

PC机与单片机的远距离多机通信问题

李教, 许杰

(空军工程大学 电讯工程学院, 陕西 西安 710077)

摘要:单片机被广泛应用于各种控制、数据采集系统中;PC机功能强大、人机界面友好,是单片机所不能及的。由单片机和PC机构成的多机系统可以实现更加复杂的控制。文中从硬件构成、通信过程、软件实现等方面着重介绍了PC机和单片机的远程多机通信。

关键词:RS-485;远程通信;单片机;VB通信控件

中图分类号:TN913.2 **文献标识码:**A **文章编号:**1009-3516(2001)04-43-46

在比较复杂的控制和数据采集系统中,往往需要多个单片机独立构成子系统,完成各自的采样和控制任务,整个系统的协调、复杂的数据处理、输出及各种人机界面需要PC机来完成。在这种系统中,关键在单片机和PC机之间的数据传递。本文主要讨论PC机和单片机的串口通信问题。

1 系统的硬件构成

PC机和单片机的串行通信是利用PC机的RS-232串行通信口实现的。该口采用RS-232通信标准,其结构简单,只要三根线就可以完成通信任务;但缺点是带负载能力差、通信距离不超过十几米。为了进一步地扩大通信范围可以采用RS-485通信。RS-485通信采用差动的两线发送、两线接收的双向数据总线两线制方式,其通信距离可达1200m^[1]。

为了解决RS-232到RS-485的电平转换,可选用一块Evoc的远端数据采集与控制模块ADAN-4520。ADAN-4520是带隔离的RS-232到RS-485的电平转换器。其中RS-232四线(RTS、TXD、RXD、GND),RS-485两线(OUT、IN)。硬件连接简单如图1所示。它和PC机的连接可以通过ADAN-4520上的标准RS-232串口插座插在PC机的串口1或串口2上。只是在和下位机连接时要注意ADAN-4520的电源地和所有下位机的地接在一起,用来提高传递信号的质量。

下位机除了根据每一个子系统要求设计硬件外,还需增加一片RS-485接口芯片。以MAX485为例:它是适用于RS-485和RS232通信的低功率收发器,每片包括一个驱动器和一个接收器^[2]。它和51单片机的连接如图2所示,其中由P1.6控制MAX485的输入和输出。

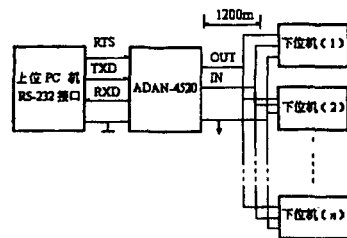


图1 系统连接框图

2 多机通信过程

这里介绍一种简单可靠的通信过程。在通信协议中规定:当从机接受到命令代码时向PC机发送一批数据。首先由上位PC机发送所要寻址的下位机地址(以01H为例),当所有从机接受到01H后关中断,进入中断服务程序和本机地址比较;地址不相符开中断退出中断服务程序;地址相符的01H号从机回送01H给PC机。当PC机接受到回送的地址后,握手成

收稿日期:2000-09-25

作者简介:李教(1963-),男,江苏南京人,讲师,博士生,主要从事控制理论研究。

功。PC机发00H命令给01H从机命令其发送一批数据,数据发完后关串行中断。实际运行过程特别注意:1)上下位机的时序配合问题,一定要根据发送数据的长短设置延时等待。2)要设置出错处理程序,例如:当80H从机接受不到00H命令时能跳转出来。

3 通信软件设计

3.1 PC机通信程序设计

在Windows98环境下,虽然调用Windows的API应用程序接口函数也能进行串行通信,但一般用户较难掌握^[3]。这里介绍如何利用VB6.0通信控件(MSComm)来编制PC机通信软件的方法。该方法简单易掌握。MSComm控件很多,在使用之前,需了解其重要属性。

| 属性 | 描述 |
|----------------|---|
| CommPort | 设置并返回通信端口号 |
| Settings | 设置并返回波特率(BBBB)、校验(P)、数据(D)、停止位(S) |
| InBufferSize | 设置并返回发送缓冲区的大小,以字符为单位 |
| OutBufferSize | 设置并返回发送缓冲区的大小,以字符为单位 |
| PortOpen | 设置并返回端口状态,也可以用于打开和关闭串口 |
| InBufferCount | 返回接收缓冲区等待读出字符数量,当InBufferCount=0时接收缓冲区清零 |
| OutBufferCount | 返回发送缓冲区等待输出字符数量,当OutBufferCount=0时发送缓冲区清零 |
| InputMode | 设置并返回接收类型(字符方式,二进制方式) |
| InputLen | 设置并返回从接收缓冲区读取的字符 |
| Input | 返回并清空接收缓冲区数据 |
| Output | 把字符串写入发送缓冲区 |

使用MSComm控件编写的PC机到单片机通信子程序一般由三部分组成:通道初始化程序,发送和接收程序,数据处理程序。

3.1.1 通道初始化程序

通过通道初始化程序,我们完成PC机串口工作方式的设置。

```
MSComm1.CommPort = 2  设串行通信为串口2
```

```
MSComm1.InBufferSize = 512  设输入缓冲区长度
```

```
MSComm1.OutBufferSize = 510  设输出缓冲区长度
```

```
MSComm1.PortOpen = True  打开串口
```

```
MSComm1.InputLen = 0  设置读出时读出输入缓冲区所有有用数据
```

```
MSComm1.InputMode = comInputModeBinary  设串入数据为二进制方式
```

```
MSComm1.Settings = "9600,n,8,1"  设波特率为9600,不校验,数据8位,停止位1位
```

在通道初始化时要注意校验位的设置,一般情况下在校验位的设置时应采取不设校验位($P=n$),这是因为设了校验位($P=e$ 偶校验, $P=o$ 奇校验)后,当由下位机发送过来的数据不满足校验规则时,PC机将接收不到发来的数据而只得到一个“3FH”的错误信息。所以在多数据传送时要避免使用校验位来验证接收数据是否正确,应采取其它方法来验证接收是否正确,如可发送这批数据的累加和和累减差等。

3.1.2 发送和接收程序

在发送和接收前首先要实现握手,握手成功后才能发送和接收数据,可用下面程序实现握手。

```
MSComm1.Output = state  发出待寻访下位机地址
```

```
MSComm1.OutBufferCount = 0'  清空输出缓冲区
```

```
MSComm1.InBufferCount = 0'  清空输入缓冲
```

```
Do  等待回送地址
```

```
DoEvents
```

```
Loop Until MSComm1.InBufferCount <> 0  输入缓冲区有数据
```

```
Array() = MSComm1.Input  把串入缓冲区的数据放入字节数组 Array()
```

```
If Array(0) <> stae Then  返回地址不正确
```

```
MsgBox "shakfalure", 52, "error" 握手失败
```

```
End If 握手成功
```

握手成功后就可以发送和接收数据。发送数据有两种方式:

1)按字符数据发送:MSComm1. Output = "This is a line of text"

采取这种方式发送数据时要注意一次发送的数据不能太长,否则可能会发生溢出错误。

2)按字符数组数据发送:这时必须事先定义、赋值字符数组

```
Dim Array as byte() 定义动态数组
```

```
ReDim Array(1) 重定义数组大小
```

```
Array(0) = 0
```

```
Array(1) = 1
```

```
MSComm1. Output = Array 把字符数组 Array 放入发送缓冲区
```

接收数据时一定要注意以下问题:

1)时序的配合问题,以免出错。如果知道接收的数据长度,可采用下面方法:

```
Do Until MSComm1. InBufferCount < 10 等待 10 个字节的数据接收完
```

```
DoEvents
```

```
Loop
```

```
Buffer = MSComm1. Input
```

如果不知道发过来的数据长度,可采用下面方法:

```
Do
```

```
DoEvents
```

```
Loop Until MSComm1. InBufferCount > 0 有数据发来
```

```
Timer. Enabled = True 等待下位机发送数据
```

```
Do
```

```
DoEvents
```

```
Loop Until timer_on = True
```

```
Timer. Enabled = False
```

```
timer_on = False
```

```
Array() = MSComm1. Input 把缓冲区数据放入字节数组 Array()中
```

这时候必须等待下位机数据发送到一定时间后才能从缓冲区读数,否则会造成数据丢失。延时的时间可以由接收的数据长短决定。同时不要定义动态数组长度。

2)当接收(或发送)大于128的字符数据时不能按字符方式进行,应按二进制字符数组方式,否则就会出错。这是因为中文Windows使用的是双字节字符集(DBCS)系统。DBCS系统使用0-128之间的数字表示ASC II字符,大于128的数字仅作为前导字符,并不表示实际意义。

3.1.3 数据处理程序

在数据处理之前必须从动态数组Array()中取出数据。取数时可以用UBound()函数得到动态数组Array()的长度。再按要求赋予相应的数组。下面给出两种赋值方法:

```
For i = 0 To UBound(Array)
```

```
HexString(i) = Hex(Array(i)) 如果需要十六进制数,把Array()放入数组 HexString()
```

```
MyString(8 * i + 0) = Not((Array(i) And (&H1) = 0) '如果是布尔变量,取出Array()的每一位放入数
```

```
组 MyString()
```

```
MyString(8 * i + 1) = Not((Array(i) And (&H2) = 0)
```

```
.....
```

```
MyString(8 * i + 7) = Not((Array(i) And (&H80) = 0)
```

```
Next i
```

3.2 单片机通信程序设计

MS-51单片机功能强大,适合于多机通信。在编程时应使所有单片机串口工作于模式2或模式3;要

注意上下位机的波特率要一致。并使所有单片机 $SM2 = 1$ 使它们处于只接收地址帧状态,由于上位机不设校验位,故在某下位机向 PC 机发数时应使该机 $SCON.TB8 = 0$ 以避免其它下位机当作地址接收。还要注意的就是单片机通信程序在结构上要 and PC 机通信程序相配合。下面就串口初始化、数据接收和数据发送分别加以介绍。

3.2.1 串口初始化

在串口初始化中完成对串行口 $SCON$ 、波特率的设置。多机通信时所有下位机串口工作于模式 2 或模式 3,并置 $SM2 = 1$ 使地址帧可中断所有从机,而这些从机将不理睬数据帧。同时在某一下位机向 PC 机发数时,应使其 $TB8$ 置 1 表示发送的是数据,以免其它下位机当作地址误接收。

MOV $SCON, \#1110000B$; 设置串口工作模式 3, $SM2 = 1$, $REN = 0$ 禁止接收

MOV $TMOD, \#20H$; 设置定时器 1 为模式 2

MOV $TH1, \#0FDH$; 设置波特率为 9600

MOV $TL1, \#0FDH$

SETB $TR1$; 启动定时器 1

SETB ES ; 开串行口中断

3.2.2 串行接收和发送

为了尽量少占用 CPU 时间,单片机串行通信程序一般放在中断里进行,当 PC 机发出呼叫信号时,单片机就转入串行通信服务程序,完成接受或发送。

1) 接收

SETB REN ; 允许接收
 SETB $P1.6$; 置 485 为接收口
 CLR RI ; 清接收中断标志
 MOV $A, SBUF$; 把缓冲区内容放入 A

2) 发送

CLR $P1.6$; 置 485 为发送口
 CLR REN ; 禁止接收
 CLR $TB8$; 表示发送的是地址
 MOV $SBUF, A$ 发一个数据
 JNB TI, S 等待发完
 CLR TI 发完后清发送中断标志

以上介绍的实现 PC 机与单片机的远距离多机通信硬件系统和编程方法可以较好地解决 PC 机与单片机的远距离多机串行通信问题。我们在实际应用中已证明了这种方法简单可靠。

参考文献:

- [1] 邬宽明. 单片机外围器件实用手册[M]. 北京:北京航空航天大学出版社,1998.
 [2] 尹红,唐煜. 单片机 RS-485 多机通讯的实现[J]. 计算机应用 1999,19(1):52-54.
 [3] 贾蕙芹,张拥军. 用 VB 通信控件开发微机与单片机的串行通信程序[J]. 现代电子技术,1998,(7):13-14.

The Problem of the Remote Serial Communications Between Computer and Chip Microcontrollers

LI Jiao, XU Jie

(The Telecommunication Engineering Institute of the Air Force Engineering University, Xi'an 710077, China)

Abstract: Chip microcontrollers are widely used in the system of data measurement & control. Computer have far more powerful function and friendly interface than chip microcontrollers. More complicated control could be realized by combining computer with chip microcontrollers. In this paper we presents the system constitution, the process and the software realization of the remote serial communications between computer and chip microcontrollers.

Key words: RS-485; remote communication; chip microcontroller; mscmm. control of visual basic