

WINDOWS3.1 与 8031 下位机 通信的一种新方法^{*}

朱跃杰 张奕鑫

(华侨大学机电工程系, 泉州 362011)

摘要 在 WINDOWS 编程中, 通过对 ANSI C 某些库函数的合理使用, 将其嵌入 WINDOWS 并实现以 WINDOWS3.1 编程为主体的程序同 8031 下位机串行通信. 解决工业系统中通过主机控制下位机的适时控制问题.

关键词 WINDOWS3.1, 通信, 函数

分类号 TP 311

在当前, 工业控制中, MCS51 系列 8 位单片机具有测控功能强、可靠性高、扩展灵活等特点. 它所构成的各种类型的应用系统, 已开始深入到各个技术领域, 如在电子技术改造和高科技领域中扮演着越来越重要的角色. 在软件编程中, 以 DOS 作为操作系统编写的程序, 也逐渐被基于 WINDOWS 操作系统的软件所取代. WINDOWS 操作系统的软件编程方法, 它与 DOS 的软件编程方法大相径庭. WINDOWS 的编程是一种多任务编程, 它允许在内存中运行多个不同程序代码段, 表现在界面上是多窗口的显示以及前台后台的处理. 总之, WINDOWS 要对整个系统资源进行合理的安排和使用, 避免了各任务间的相互冲突. 因此, WINDOWS 的编程要比 DOS 编程复杂的多. 为此, 本文介绍一种基于 WINDOWS 编程以实现 WINDOWS3.1 与 8031 单片机串行通信的新方法, 其程序经上机调试并通过.

1 关于 WINDOWS3.1 通信的一般规则

WINDOWS 与 DOS 在通信方面存在重要差别. 表现在 DOS 可以直接利用其内部的中断通道 IRQ3(COM2), IRQ4(COM1) 及 BIOS 中断 INT14, 以实现与下位机的串行通信. 其中, IRQ3 和 IRQ4 的中断服务例程只适用于固定波特率 1200 Hz 的通信. 而 BIOS 的 INT14 中断则有一比较大的可选波特率范围, 可由设计人员根据实际需要来选择波特率. 在 WINDOWS 编程中, 应用程序接管计算机的全部资源, 其中包括输入和输出设备、内存和显示器以及 CPU 本身. 因此, 在编写 WINDOWS 应用程序时, 不能直接访问串、并行端口, 以确保 WINDOWS 对这些资源的绝对控制. WINDOWS 作为一种设备无关的操作系统, 通过在硬件和软件之间增加设备驱动程序, 其实现与设备具有无关性. 设备驱动程序一方面连接可供

WINDOWS 使用的硬件设备, 另一方面为应用程序提供统一的编程接口^[1]. 串行通信也一样. WINDOWS 的串行通信程序是 COMMDRV. 应用程序调用标准的 API 通信例程, 将要发送的数据传递给通信驱动程序, 由驱动程序将数据从串行端口发送出去. 同样, 通信 9 驱动程序接受到达串行端口的数据, 然后通过标准的 API 通信例程通知应用程序进行处理. WINDOWS 在这方面提供诸如 Open Comm, Build Comm DCB 等多达 17 个标准函数以供使用, 功能非常强大, 但也增加了理解和编程的难度.

2 使用 ANSI C 函数服务 WINDOWS

C 语言是 WINDOWS 应用程序的开发语言, WINDOWS 的许多编程特征都是用 C 语言来实现的(也可用 PASCAL 语言或汇编语言来实现). 尽管 WINDOWS 有许多自身携带的函数可使用, 但由于 WINDOWS3.1 是 16 位的操作系统, 故其本身也可以使用 ANSI C 中的许多函数. 同时, 它也可以通过直接嵌入汇编语言来完善某些功能. 体现在通信方面, ANSI C 提供 BIOSCOM, INPORT, INPORTB, OUTPORT, OUTPORTB^[2]等一系列函数, 以实现串行通信的功能. 而在这些函数中, INPORT, OUTPORTB, OUTPORT, OUTPORTB 可直接为 WINDOWS 所利用. 通过这些函数对硬件端口读和写, 来解决 WINDOWS 同 8031CPU 下位机之间的串行通信问题. 这样, 不会导致程序对系统资源的独占性, 又避免了利用 WINDOWS 本身携带的通信函数而带来的程序复杂性.

3 硬件及接口电路

在 IBM-PC 机中, 控制异步通信适配器 I/O 端口的核心硬件是 INS8250 芯片. 它仅支持异步通信, 且采用标准的 RS-232C 接口. RS-232C 接口采用的是 EIA 逻辑电平, 在同 I/O 端口连接时需要电平转换. 通常的转换芯片有 MC1488, MC1489 或 ICL232. 在本次实验中采用的是 ICL232 转换芯片. 它将 EIA 电平转换成 8031 所需要的 TTL 电平, 同时也将 TTL 电平转换成 PC 机所需的 EIA 电平, 实现 8031CPU 与 IBM-PC 机的适时通信. ICL232 硬件连接情况如图 1 所示. 在 IBM-PC 机中, 控制通信异步适配器的 8250 内部, 有 9 个 8 位寄存器可被 CPU 访问. 在片选端低电平有效时, 通过引脚 A0, A1, A2 来确定何种寄存器被选中. 主要

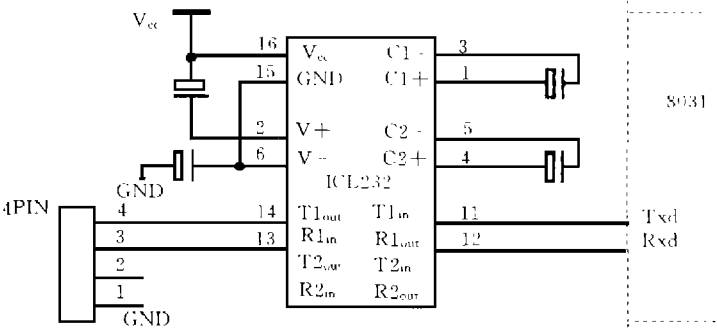


图 1 ICL232 同系统接口电路

寄存器的地址和功能: 3F8H 接收缓冲器(读)和发送寄存器(写) 除数锁存寄存器(低位) 除数锁存寄存器(高位) 波特率寄存器(低位) 波特率寄存器(高位) 模式寄存器 控制寄存器

3F9H 中断允许寄存器/除数锁存寄存器(高位); 3FAH 中断标志寄存器; 3FBH 线路控制寄存器; 3FCH MODEM 控制寄存器; 3FDH 线路状态寄存器; 3FEH MODEM 状态寄存器. 通过对 3F8H 和 3F9H 赋值, 就可以得到上位机通信时所需的波特率.

在 8031 下位机系统中, 上位机通过 ICL232 转变后传送来的信号直接送入 Txd 端, 经特殊功能寄存器 SCON 判断数据正确后, 系统转向中断, 开始接收数据. 下位机通常采用定时器 T1 进行波特率的设置. 根据接收数据的格式的不同, 确定采用方式 1 或方式 3 进行通信. 此时波特率的计算公式: 方式 1/3 波特率 = 振荡器频率 / $32 \times 12[256 - (\text{THE1})]$ ⁶⁾.

在设置完上下位机的通信波特率、数据格式等一系列通信协议后, 就可进行数据通信.

4 软件的编写

本次的上位机软件采用了 BORLAND C++ 4.5 开发工具进行编程, 下位机软件用 MCS51 汇编语言编写.

列举程序时, 省略了其它无关软件. 现假设从上位机发送 8 个字符 1, 2, 3, ..., 8 给下位机, 下位机 50H ~ 57H 单元接收数据. 同时, 下位机也发送一系列数据给上位机. WINDOWS 的消息 WM - INITDIALOG 作为初始化端口用, WM - SENDD 由文件定义作为传送数据开始的消息, 而 WM - RECE 由文件定义则作为接收数据开始的消息.

CASE WM - INITDIALOG:

```
    outportb(0x3fd, 0x80);          /* 选择串口 1* /
    outportb(0x3f8, 0x17);          /* 设置波特率参数* /
    outportb(0x3f9, 0x04);
    outportb(0x3fd, 0x03);          /* 设置数据格式* /
    outportb(0x3fc, 0x03);
    outportb(0x3f9, 0x00);          /* 禁止中断* /
```

BREAK;

CASE WM - SENDD: /* 发送数据* /

```
    for(int i= 0, i< 9, i+ + )
    {   while(! (inportb(0c3fd) & 0x20))
        outportb(0x3f8, iCh);
        iCh+ = 1; }
```

BREAK;

CASE MM - RECE: /* 接收数据* /

```
    IF(! (inportb(0))
    { j= inportb(0);
      j+ + ; }
```

ELSE MessageBox (hdlg, "TECEIVE OVER!", "CAUTION",

M - ICONEXCLAMATION MB - OK);

BREAK;

下位机接收/发送程序:

```

MOV    PSW, # 00H
MOV    TMOD, # 21H    ;初始化 T1          SETB    TR1    ;启动 T1
MOV    TH1, # 72H      ;置波特率          MOV    R7,  # 8
MOV    TM1, # 0H       MOV    R0,  # 50H
MOV    SCON, # 50H     ;初始化 SCON       SETB    TR1    ;启动 T1

WAIT: JBC    RI,  TR    ;等待接收
      SJMP   WAIT
TR:   MOV    A,    SBUF
      MOV    @R0,  A
      DINZ   R7,  WAIT
SENT: MOV    A,    @R1,    ;发送数据
      MOV    SBUF,  A
      INC    R1
      AJMP   SENT
      RET

```

5 结束语

通过合理的利用 ANSI C 有关函数编写 WINDOWS 通信程序, 可以通过简洁的方法实现 WINDOWS 与 8031 工控机的通信功能. 同时, 它能避免利用 WINDOWS 本身携带的通信函数而带来的程序复杂性, 减少出现错误的机会.

参 考 文 献

- 1 吕东庚. DCS 监督操作软件的设计方法研究. 华侨大学学报(自然科学版), 1996, 17(4): 429~434
- 2 施小龙, 葛玉宝, 邓明辉等. BORLAND C++ 3.1 库函数快速参考. 北京: 学苑出版社, 1994. 273~337
- 3 何立民. MCS-51 系列单片机应用系统设计. 北京: 北京航空航天大学出版社, 1992. 371~391

A New Method of Communication between WINDOWS3.1 and 8031 Microcomputer

Zhu Yuejie Zhang Yixing

(Dept. of Electro. Mech. Eng., Huaqiao Univ., 362011, Quanzhou)

Abstract For solving timely control of main frame over 8031 microcomputer at low-level position in industrial system, the serial communication between the programs with WINDOWS3.1 programming as main body and 8031 microcomputer at low-level position can be realized by embedding some built-in function of ANSI C into WINDOWS3.1 as its rational use.

Keywords WINDOWS3.1, communication, function