实用隔离型 RS-485 通信接口的设计

A Method of Designing a Practical Isolated RS-485 Communication Interface

(西安)西北工业大学 (710072) 李玉忍 谢利理

【摘要】运用 MAX1480B 芯片设计完成了实用隔离型 RS-485 通信接口,方便地完成了 RS-232 与 RS-485 的转换,其性能稳定可靠。同时介绍了通信软件设计要点及编程方法。

关键词:隔离,网络,通信,半双工

Abstract: A method of designing a practical isolated RS-485 communication interface with MAX1480B is discussed. The converting electric circuit of RS-232 to RS-485 is used in multi-computer communication real-time monitoring system. The result deriving from the field debugging indicates that the system communication is reliable and has good performance. In the end, the design and programming of the communication software are given.

Key words: isolation, network, communication, semi-duplex

计算机技术的发展,使得不论是厂矿、机关,还是工业控制及测量领域,计算机之间的信息交换、资源共享,受到人们的极大重视,已取得了很大发展。网络将各个分散的计算机连在一起,形成一个网络系统,实现了信息和资源共享。网络技术已广泛地应用于国民经济的各个部门,网络的形式和结构也千变万化。而在工业控制及测量领域较为常用的网络之一就是物理层采用 RS-485 通信接口。因为 RS-485 具有网络连接方便、抗干扰性能好、传输距离远等特点。一般情况下计算机均有 RS-232 通信接口,而 RS-232 通信接口除数据传输距离短、抗干扰能力差等缺点外,它只适合两机之间数据通信,无法将更多的计算机组成一个网络。这样,就提出了将 RS-232 转换成 RS-485 的要求。本文所介绍的将 RS-232 转换成实用隔离型 RS-485 通信接口的方法是利用美国 Maxium 公司单电源隔离型 RS-485 接口芯片 MAX1480B 设计而成的。

1 MAX1480B 简介

MAX1480B 将光电耦合器、隔离型 DC-DC 变换器和 RS-485 驱动器集成在一个芯片内,是一个完整的输入/输出在电气上隔离的 RS-485 数据接口芯片,其主要特性如下:

① 隔离的数据接口。典型隔离电压的有效值大于 1 600 V;

② 具有用于无差错数据传送的转换速率限制特性;

③ 最高传输速率达 250 Kb/s;

④ 相对于隔离地具有 - 7 ~ + 12 V 的共模输入电压范围;

⑤ 单一 + 5 V 电源供电;

⑥ 1 μW 低功率关断模式;

⑦ 具有电流限制和热关断特性的驱动器过载保护功能。

MAX1480B 更详细内容请查阅参考文献 []。

2 转换电路设计

MAX1480B 其输入端为 TTL 电平或 CMOS 电平,而标准的 RS-232 通信接口都是经过 MC1488、MC1489 将 TTL 电平转换成 + 12 V 标准的 RS-232 电平,所以转换电路中首先应将 + 12 V 通信电平转换成 TTL 电平,再送 MAX1480B 将其转换成 RS-485 差动传输。其电路原理如图 1 所示。

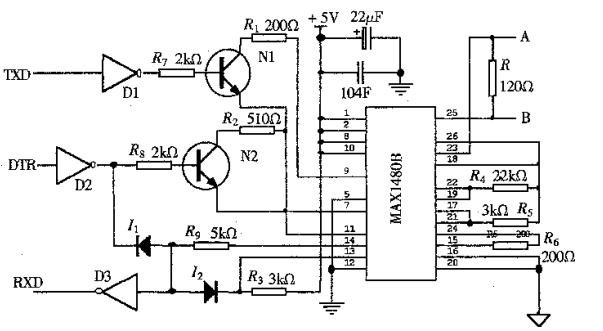

图 1 RS-232 与 RS-485 转换接口电路原理图

图中 RS-232 电平与 TTL 电平之间的转换由 ICL232 芯片完成。该芯片是由两个 RS-232 接收器和两个发送器组成,且由单一 + 5 V 电源供电。其接口

电路原理如图 2 所示。

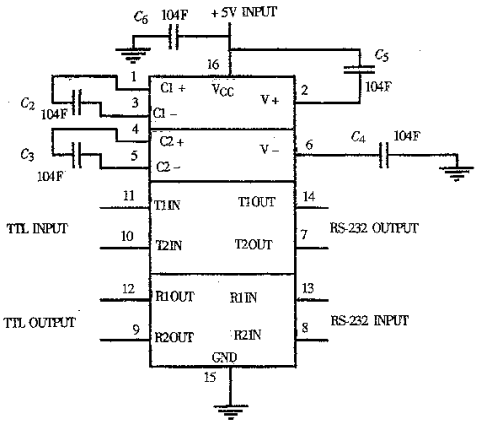

图 2　ICL232 接口电路原理图

数据发送由 RS-232 接收器 D1、R_7、R_1、N1 及 MAX1480B 芯片内光耦器件构成的电流环回路组成。“数据终端准备好” DTR(data terminal ready)使能线由 RS-232 接收器 D2、R_8、R_2、N2 及 MAX1480B 片内光耦器件构成的电流环回路组成。

该输出除具有 DTR 功能外,由于 ICL232 芯片只有两个 RS-232 接收器,所以用 D2 的输出和二极管 I_1 组成“请求发送”RTS(req to send)使能。当 DTR 有效允许发送时,RTS 无效,此时只能发送不能接收。而当 DTR 无效 RTS 有效时,只能接收数据而不能发送数据,从而保证 RS-485 通信线上数据的可靠性,避免因失控或干扰造成的收发数据混乱。数据接收由 RS-232 发送器 D3、R_3、I_2 及 MAX1480B 芯片内集电极开路的光耦器件组成。而 MAX1480B 隔离副边的 A 端为正相驱动器输出和正相接收器输入;B 端为反相驱动器输出和反相接收器输入。A、B 两端构成标准的 RS-485 数据通信接口。图 1 中 R 是网络匹配电阻,一般在网络两端的通信接口中使用。其余通信接口不使用,其阻值在 120 Ω 左右。

3　通信软件设计要点

图 1 所示电路是采用两线制差动传输,适合于半双工通信方式。既可应用到双机系统的点对点通信,又可应用于多机系统的网络通信。对于双机系统的点对点通信,其通信软件设计方法类似 RS-232,比较简单。但是对于多机系统的通信,在进行 RS-485 通信软件设计过程中,有几点须注意,如多机通信的地址识别问题、半双工方式通信的时延问题等都是很重要的。下面将讨论在多机系统通信中,如何解决 RS-485 通信中存在的这些主要问题。

(1)从机地址识别

半双工通信时,从机地址的识别是非常关键和重要的。一般解决此问题的方法是:由主机向从机发送不同的控制命令来识别各从机,但是这样的通信方式中,每个从机要接收主机和其他从机所有命令与数据,从而花费很多宝贵时间去接收处理不必要的信息,造成该从机的时间浪费。因此提出了一种简单而又实用的解决从机地址识别问题,那就是利用通信线路的奇偶校验位来识别从机的地址。

该方法为:主机发出从机地址时,主机允许奇偶校验位的奇校验,而所有从机只允许奇偶校验的偶校验。一开始主机发地址,所有从机处于听命令状态,从机奇偶校验和主机不一致,从机允许接收中断,收到主机发来的地址,所有从机进入中断服务例程,读取数据,和本机预定地址比较,当确认是本机地址时,设接收数据标志,允许接收数据。当主机发完从机地址时,其奇偶校验转换成偶校验,和从机奇偶校验一致,这样主机就和某一从机建立了数据通信关系。数据交换完毕,主机将奇偶校验位又变为奇校验。发另一从机地址,实现和另一从机的数据交换。从而实现了多从机地址识别。

(2)半双工通信的延时

在图 1 所示的电路中,当采用半双工通信方式时,通信方向的改变是通过 DTR 来控制的,根据 RS-485 通信规则,当主机(或从机)发送数据(或命令)时,DTR 应置为高电平,这时 MAX1480B 的驱动输出端(A 和 B)处于有效状态,即发送状态。当主机(或从机)发送完数据后,DTR 置成低电平,这时 MAX1480B 的驱动输出端(A 和 B)处于接收状态。这样,通过 DTR 的控制就可以容易地实现主机和从机的数据通信。

但是在数据发送到接收的转换过程中,DTR 由高电平变成低电平时,必须经过一段延时,否则会造成数据发送不完整或接收数据受阻。这主要是因为主机或从机发送的最后一个数据要经过图 1 的 RS-232/RS-485 转换接口电路中反相器和 MAX1480B 内部传输电路的延迟才能到达 RS-485 总线上去。如果主机或从机发送最后一个数据时,DTR 立即由有效转换成无效,该数据还未传输到 RS-485 总线上就将总线置成接收状态,从而造成最后一个数据发送受阻。因此,在数据发送到接收的转换过程中,DTR 的有效到无效转变必须经过一段时间的延迟。该延时可以通过软件编程来实现。在程序设计中,通过一段延迟程序就可容易地解决。

(3) 软件实现方法

主从机工作方式都采用中断方式,程序中开两个缓冲区,即数据接收缓冲区和数据发送缓冲区,它们的最大长度为 BUFsize。当要发送数据时,只要向发送缓冲区写入待发数据,然后允许发送中断,在中断服务程序中自动将缓冲区的数据发出去。接收也采用中断方式,所以只要判断接收缓冲区不空,则说明已接收到数据。所附程序以主、从机初始化串行口程序、中断通信子程序(以 COM1)为例。程序实现的是两线制半双工

数据交换,数据方向控制开关由 3FC 寄存器的 D_0 控制,发送数据时,3FC 的 D_0 为 1,接收时 D_0 为 0,此位直接控制线路 DTR(见图 1)。必须特别注意的是 DTR 开关换向的时候,发送完数据不能马上将 DTR 转向接收,如果那样会切断数据发送,必须等到发送寄存器空。程序如下:

```
# include < dos.h >
# include < bios.h >
# include < conio.h >
# include < stdlib.h >
# include < mem.h >
# define BUFsize 2048
# define Address 1
typedef struct {
                short insertHere,removeHere;
                unsigned short length;
                unsigned char buf[BUFsize];
                }Buffers;
void interrupt NewInt0CH(void);
static void interrupt ( * OldInt0CH(void) = NULL);
static unsigned char OldPort21H = 0;
static unsigned int Com1Buf = 0;
static unsigned char SendFlag;
static void restoreCom(void);
Buffers bufCom1;//Com 口通信接收缓冲区
Buffers bufcoms1;//Com 口通信发送缓冲区
void initCom(void)//COM1 初始化
{
  bufCom1.insertHere = 0;
  com1Buf = 0;
  bufCom1.removeHere = 0;
  bufCom1.length = 0;
  OldInt0CH = getvect(0x0c);
  OldPort21H = inport(0x21);
  Outportb(0x21,Oldport21H  0x18);
  setvect(0x0B,NewInt0BH);
  setvect(0x0C,NewInt0BH);
  outportb(0x3FB,0x80);
  outportb(0x3F8,0x70);
  outportb(0x3F9,0x00);
  outportb(0x3FB,0x1F);
  outportb(0x3FC,0x0b);
  outportb(0x3F9,0x04);
  inport(0x3F8);
  outportb(0x21,OldPort21H&0xE7);
}
//下段程序功能为从 COM1 硬中断服务子程序
void interrupt NewInt0CH(void)
{
  char byte;
  while(((byte = inport(0x3fa))&0x01) = = 0){
    switch(byte){
      case 0x06:
        if(inport(0x3fd)&0x04)
          if(inport(0x3f8) = = (Address))
            outport(0x3f9,0x03);
          else outport(0x3f9,0x04);
        break;
      case 0x4:
        bufCom1.buf[bufCom1.insertHere] = inport(0x3f8);
        bufCom1.length ++ ;
        if( ++ bufCom1.insertHere = = Bufsize)
          bufCom1.insertHere = 0;
        break;
    case 0x02:
      if(bufComs1.length = = 0)
        {outport(0x3fc,0x09);
          outport(0x3f9,0x01);
          break;
        };
      outport(0x3fc,0x0b);
      outport(0x3fb,SendFlag);
      byte = bufComS1.buf[bufComS1,removeHere];
      if( ++ bufComS1.removeHere = = Bufsize)
        bufComS1.removeHere = 0;
      outport(0x3f8,byte);
      - - bufComS1.length;
      break;
      };
    }
  enable();
  outport(0x20,0x20);
}
//下段程序功能为恢复原 COM 中断号子程序
void restorecom(void)
{
  outporb(0x21,inportb(0x21)|0x18);
  outportb(0x3FB,0x03);
  outportb(0x3FC,0);
  setvect(0x0C,OldInt0CH);
  outportb(0x21,Oldport21H);
}
```

4 结论

在工业控制、测量及远距通信中,网络通信的可靠性常常会受干扰及传输距离的影响。有时由于计算机与通信网络无法隔离或者因为隔离要增加电源等设备使系统使用不便,造成设备的维护性差、可靠性下降。采用以上 RS-485 转换电路,在不需外加电源的情况下,很容易实现 RS-232 到 RS-485 的转换。转换后的总线符合 RS-485 标准。除具有 RS-485 总线的特点外,还具有电路设计简单、电气上完全隔离、抗干扰能力强、可靠性高等优点。该电路已成功地应用于某广播电台发射台计算机监控系统,经长期运行实践证明,该转换电路稳定可靠。

参考文献

1 胡戎.单片机电源隔离型 RS-485/RS-422 数据接口芯片.电子技术应用,1995,(5)
2 方建淳,夏东培.ICL232 单电源双 RS-232 发送/接收器及其应用.电子技术应用,1993,(7)

作者简介:李玉忍,男,1962 年生,现为西北工业大学自动控制系副教授。

(收稿日期:1999-10)□