

LonWorks 与 RS232/RS485 的接口设计

赵建民 林 勃 侯树春 (清华同方股份有限公司控制工程公司 100084)

摘 要 对 LonWorks 网络和串行 RS232/RS485 的接口设计作了介绍。**关键词** LonWorks RS232 RS485 网关

LonWorks 技术是比较流行的一种现场总线技术,它把相应协议(名称为“LonTalk 协议”)嵌入相应芯片(名为“Neuron”的芯片),用户采用该芯片及相关配件(收发器及需要的时候外加 RAM/ROM 等)设计出自己的各种应用节点,然后利用各节点和路由器/中继器等构造成 LonWorks 网络(网络支持总线式、星形式和自由拓扑式等各种网络拓扑结构)。

Neuron 芯片可配置为与各种通信介质接口(双绞线、电力线、光纤、无线、红外线等,其中双绞线在 78K bps 下直接通信距离可达 2700m),可在大的数据通信率范围内运行(4800bps ~ 1.25Mbps)。各 LonWorks 开发商/用户还组成 LonWork 协会,如果用户采用该标准所制定的规范而设计自己的 LonWorks 产品/网络,则十分容易与其他通过 LonMark 标准的产品作到无缝相连,但如果要与其他总线的系统相连,则需通过网关连接。

要实现 LonWorks 网络和其他系统的网关,一种方法是设计 LonWorks 网络和串行 RS232/RS485 的接口,通过它再和其他系统互连。目前有 3 种方法来实现 LonWorks 总线到 RS232/RS485 的接口:①购买 Echelon 公司的 PSG-10/PSG-2 等成型的设备或其他厂商的产品,在他们设计的 LonWorks 总线到 RS232/RS485 的接口硬件基础上,调用他们提供的相关发送/接收的库函数,从而开发出自己的针对专门系统的网关程序。②自己设计硬件,并且采用 Neuron 芯片的专用串行 RS232 接口。③自己设计硬件并开发库函数,但采用扩展的串行接口芯片而实现高速串行 RS232 接口。采用第一种方法,特点是实现时间短,不用作太多开发,但缺点是价格昂贵;采用第二种方法,即直接利用 Neuron 芯片的 IO8 和 IO10 及芯片的内嵌函数而实现它和 RS232 的接口,缺点是速度慢(最高仅为

9600bps),但由于开发简单,许多厂商推出这种串行接口;采用第三种方法,利用 16C550 串行接口芯片和 Neuron 芯片相连,既能通过开发过程熟悉 Neuron 芯片和其他控制芯片的接口,实现高达 128K bps 的通信速率,又能大大节约成本。在时间要求不是很紧的情况下,我们采用了第 3 种方法。

整个原理框架如图 1 所示,接口可以分为 3 大部分:(1)围绕 Neuron 芯片的 10MHz 晶振,FTT-10A 自由拓扑收发器,AT29C512 的 Flash ROM (里面有 48K bytes 的容量可供用户编写网关程序,足够编写大型的复杂程序),RAM6264 (8K bytes 的 RAM,在高速串行通信下,可用作大数据量的数据暂存区域)。(2)围绕 NS16C550 通用异步串行接口芯片的设计,它和 Neuron 芯片的接口设计是整个系统的关键,NS16C550 在 18.432MHz 下,可达 128K bps 的通信速率。(3)围绕 RS232/RS485 接口芯片 MAX202/MAX1480B 的设计,这些是和 16C550 的接口设计,RS232 采用 MAX202 芯片,RS485 采用带隔离的 MAX1480B 芯片,由此达到和远端

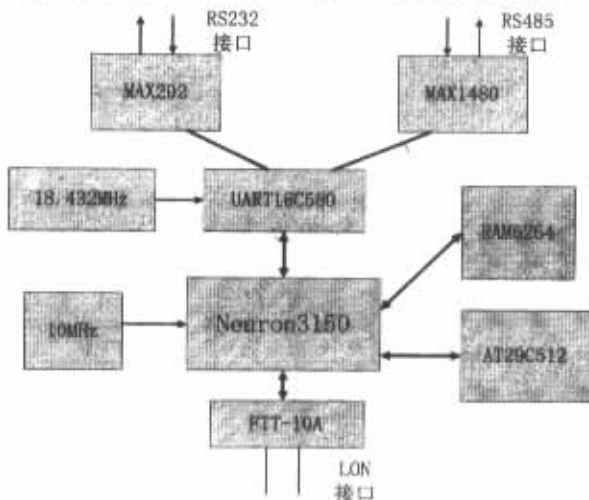


图 1 原理框架图

RS485 的隔离型连接, 增加系统的抗干扰性。图 2 给出 16C550 和 MAX1480 的接口图, 而整个电路的原理图, 由于涉及技术保密, 在此不宜给出。其中

IO1 接到 Neuron 芯片的 IO1 管脚, 控制 MAX1480 的接收/发送状态的切换, SIN/SOUT 连至 16C550 的相应引脚。

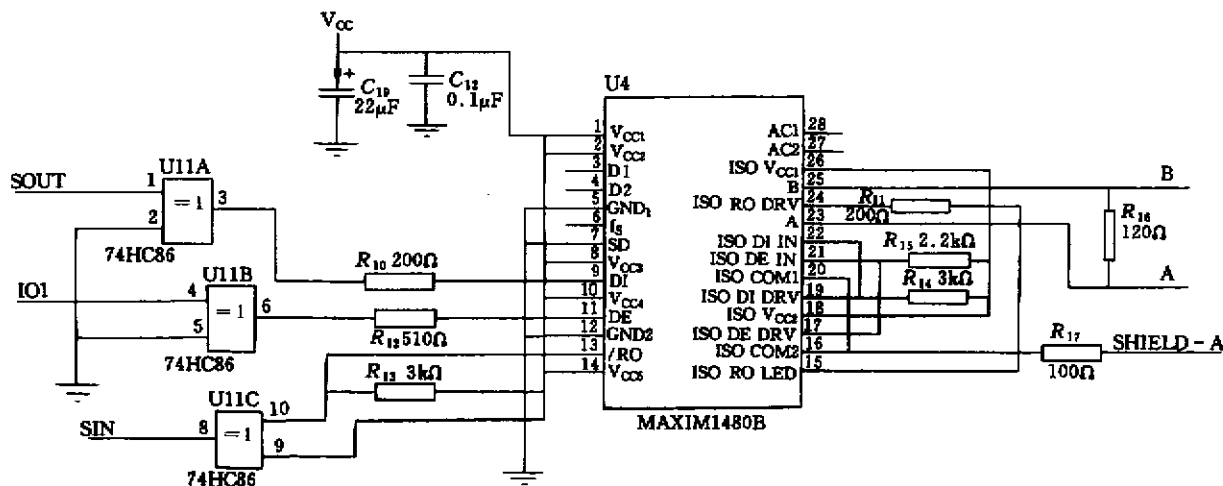


图 2

整个调试的重点, 集中在 Neuron 芯片和 16C550 的接口上。我们采用了 Neuron 的 Memory-mapped I/O 方式寻址 16C550。由于 Echelon 所给的相关资料比较缺乏, 使得在初期调试时遇到一点困难, 特别是 16C550 和 Neuron 芯片接口出错时, 很容易造成 Neuron 芯片出错而不能重新上电恢复; Neuron 芯片的 64-LED Quad Flat-Pack 贴片封装形式使得焊上后又难以卸下来, 为此我们专门做了一个转换插座, 可以随意更换 Neuron 芯片, 从而避免了上述问题。下面给出了相关 16C550 的初始化程序, 读者从中可以学习如何应用 Memory-mapped I/O 方式寻址 (其中 pDevice->controlreg0 指向 16C550 的基地址):

```
//initial
```

```
io_out(io_csl, 1); //16C550 片选信号
```

```
pDevice->controlreg3 = 0x80; //设置 LCR 寄存器 DLAB = 1, 以设置波特率
```

```
pDevice->controlreg0 = 9; //设置波特率为 128Kbps
```

```
pDevice->controlreg1 = 0x0;
```

```
pDevice->controlreg3 = 0x3; //数据格式: 8 bits, 1 stop bit, 无校验
```

```
pDevice->controlreg2 = 0x87; //设置 FIFO
```

```
pDevice->controlreg1 = 0x0; //关闭所有中断
```

调试调圆后, 与 RS232/RS485 的通信速率达到 128K bps, 成功地达到了预期的目的, 并且编写了查询/中断方式接收字符、发送字符/字符串的库函数, 为以后编写与专门系统的网关程序打下了基础, 并且通过开发, 丰富了 Neuron 芯片的应用。