**AVR通讯--同步异步区别**

我们都知道数据通讯就是两个通讯主体之间发送、接收数字信号。假设要发送以下数据：12，23，34，45，56，67，78，89。接收方要正确接收这些数据，就必须知道数据什么时候开始发送，什么时候结束，要不然，可能会将数据接收成22，33，44，55，66，77，88。

    对于同步通讯协议，发送方在发送数据之前先发出一个特殊的电信号，让接收方准备好接收数据，然后发送方就将以上数据全部连续发出，发送完毕后，再发送一个特殊的电信号表示数据发送结束。我们可以用以下图表示同步信号通讯。

|  |  |  |
| --- | --- | --- |
| 开始 | 1223344556677889 | 结束 |

                       数据包

     然后，接收方按照事先约定，即每两位一个数值，将数据包分成一个个数值。

    对于异步通讯协议，发送方每发送一个数据都要发一“开始”标志，每个数据发送结束后都发出一个“结束”标志。用下图表示异步通讯信号：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 开始 | 12 | 结束 | 开始 | 23 | 结束 | 开始 | 34 | 结束 |  |  |  |  | …… |

    因此，对于发送方发送数据的时间不确定（即发送一个数值，另一个数值可能几秒后才发送）的情况，适合使用异步通讯。但POS通讯是将数据（100-200个数值）一次性发出去，显然使用同步通讯比使用异步通讯效率高。

    因此，POS通讯中的同步、异步是两种不同的通讯协议，它是信号发送方和信号接收方的一种约定。通信过程中，同步信号的作用非常重要，尤其对于串行方式的数据传输。同步信号的作用简单讲有两个作用，1是“尺子”的作用，用于丈量一个比特的宽度；2是决定丈量的“起点”位置。收发双方必须采用相同的“尺子”，从相同的“起点”开始丈量数据线上的电平变化，才能保证数据传输的正确。   
  
           因此，任何方式的串行通信，同步的过程必须有，也就是说通信双方必须保持“同步”。   
  
          我们看最典型的SPI、I2C，在这两种方式中，都专门有一个CLK信号线，由通信的一方产生一个CLK，通过CLK信号线传到另一方，双方就就按这个CLK的控制工作，CLK的宽度就是一个数据位的宽度，而CLK的上升或下降沿，就是“起点”标志。这种在通信过程中明显有个CLK线专门传送同步信号的方式，就是同步通信。   
  
           同步通信由于有专用的CLK线控制，因此通信双方比较容易实现“同步”，因此速度比较快。但是对于长距离的通信，同步方式就不行了，1是由于需要专门一个信号线，成本提高。2是通信线越长，上面的干扰就越多，通信的速度也上不去。因此同步方式多是作为同一PCB板上芯片级之间的通信接口使用。   
  
          而长距离通信多使用“异步”通信方式，这里的“异步”不是指通信双方不需要同步，而是指通信双方之间不使用专用的同步信号线传送CLK，而是各自仅依赖于自己的系统时钟（这个就是异步的！），再根据约定的规程，调节自己的“步伐”达到双方的同步。   
  
          如果掌握这些最基本的概念，那么你能了解和面对通信中出现的问题，找到解决办法。比如对于UART的通信，你就知道为什么要规定双方要采用相同的帧结构，波特率了，如果设置错误会导致什么现象？而此时对双方的系统时钟不仅要求要更加准确和稳定，而且还要使用11.0592这些特殊的晶体，如果使用12M晶体，会产生什么后果等等。   
  
          AVR的USART可以工作在异步方式，也可以工作在同步方式，工作在同步方式就要多使用一个I/O口，作为CLK信号线。但AVR的USART的同步方式是个“鸡肋”，因为其同步传输方式工作时还采用异步数据传输的数据帧结构，即规定有1个起始位，8个数据位，1个停止位等。这样就使得AVR中的USART同步方式不能与一些简单的串行接口芯片（如164，165等）连接使用（这点没有51好）。   
  
         如果在板上只是要实现2个AVR之间的通信，到是可以考虑双方使用同步方式，这样就不需要采用特殊规格的晶体，而且可以使用最高的速度。

　数据通信可分为同步通信和异步通信两大类：

　　同步通信要求接收端时钟频率和发送端时钟频率一致。发送端发送连续的比特流。

　　异步通信时不要求接收端时钟和发送端时钟同步。发送端发送完一个字节后，可经过任意长的时间间隔再发送下一个字节。

　　异步通信的通信开销较大，但接收端可使用廉价的、具有一般精度的时钟来进行数据通信。

同步与异步传输的区别

1,异步传输是面向字符的传输，而同步传输是面向比特的传输。  
2,异步传输的单位是字符而同步传输的单位是桢。  
3,异步传输通过字符起止的开始和停止码抓住再同步的机会，而同步传输则是以数据中抽取同步信息。  
4,异步传输对时序的要求较低，同步传输往往通过特定的时钟线路协调时序。  
5,异步传输相对于同步传输效率较低。  
   
    同步传输方式中发送方和接收方的时钟是统一的、字符与字符间的传输是同步无间隔的。  
    异步传输方式并不要求发送方和接收方的时钟完全一样，字符与字符间的传输是异步的。

    在网络通信过程中，通信双方要交换数据，需要高度的协同工作。为了正确的解释信号，接收方必须确切地知道信号应当何时接收和处理，因此定时是至关重要的。在计算机网络中，定时的因素称为位同步。同步是要接收方按照发送方发送的每个位的起止时刻和速率来接收数据，否则会产生误差。通常可以采用同步或异步的传输方式对位进行同步处理。

　　 1. 异步传输（Asynchronous Transmission）：异步传输将比特分成小组进行传送，小组可以是8位的1个字符或更长。发送方可以在任何时刻发送这些比特组，而接收方从不知道它们会在什么时候到达。一个常见的例子是计算机键盘与主机的通信。按下一个字母键、数字键或特殊字符键，就发送一个8比特位的ASCII代码。键盘可以在任何时刻发送代码，这取决于用户的输入速度，内部的硬件必须能够在任何时刻接收一个键入的字符。

　　异步传输存在一个潜在的问题，即接收方并不知道数据会在什么时候到达。在它检测到数据并做出响应之前，第一个比特已经过去了。这就像有人出乎意料地从后面走上来跟你说话，而你没来得及反应过来，漏掉了最前面的几个词。因此，每次异步传输的信息都以一个起始位开头，它通知接收方数据已经到达了，这就给了接收方响应、接收和缓存数据比特的时间；在传输结束时，一个停止位表示该次传输信息的终止。按照惯例，空闲（没有传送数据）的线路实际携带着一个代表二进制1的信号，异步传输的开始位使信号变成0，其他的比特位使信号随传输的数据信息而变化。最后，停止位使信号重新变回1，该信号一直保持到下一个开始位到达。例如在键盘上数字“1”，按照8比特位的扩展ASCII编码，将发送“00110001”，同时需要在8比特位的前面加一个起始位，后面一个停止位。

　　异步传输的实现比较容易，由于每个信息都加上了“同步”信息，因此计时的漂移不会产生大的积累，但却产生了较多的开销。在上面的例子，每8个比特要多传送两个比特，总的传输负载就增加25%。对于数据传输量很小的低速设备来说问题不大，但对于那些数据传输量很大的高速设备来说，25%的负载增值就相当严重了。因此，异步传输常用于低速设备。  
如果往磁盘里写，异步是先写到Buffer，再写到目标磁盘，比如原写 1 2 3 4 5，而目标端只写了1。同步是不等收到没收到都会写到本地磁盘，容易造成Buffer阻塞。同步传输通常要比异步传输快速得多。接收方不必对每个字符进行开始和停止的操作。一旦检测到帧同步字符，它就在接下来的数据到达时接收它们。另外，同步传输的开销也比较少

　　 2. 同步传输（Synchronous Transmission）：同步传输的比特分组要大得多。它不是独立地发送每个字符，每个字符都有自己的开始位和停止位，而是把它们组合起来一起发送。我们将这些组合称为数据帧，或简称为帧。  
　　数据帧的第一部分包含一组同步字符，它是一个独特的比特组合，类似于前面提到的起始位，用于通知接收方一个帧已经到达，但它同时还能确保接收方的采样速度和比特的到达速度保持一致，使收发双方进入同步。  
　　帧的最后一部分是一个帧结束标记。与同步字符一样，它也是一个独特的比特串，类似于前面提到的停止位，用于表示在下一帧开始之前没有别的即将到达的数据了。  
　　同步传输通常要比异步传输快速得多。接收方不必对每个字符进行开始和停止的操作。一旦检测到帧同步字符，它就在接下来的数据到达时接收它们。另外，同步传输的开销也比较少。例如，一个典型的帧可能有500字节（即4000比特）的数据，其中可能只包含100比特的开销。这时，增加的比特位使传输的比特总数增加2.5%，这与异步传输中25 %的增值要小得多。随着数据帧中实际数据比特位的增加，开销比特所占的百分比将相应地减少。但是，数据比特位越长，缓存数据所需要的缓冲区也越大，这就限制了一个帧的大小。另外，帧越大，它占据传输媒体的连续时间也越长。在极端的情况下，这将导致其他用户等得太久。