串口通信

RS232信号是相对于地的电压信号，用在点对点传输。RS422&485信号是两根线，差分传输。RS422是RS232的扩展，目的也是用在点对点传输。RS485用于多点传输，一般是主/从结构组网。

# RS422 & RS485防护电路

如果应用环境恶劣，一定要在A/B线端外加TVS管、共模电感或接地等保护措施。推荐电路如下：



参看slla292a.pdf。保护电路如下：



Therefore, when selecting a TVS make sure its stand-off voltage is above or equal to the maximum bus voltage potential during normal operation, VWM ≥ VA, VB.

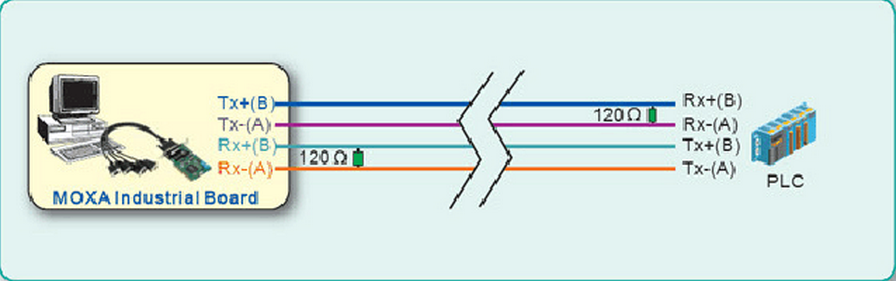
For example, to comply with the RS-485 specified common-mode voltage range of –7V to 12V, TVS stand-off voltages should be VWM ≥ 12V. Depending on the power rating of the TVS chosen, the maximum clamp voltages can range from 25V up to 35V, which is significantly higher than the maximum bus voltage of 14V of a standard transceiver. In this case, the internal protection circuit of the transceiver must absorb the remaining clamp energy to protect the device from damage.

布局布线如下：

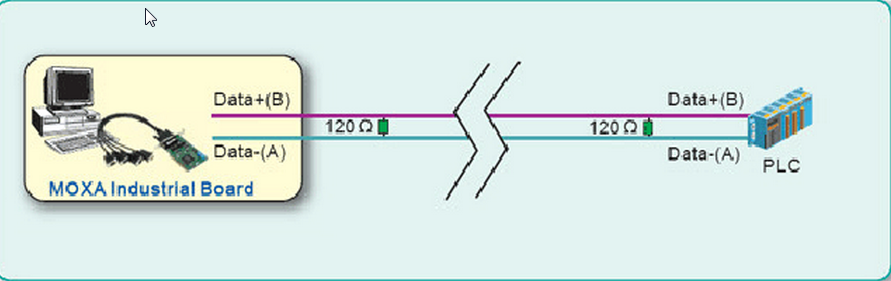


# 连接方式

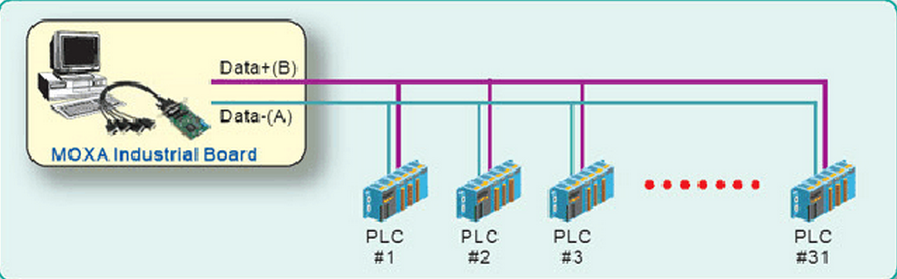
1.四线RS422接线图 全双工 点对点；



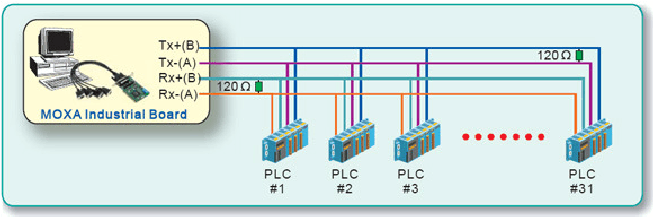
2. 二线RS485接线图 半双工 点对点；



3. RS485 半双工 点对多点



4. RS422 全双工 点对多点



# RS232通信协议

RS232协议是异步协议

Asynchronous communication

This interface uses an "asynchronous" protocol. That means that no clock signal is transmitted along the data. The receiver has to have a way to "time" itself to the incoming data bits.

In the case of RS-232, that's done this way:

1. Both side of the cable agree in advance on the communication parameters (speed, format...). That's done manually before communication starts.
2. The transmitter sends a "1" when and as long as the line is idle.
3. The transmitter sends a "start" (a "0") before each byte transmitted, so that the receiver can figure out that data is coming.
4. After the "start", data comes in the agreed speed and format, so the receiver can interpret it.
5. The transmitter sends a "stop" (a "1") after each data byte.

Let's see how looks the byte 0x55 when transmitted:

http://www.fpga4fun.com/images/SerialCommunication55.gif

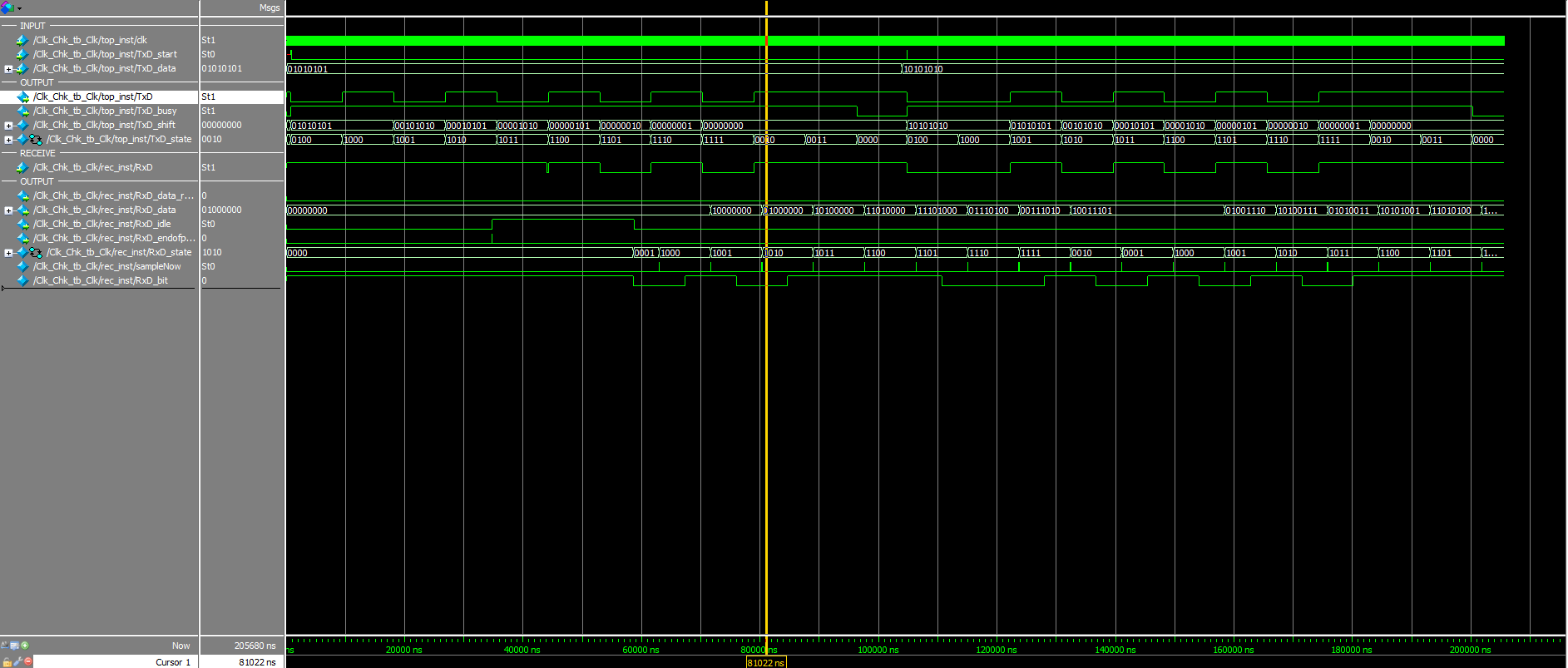
Byte 0x55 is 01010101 in binary.  
But since it is transmitted LSB (bit-0) first, the line toggles like that: 1-0-1-0-1-0-1-0.

Here's another example:

http://www.fpga4fun.com/images/SerialCommunication.gif

Here the data is 0xC4, can you see it?  
The bits transitions are harder to see. That illustrates how important it is for the receiver to know at which speed the data is sent.

如果接收端是在一个字节的中间接收数据，则会导致接收的数据混乱，如果两个字节中间没有停止位间隔，会导致RxD\_data\_ready无效。



当两个字节有足够的停止位间隔时，第二个字节可以正常接收，

