

# 基于 FPGA 的 RS485 通信接口实验手册

小梅哥编写，未经许可，不得用于任何其他用途

## RS485 介绍

RS485 是一个定义平衡数字多点系统中的驱动器和接收器的电气特性的标准，该标准由电信行业协会和电子工业联盟定义。使用该标准的数字通信网络能在远距离条件下以及电子噪声大的环境下有效传输信号。RS-485 使得廉价本地网络以及多支路通信链路的配置成为可能。

RS485 有两线制和四线制两种接线，四线制只能实现点对点的通信方式，现很少采用，现在多采用的是两线制接线方式，这种接线方式为总线式拓扑结构，在同一总线上最多可以挂接 32 个节点，如图 1 所示。

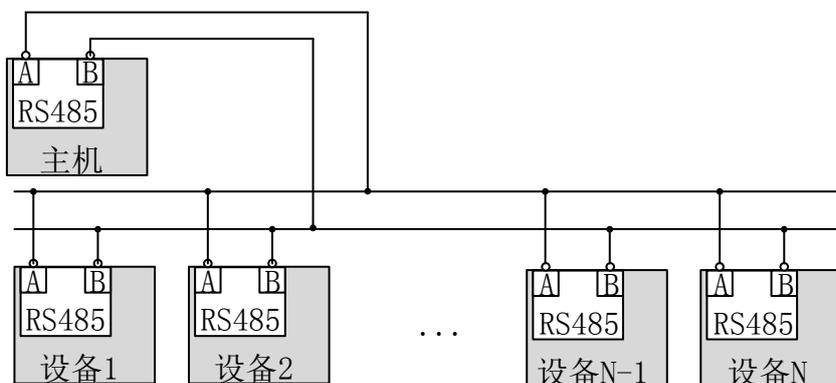


图 1 RS485 组网结构

在 RS485 通信网络中一般采用的是主从通信方式，即一个主机带多个从机。很多情况下，连接 RS-485 通信链路时只需要简单地用一对双绞线将各个接口的“A”、“B”端连接起来，忽略信号地的连接。这种连接方法能够在许多场合正常工作，但是在更加恶劣的工作环境中，还是应该保证所有设备都良好的接地，以保证通信的可靠性。

数字电路中，由 TTL 电子元器件组成电路使用的电平，电平是个电压范围，规定输出高电平 $>2.4V$ ，输出低电平 $<0.4V$ 。在室温下，一般输出高电平是  $3.5V$ ，输出低电平是  $0.2V$ 。最小输入高电平和低电平：输入高电平 $\geq 2.0V$ ，输入低电平 $\leq 0.8V$ ，噪声容限是  $0.4V$ 。

RS-485 的电气特性：逻辑“1”以 A、B 两线间的电压差为 $+ (2\sim 6) V$ 表示；逻辑“0”以两线间的电压差为 $- (2\sim 6) V$ 表示。接口信号电平比 RS-232-C 降低了，就不易损坏接口电路的芯片，且该电平与 TTL 电平兼容，可方便与 TTL 电路连接。虽然标准情况下要求压差为  $2\sim 6V$ ，但是实际上大多数的收发器芯片都将压差限制到了不低于  $200mV$  即可，这样能够确保数据不会在网络负载很大的时候因为压差降低而被遗漏。

在很多工业设备，尤其是电力系统的仪器仪表中，大多使用 RS485 接口实现设备和主控站的通信。RS485 接口属于电气接口，实现的是通信网络中的物理层。基于 RS485 物理层接口，使用 UART 控制器接收和发送按照标准串口协议编码的数据，即可实现设备间最基本的数据收发。然后，再使用常见的上层应用层协议，典型的例如 MODBUS 协议，完成多个设备的通信管理。该通信层级模型如图 2 所示：

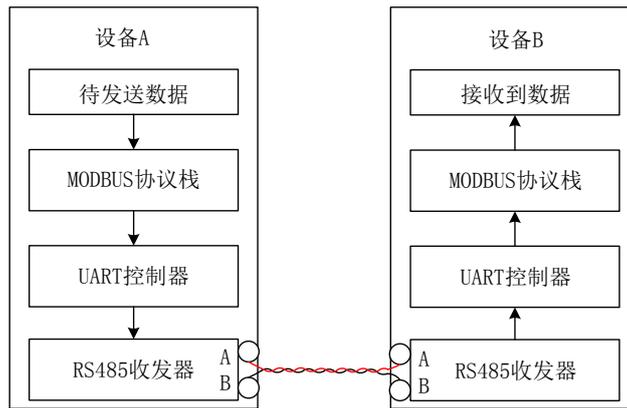


图 2 基于 MODBUS 的 RS485 通信数据流图

发送设备发送数据时，数据走向为：

1. MODBUS 协议栈将用户数据打包为一个 MODBUS 数据帧；
2. UART 驱动以字节为单位将 MODBUS 数据帧交付给 UART 控制器；
3. UART 控制器将字节数据完成并串转换，然后通过发送脚送到 RS485 收发器的驱动器输出（DI）端；
4. RS485 收发器将 DI 端每一个 bit 的数据转换为 RS485 标准的电平，通过双绞线传输到远端。

接收设备接收数据时，数据走向为：

1. RS485 收发器检测到双绞线上的 RS485 电平的每一个单 bit 数据，将其转换为普通的 TTL 电平信号，并通过接收器输出脚（RO）脚送给 UART 控制器的接收脚；
2. UART 控制器将接收脚上的每一个单 bit 信号按照 UART 协议进行接收解码，得到以字节为单位的实际数据。
3. MODBUS 协议栈将 UART 接收到的多个字节数据进行解析得到完整的数据帧，并解帧得到实际的数据部分

在实际应用时，MODBUS 协议栈一般由 CPU 通过软件方式实现。虽然使用 FPGA，以 HDL 的方式也能实现一些特定的功能，但是总体来说复杂度高，灵活性差，应用局限性大。如果确实希望在 FPGA 内部实现 MODBUS 协议，除了使用 HDL 编程以外，还可以使用更加灵活的 SOPC 技术，通过在 FPGA 内部嵌入 NIOS II 或其他软核 CPU 以软件编程的方式实现。

鉴于上述情况，本实验不对如何实现 MODBUS 协议栈做介绍，仅介绍如何完成基本的数据收发。

## RS485 收发电路介绍

上文提到，要实现基于 RS485 接口的数据收发，需要有 RS485 的物理层收发器。能够提供 RS485 收发器的厂家有很多，常见 RS485 收发器有 MAX485、SP3485（3.3V 供电）、ADI 公司磁隔离方式的 ADM2587 以及周立功单片机有限公司开发的 RSM485 收发模块等。在 AC620 开发板上，使用的是能够支持 3.3V 供电的 SP3485 芯片，该芯片电路简单，能够实现完整的 RS485 收发器功能。表 1 为该收发器的功能框图和引脚功能介绍。

表 1 SP3485 芯片引脚功能介绍

功能框图	No.	功能	功能描述
	1	RO	接收器数据输出脚，输出接收到的数据位
	2	RE	接收器使能脚，低电平有效
	3	DE	驱动器使能脚，高电平有效
	4	DI	驱动器数据输入脚，待发送的数据位由此输入
	5	GND	地
	6	A	未反转的发送器输出/接收器输入脚
	7	B	经过反转的发送器输出/接收器输入脚
	8	VCC	电源引脚

图 3 为 AC620 开发板上 RS485 收发器的电路图

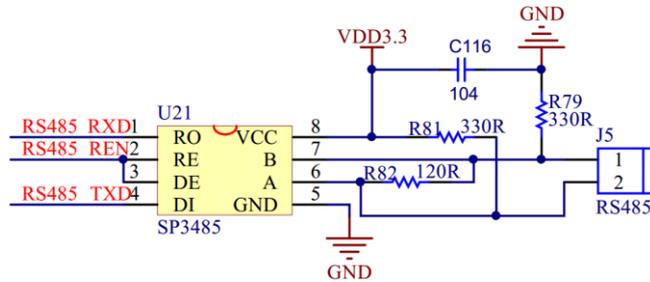


图 3 AC620 开发板 RS485 收发器电路图

可以看到，SP3485 的 RE 和 DE 两个引脚被连接到了一起，然后接到 FPGA 上，这是因为两者的有效状态本身就是相反的，RE 为低电平有效，DE 为高电平有效。因此，同一个信号控制这两个引脚，总能保证任意时刻接收和发送器中有且仅有一个被使能。

- 当 RS485\_REN 信号为低电平时，RE 有效，DE 无效，因此 SP3485 芯片中的接收器被使能，而发送驱动器被关闭，芯片处于接收状态；
- 当 RS485\_REN 为高电平时，RE 无效，DE 有效，因此 SP3485 芯片中的接收器被关闭，而发送驱动器被使能，芯片处于发送状态。

## FPGA 逻辑设计说明

本设计基于小梅哥的《FPGA 设计思想与验证方法》视频教程中“11 串口发送模块设计与验证”和“12 串口接收模块设计与验证”两个课程对应的随堂工程改编而来。使用时，在“11 串口发送模块与验证”随堂工程基础上添加“12 串口接收模块设计与验证”随堂工程中的“uart\_byte\_rx.v”文件，然后对顶层文件稍加修改实现。

修改后的工程框架如图 4 所示：

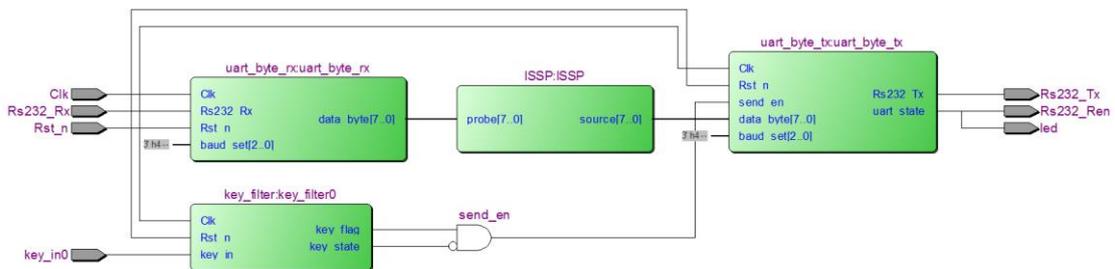


图 4 RS485 收发工程功能框图

可以看到，对于发送数据，还是和“11 串口发送模块与验证”视频中介绍的一样，通过按下按键产生发送使能信号，将 data\_byte 端口上的数据通过 Rs232\_Tx 发出。而 data\_byte 端口又连接到了 In-System Sources and Probes 工具的信号源端口（source）上，这样，通过在 In-System Sources and Probes 工具中给 source 设置不同的值，就能修改需要通过 uart\_byte\_tx 发送的具体数值了。

同时，为了保证发送时 SP3485 芯片中的发送驱动器是使能的，使用了 uart\_byte\_tx 模块中的 uart\_state 端口驱动 SP3485 的 RE 和 DE 信号。当 uart\_byte\_tx 模块处于数据发送过程中时，uart\_state 信号为高电平，此时 DE 使能，RE 无效，SP3485 芯片的发送驱动器被使能，DI 上的信号就能够通过发送驱动器变为 RS485 电平，经由 A、B 两个端口输出。实际上，小梅哥在早期讲解这个 uart\_byte\_tx 模块的设计时特意加上这个 uart\_state 端口，就是为了方便该模块用于 RS485 接口通信的。

当 uart\_byte\_tx 模块数据发送完成，处于空闲状态时，uart\_state 保持低电平，此时 SP3485 芯片中的接收器被使能，发送驱动器被关闭。所以，只要 RS485 总线上有任何电平变化，都能够进入 SP3485 的接收器，并转换为 TTL 电平，由 SP3485 的 RO 脚传递给 FPGA，再接入 uart\_byte\_rx 模块的 Rs232\_Rx 信号上，完成串行数据的接收。接收到的数据连接到了 In-System Sources and Probes 模块的探针端口（Probes）上，方便在 In-System Sources and Probes 工具中实时观察接收到的数据。

## RS485 实验验证

1. 验证时，有能力的请直接按照上文描述的内容，通过两个课程的随堂源码搭建对应的应用工程。不熟悉的也可以直接使用我们提供的工程源码，在 AC620 的“设计实例 --> AC620\_Verilog\_Example”里面能够找到，工程名为 [AC620\\_RS485\\_RX\\_TX.rar](#)。
2. 按照下述引脚分配表完成工程的引脚分配

信号名	信号方向	FPGA 管脚
Clk	Input	PIN_E1
Rs232_Ren	Output	PIN_D16
Rs232_Rx	Input	PIN_C16
Rs232_Tx	Output	PIN_D15
Rst_n	Input	PIN_E16
key_in0	Input	PIN_M16
led	Output	PIN_A2

3. 使用一个 USB 转 RS485 转换器，例如图 5 所示，USB 一端连接电脑，RS485 一端连接开发板上的 RS485 接口。该转换器一般使用的是 CH340 实现的 USB 转串口，因此插到电脑上会识别出一个 COM 端口，在电脑上使用时，就像使用串口功能一样。



图 5 USB 转 RS485 模块

笔者手里没有这个转换模块，但是有我们自己研发的多合一调试器模块，该调试器能够支持 USB 转 TTL、USB 转 RS232、USB 转 RS485 功能，同时还带有 USB Blaster 和 ST\_LinkV2 功能，使用很方便，因此笔者就使用该调试器的 USB 转 RS485 功能完成本实验的调试。

将该调试器的 RS485 接口连接到 AC620 开发板的 RS485 接口上，如图 6 所示。在电脑上选择对应的 COM 端口就可以实现 PC 和 AC620 开发板通过 RS485 接口通信了。

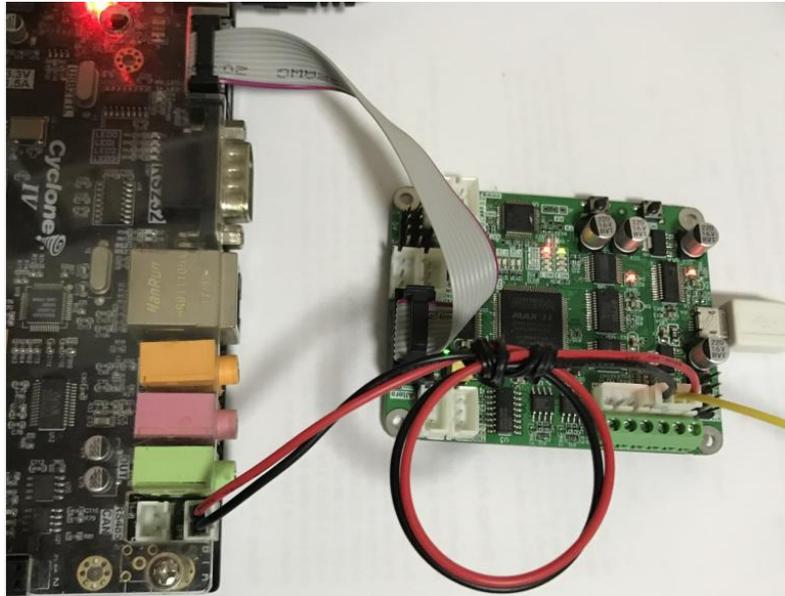


图 6 使用多合一调试器实现 RS485 调试

4. 打开串口调试软件（推荐使用“串口猎人”），选择 PC 上该 USB 转 RS485 模块对应的端口号，设置波特率为 115200。
5. 全编译工程，下载 sof 文件到开发板中。
6. 打开 In-System Sources and Probes 工具，将 source 和 probe 的显示格式都切换为 hex 格式，然后在 source 的 data 栏中输入希望发送的数据，然后按下开发板上的 S0 按键，可以看到，串口猎人上对应显示出了接收到的数据，如图 7 所示。

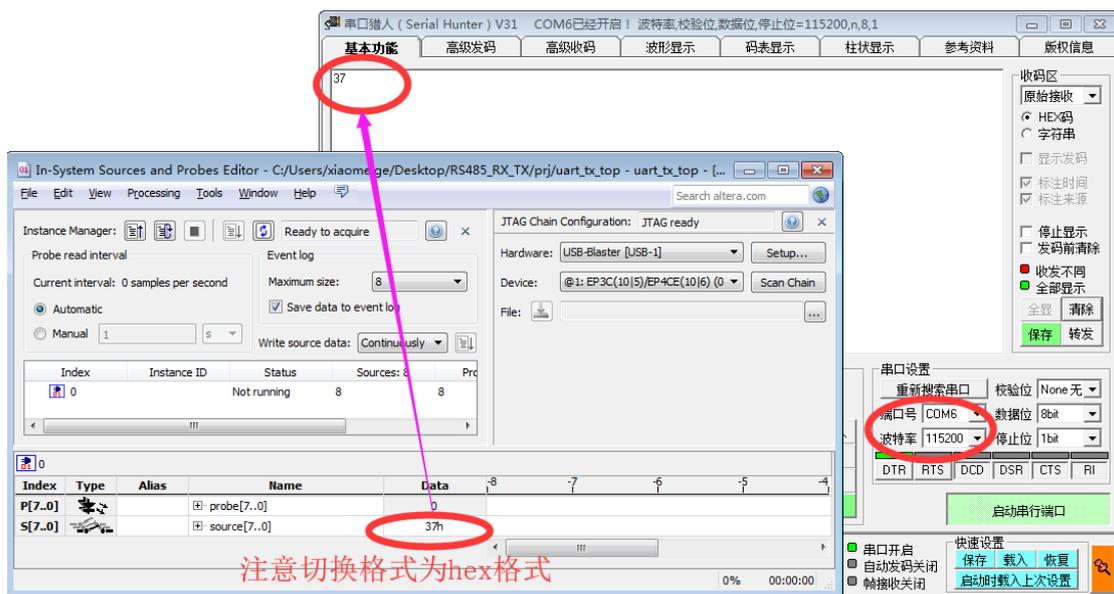


图 7 FPGA 发送数据

- 选中 In-System Sources and Probes 中的 probe，然后点击持续读取按钮，在串口猎人中输入一个字节的的数据并点击发送，则可以看到 In-System Sources and Probes 中的 probe 数据立即更新为了接收到的新数据，如图 8 所示。

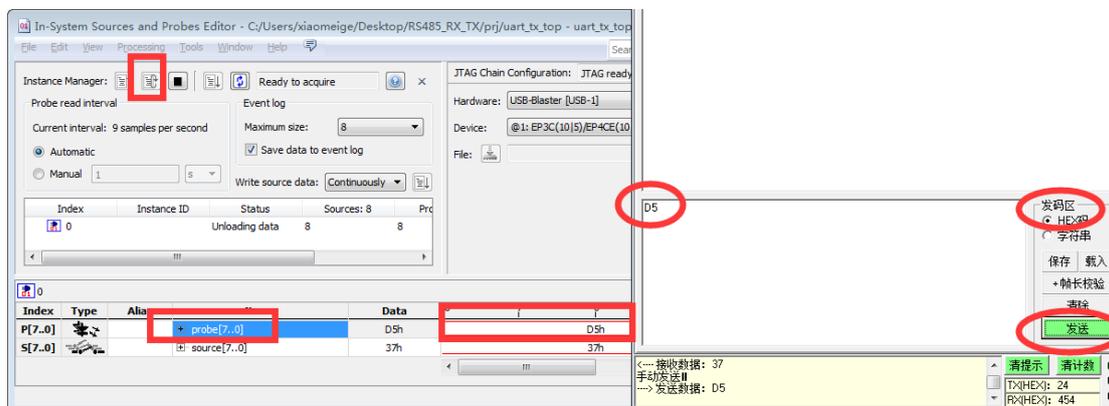


图 8 FPGA 接收 RS485 数据

使用示波器测量 RS485 总线上的 A 和 B 端，如图 9 所示。其中蓝色波形为 A，黄色波形为 B。可以看到，第一位波形，A-B 的差值为-1.24V 左右，代表 0，对应了 UART 串口协议的起始位，第二位，A-B 的差值为 1.24V，代表 1，而实际发送的数据 37h 的二进制值为 0011 0111b，bit0 的值为 1，刚好与波形对应。

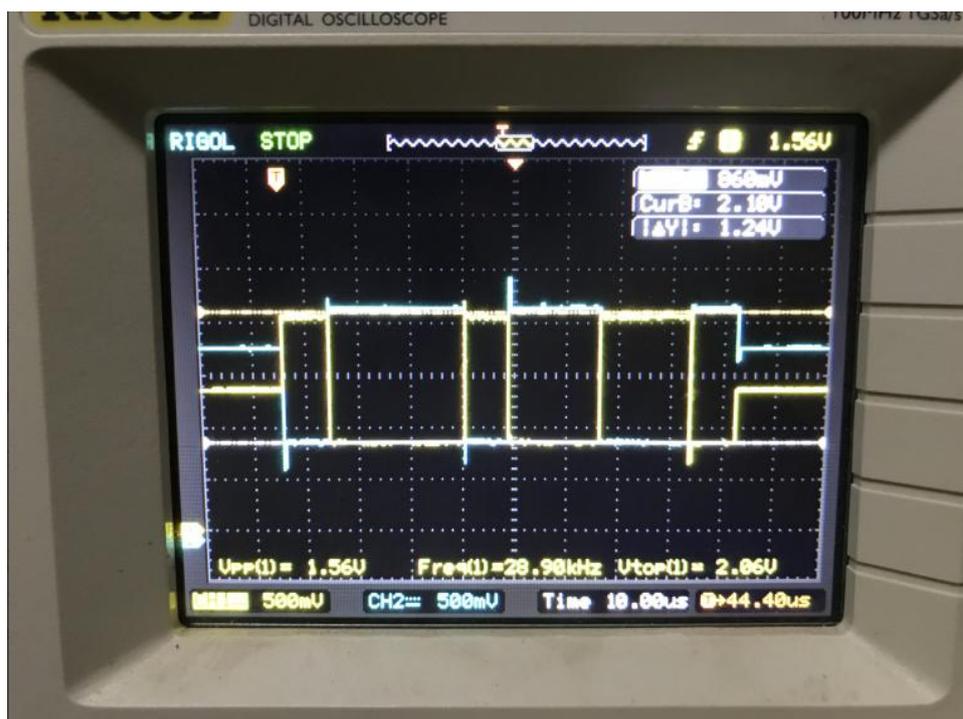


图 9 SP3485 传输数据波形图

## 总结

自此，基于 FPGA 的 RS485 基本通信就实现了，用户可以基于此工程，编写上层的应用逻辑，以实现更加复杂多样化的定制功能。当然，笔者更推荐使用 NIOS II 软核 CPU 实现这些功能。