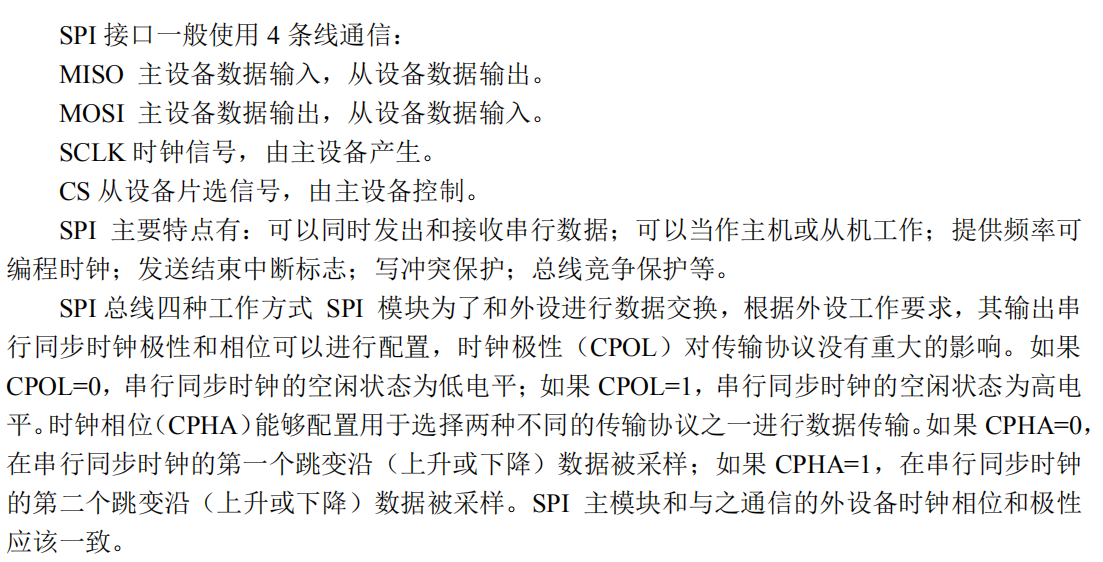
原本买来六个ADXL345来读取不同地方的加速度信息，datasheet里说可以通过对某个引脚置高置低来选择其中一个地址。然而发现并不能改变地址！于是只好想出这样一个邪恶的方法。

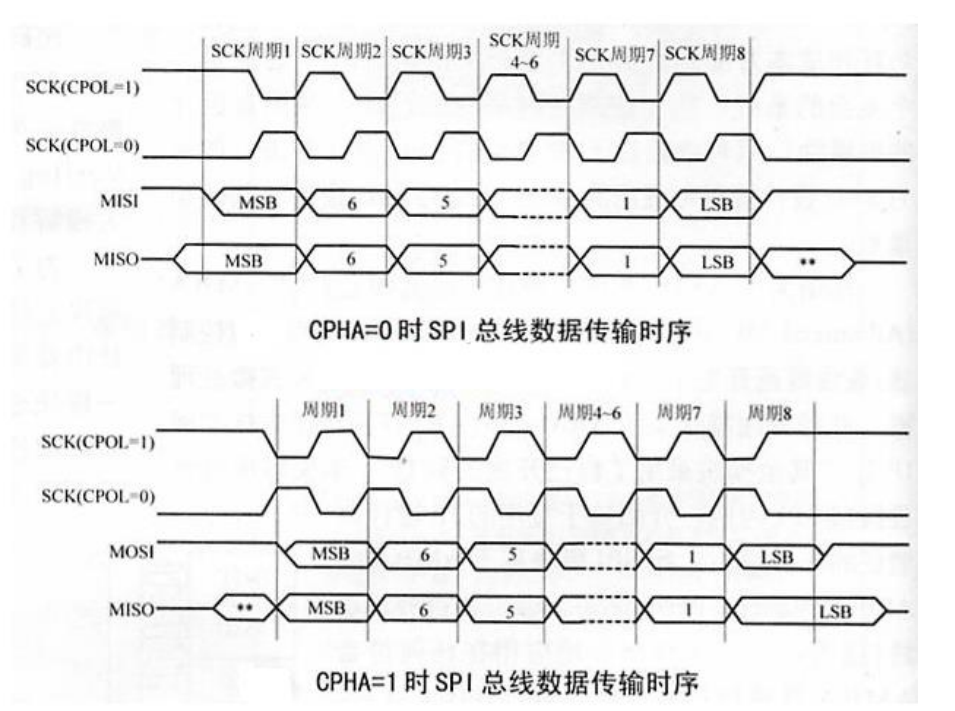
用到的工具是模拟开关，楼主以CD4053为例，这货非常便宜，几元钱可以买一大把。

简单来说模拟开关相当于多个单刀双掷开关，可以通过IO口输出高低电平控制某两路连通。

我们知道I2C通信有 SCL和SDA两根线，我们通过调整模拟开关使仅有一个模块完整接入I2C BUS即可实现只对其中一个通信。

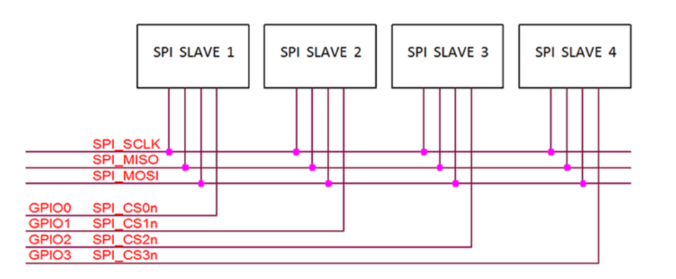
1. SPI（串、同步、全双工、传输快，3线或4线（SCLK/SIMO/SOMI/SS片选）、距离远）（Serial Peripheral interface串行外围设备接口）（主要应用EEPROM，FLASH，实时时钟，AD 转换器，还有数字信号处理器和数字信号解码器之间。）（几十M）



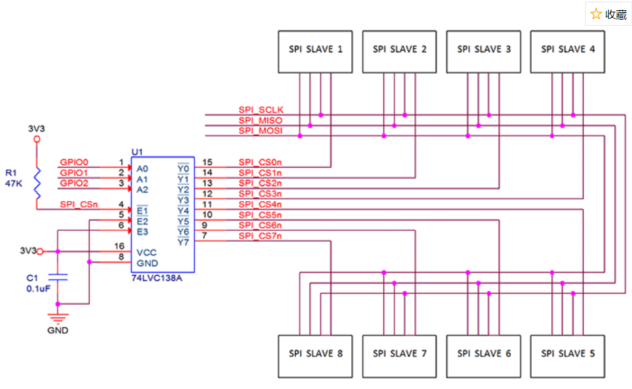


多路spi从设备连接方法

①GPIO模拟CS信号



②采用3 - 8译码器扩展8路SPI



1. USB（串、同步、半双工、4线（VBUS/GND/D+/D-）、传输速率快、传输距离近）

USB接口有4根线。两根电源线，两根信号线。USB接口的输出电压和电流是+5V 500mA。当数据线D+和D-的电压差大于200mV是表示输出为1，电压差小于200mV输出为0。在 USB 主机上，D-和 D+都是接了 15K 的电阻到低的，所以在没有

设备接入的时候，D+、D-均是低电平。而在 USB 设备中，如果是高速设备，则会在 D+上接一 个 1.5K 的电阻到 VCC，而如果是低速设备，则会在 D-上接一个 1.5K 的电阻到 VCC。这样当 设备接入主机的时候，主机就可以判断是否有设备接入，并能判断设备是高速设备还是低速设 备。

对于每个usb系统来说，都有一个称为host控制器的设备，该host控制器和一个根hub作为一个整体。这个根hub可以连接设备或者子hub，每个子hub又可以连接设备或者子hub。一条usb总线上最多可以接127个设备。

USB数据传逻辑结构

USB设备在逻辑上分成了几个层次，分别是设备层、配置层、接口层和节点层

参考博客：

https://blog.csdn.net/qq\_36016407/article/details/76718609?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522159780419619195162544505%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=159780419619195162544505&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_ecpm\_v3~pc\_rank\_v3-2-76718609.pc\_ecpm\_v3\_pc\_rank\_v3&utm\_term=USB总线&spm=1018.2118.3001.4187

1. RS232（串、异步、全双工，最大速率20k，传输距离15m，（不是呀传送控制信号）3线（tx，rx，d））常用9口接口、只能一对一

单片机TTL电平：3.3-5V对应1,0v表示0

rs232电平：+3V～+15V对应1，-3V～-15V对应0

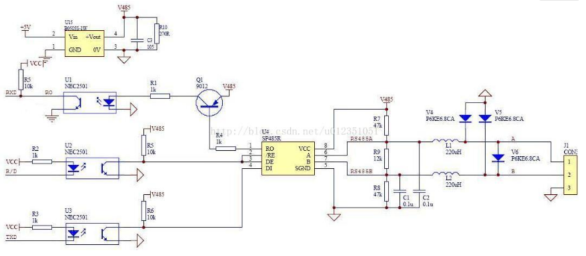
电平转换芯片Max232

Modem和调至解调器可以远距离传输

平时0Modem

1. RS485（差分、双绞线、32个节点、最远通讯1200m，支持全双、半双工，传输速度快最高10M），芯片MAX485

发送驱动器A、B之间的正电平在+2～+6V，是一个逻辑状态，负电平在-2～6V



8、can接口、canopen协议