

机器人等离子喷涂数据通信及显示技术的研究

朱六妹,刘明,王伟,田非

(华中科技大学 材料学院,湖北 武汉 430074)

摘要:利用 VB6.0 提供的 MSCOMM 通信控件,开发了 Windows 环境下的串口通信程序。以等离子喷涂控制系统为例,介绍了 PC 机与多片单片机进行串口通信的软件特点,以及利用 MSCOMM 控件制定通信协议和收发串口数据的方法。试验结果表明,该方法不仅为实时加工系统提供了丰富的图形显示界面,而且具有简单易行、运行可靠、功能强等特点。

关键词:等离子喷涂;串口通信;MSCOMM 控件

中图分类号:TG409

文献标识码:A

文章编号:1001-2303(2006)07-0048-04

Communication and display of data in robotic plasma spray system

ZHU Liu-mei, LIU Ming, WANG Wei, TIAN Fei

(Welding Research Institute, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: It is convenient to write a serial port communication program under Windows environment by using MSCOMM controller and VB6.0. The important properties of the MSCOMM controller was described, and the software design was illustrated for the communication between PC and single-chip microcomputer by a practical project. For the slave computer, the MSCOMM controller is used to receive and transmit serial data. The engineering practice proves that it is easy in implementation, applicable in functions and reliable in operation.

Key words: plasma spraying; COM port communication; MSCOMM controller

0 引言

等离子喷涂技术是利用等离子焰流为热源,将加热至熔融状态的喷涂材料通过高速气流使其均匀雾化,在工件表面形成致密的涂层,从而达到表面强化、改善工件使用性能的目的。在我国,目前已生产出晶闸管整流式磁放大器式喷涂用电源、喷枪和送粉器,但是对于控制系统,仍采取传统的继电控制方式和 LED 显示,不能直观地显示实时工艺过程和工艺参数曲线^[1]。为此,本研究开发了 1 套分布式等离子喷涂控制系统,采用 2 套嵌入式单片机作为下位机进行数据采集和过程控制,并通过串行口与基于 PC104 总线的 3.5" 单板工控机进行通信,完成实时工艺参数的图形显示和数据文件的存储,取得了令人满意的结果。

的 2 个串行口(RS232)经过 MAX232 专用芯片与单片机的 UART 接口相连,实现全双工的数据通信。单片机 1 和单片机 2 通过 TXD 向工控机发送喷涂工艺过程的状态信息和现场采集的电流、电压、送粉量、温度等数字量,同时经过 RXD 接收工控机发来的应答信号和校验码。为了保证数据通信的可靠性,在每个通道上采用硬件握手线,同时在软件上采用累加校验和奇偶校验的方法,确保数据在通信过程中的正确性。

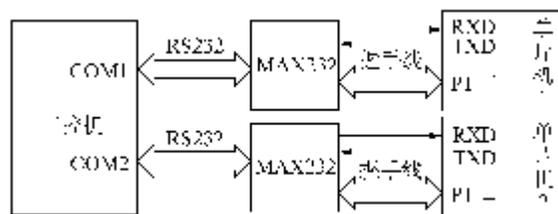


图 1 通信系统接口电路

1 通信系统的组成

通信系统的接口电路如图 1 所示,其中工控机

收稿日期:2005-12-01;修回日期:2006-03-01

作者简介:朱六妹(1947—),女,上海市人,副教授,主要从事焊接过程智能控制及 CAD 设计工作。

2 单片机通信程序设计

2.1 显示工艺状态过程的通信程序设计

图 2 是显示工艺状态的通信子程序框图。其中

存放在标记单元中的数据是表示工艺状态的数字量,是预先规定好的,如:01H 表示接通主电路,10H 表示断开主电路,02H 表示送气,20H 表示停止送气等。当单片机发送代表工艺状态的数字量和奇偶校验码以后,等待接收上位机发回的反馈信号和奇

偶校验码,确定无误后,上位机才显示工艺状态的信息。否则单片机将重新发送原来的数字量,直到接收的数据和奇偶校验码正确为止。单片机 1 除了实现对工艺过程的控制以外,同时完成与上位机的数据通信任务。

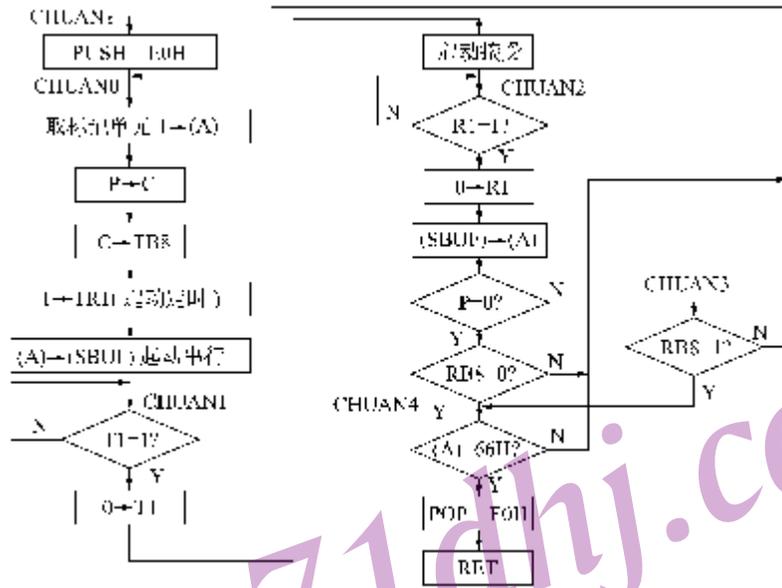


图2 带奇偶校验位的串行通信子程序

2.2 显示工艺参数的通信程序设计

本系统中的第 2 套单片机的主要任务是对喷涂中的电流、电压、送粉量、温度等进行实时数据采集。其工作方式有 3 种:离线、在线和通信。离线方式的设计是为了实现高速数据采集,用于记录连续参数的局部特征,其最大存储量为 65 536,程序的设计保证了数据的连续性,同时能使数据首尾相连和重复覆盖,从而保证了采样过程不受时间的限制。在线工作方式是一边采集,一边将采集的数据通过串口传送到上位机进行图形显示。图 3 是实现采用累加和校验法的数据通信程序框图。单片机 2 采用同步通信方式,在每次发送的一帧数据中包括电流、电压、送粉量、温度的采样值以及对这些采样值进行累加的和(不考虑进位位)。当上位机收到这一帧数据后,也进行累加运算,同时将计算结果发送给单片机,单片机在收到累加和校验码后进行判断,当发送的累加和与接收的累加和相等时,单片机才进行下一次的采样和通信,否则重新发送。通信方式是将离线方式所采集的数据传送到上位机,其采用的通信方法和校验方法与在线方式相同。

上位机采用基于 PC104 总线的嵌入式单板 3.5"微机。PC104 总线与台式机 PC(AT)总线完全兼

容,具有高质量、高可靠性、小尺寸、低功耗的特点。其主频为 667 MHz,并配备了 12.1"液晶显示屏,操作系统选用 Windows 2000。

3 上位机通信编程设计

上位机的主要任务是开发串口通信和波形显示的程序。在此采用在 Windows 应用程序中调用微软提供用于串行通信控制的 ActiveX 控件 MSCOMM,很方便地实现设备间的串口通信^[9]。在对 MSCOMM 的操作过程中,一些通信过程中的底层操作都被屏蔽,只需了解如何通过给定函数实现对串口的操作、完成数据的处理和显示,从而完成 PC 机与单片机的数据传输任务。

3.1 界面设计

进行界面设计时,除了常规的标签和按钮控件外,还采用 Shape 控件颜色的变化来模拟指示灯的开合,并用 Picture 控件来实时动态显示数据。对于 MSCOMM 控件,其属性及其意义如下:

CommPort—该串口占用的端口号;Settings—设置串口的通信波特率、奇偶校验位、数据位个数、停止位个数;InputMode—接收模式,即是二进制方式接收(1)还是文本方式接收(0,默认);PortOpen—

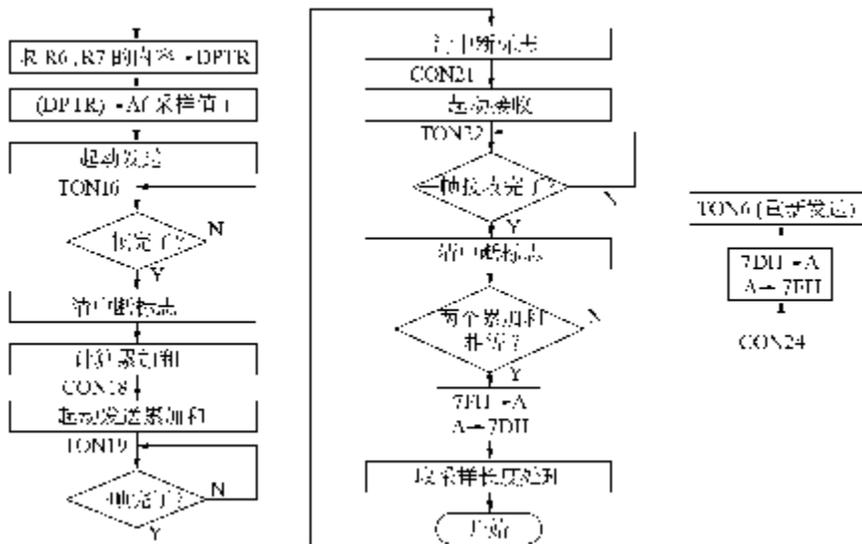


图 3 采用累加和校验的数据通信程序框图

是否打开串口; Inputlen—从接收缓冲区每次读取数据的个数, 默认为 0, 即每次读取整个缓冲区; InBufferCount—接收缓冲区内的数据个数; Input—读接收缓冲区内的数据; Output—向发送缓冲区写数据; DTREnable—Data Terminal Ready 线是否有效(PC→Modem); RTSEnable—Request To Send 线是否有效(PC→Modem); DSRHolding—Data Set Ready 线是否有效(Modem→PC); CTS Holding—Clear To Send 线是否有效(Modem→PC)。

采用 MSCOMM 控件设计的界面如图 4 所示。

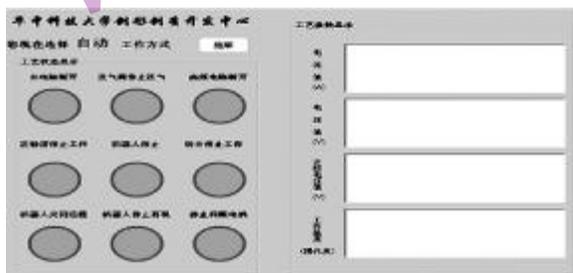


图 4 利用 MSCOMM 控件设计的等离子喷涂工作界面

3.2 通信程序设计

用 VB6.0 开发上位机程序中, 上位机完成通信任务主要有定时查询和中断处理 2 种方式。在查询方式中, 通过设置定时器的 Interval 属性定时查询 InBufferCount 属性, 以判断接收缓冲区是否接收到了相应数目的字符或字节, 若满足要求, 则可以通过 Input 属性以读取缓冲区内的数据并进行处理, 否则继续等待直到 InBufferCount 属性满足要求; 在中断处理方式中, 当串口发生接收发送事件以及发

生传输错误时, 都会产生一个 On_Comm 事件以通知程序处理, 通过查询通信控件的 CommEvent 属性来判断现在的串口状态并处理通信事件, 如对于接收事件, CommEvent 属性的值为 comEvReceive。本研究采用了定时查询的方式。

3.2.1 初始化程序

程序开始接收数据之前需要对串口进行初始化, 其语句如下:

```
Private Sub Form_Load( )
    comm1.CommPort=1
    comm1.Settings="9600,n,8,1"
    comm1.InputMode=comInputModeBinary
    comm1.InputLen=1
    If Not comm1.PortOpen Then
        comm1.PortOpen=True
    End If
    Timer1.Enabled=True
End Sub
```

对于串口 2 的初始化也是放在 Form_Load() 过程中, 方法与串口 1 类似, 即每隔一段时间查询 2 个串口的状态, 当查询状态符合协议要求时, 则开始处理串口数据。

3.2.2 数据的读取

串口 1 数据处理主要在函数 Comm1DealData() 中。在该函数中, 首先要把数据从接收缓冲区内读出来。单片机与 PC 机是以二进制方式进行传输的, 用 Input 属性从缓冲区读出的数据不能直接赋值给

Byte 类型的变量,必须先将数据存储在 Variant 型变量中,然后再把 Variant 型变量赋值给 Byte 型变量。

在 Comm1DealData()函数中,读取数据部分如下:

```
Dim temp1 as Variant
Dim Data1 as Byte
temp1=comm1.Input
Data1=temp1(0)
```

每接收到一个数据后,PC 机就返回一个特定的数据告诉单片机可以进行下一步的操作,并通过图形控件 shape 的颜色变化来直观告诉生产者现在所处工作状态^[3]。

串口 2 采用的是同步方式,每一帧数据包含 5 个字节,数据传输量比较大,从而导致单片机和 PC 机在大量数据传输过程中很难完全同步,因此采用了硬件握手的方式来保证数据传输的完整和正确。在 RS232 的 DB-9 接口中,除了第二、三、五脚分别表示接收端、发送端、接地端外,第四脚 DTR(Data Terminal Ready)、第六脚 DSR(Data SetReady)、第七脚 RTS(Request To Send)、第八脚 CTS(Clear)在实际中也得到广泛的应用。PC 机通过定时器中设置 DTR、RTS 的电平状态以及查询 DSR、CTS 的状态,保证了在大数据量的情况下能与单片机正常通信。设计的程序中语句如下:

```
comm2.DTREnable=True
comm2.RTSEnable=False
If Not comm2.DSRHolding And comm1.CTSHolding
```

Then

```
Call Comm2DealData( )
```

```
End If
```

在 Comm2DealData()中,对于 comm2 的数据处理,采用了以下方法:

```
Dim i as Integer
i=0
While comm2.InBufferCount>0
Comm2.InputLen=1
Temp2=comm1.Input
DataInBuffer2(i)=temp1(0)
i=i+1
```

```
Wend
```

3.3.3 数据的检验

采用了校验和的方法对数据进行检验以避免传输过程中的错误。单片机将前 4 个字节的数据(舍弃进位)相加所得之和作为第 5 个数据发给 PC 机,

PC 机此时进行运算并决定是否保存所读数据,设计的代码如下:

```
For j=0 to 3
sum=sum+DataInBuffer2(j)
next j
If sum Mod 128=DataInBuffer2(4) Then
.....
```

3.3.4 实时曲线的绘制

读出串口 2 缓冲区内的数据后就要进行绘图处理了。较常用的方法是采用 Picture 控件的 Line 方法,实现原理如下:

```
Picture1.Scale(0,ymax)-(xmax,0)
Picture1.Line(x1,y1)-(x2,y2)[,color]
```

将从串口 2 读出的数据存入定长数组,并在存入数组的同时用 Line 方法绘制曲线。当数组存满后,新采的数据将覆盖原来的老数据,并采用 Picture 控件的 Cls 方法清空绘图区开始绘制新的趋势曲线,从而实现不间断的曲线显示。实际系统所采集的数据波形图如图 5 所示。

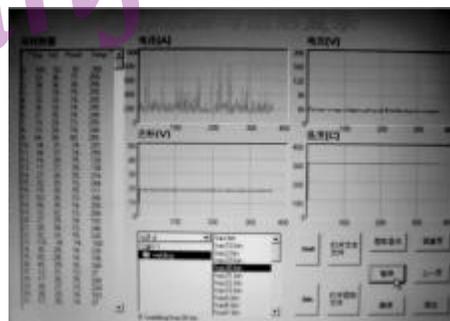


图 5 系统采集的数据波形

4 结论

采用单片机与 PC 工控机构成的多机控制系统,不仅能适用复杂的工业过程控制,而且单片机汇编语言与 VB 语言能够很容易地实现上位机与下位机的数据通信,具有强大的图形显示功能、适时性及灵活性,能增添加工设备的智能化程度,提高设备的显示精度、加工能力和使用范围。

参考文献:

- [1] 周庆生.等离子喷涂技术[M].江苏:江苏科学技术出版社,2002.
- [2] 付旭,方康玲.利用 VB6.0 实现 PC 机与单片机的串口通信[J].微计算机信息,2002,18(10):30-31.
- [3] 白鹏.Visual basic 6.0 高级编程技巧[M].西安:西安交通大学出版社,1999.