
步进电机教学演示系统

小组成员	崔明明	11 机电（1）
	唐翔	11 机电（1）
	徐灵敏	11 机电（1）
	戚炯杰	11 机电（2）
	裘焱斌	11 机电（2）
	邹玲玲	11 机电（2）
	周林	11 机电（2）
	仰昊	11 机电（2）
	林健	11 机电（2）

指导教师	李晓明
------	-----

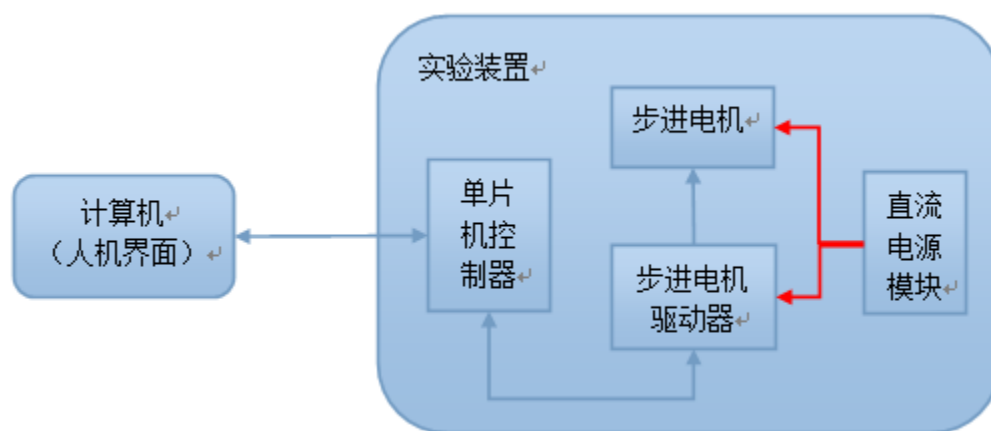
目 录

一、 目的	1
二、 系统结构说明.....	1
三、 系统实施方案.....	2
(一) 上位机软件设计	2
1. 功能设置.....	2
2. VB 程序代码	3
(二) 通讯协议.....	7
1. 概述.....	7
2. 通讯串口参数.....	7
(三) 下位机软件设计	9
四、 实物连接	15

一、目的

实现一个用于课堂演示的实验系统，在随堂教学中展示步进电机的外形，步进电机控制器的实物，阐述其各自的工作原理，在此基础上，可以演示步进电机运动过程中的各种现象，例如步进，加减速，失步等。甚至更可以让同学亲自编写一下控制程序，体验一下亲自控制步进电机的感觉。

二、系统结构说明



整个系统分成两个子系统，分别是计算机（上位机）系统，含人机界面和控制软件，以及下位机也就是实验装置子系统，包括控制器，步进电机驱动器和步进电机执行器以及电源模块。两个子系统通过USB通讯电缆连接在一起，实现两个子系统之间的通讯。

步进电机，步进电机驱动器均为现有产品，按照产品手册进行连线后即可使用。单片机控制器需要自行设计，其主要作用是根据上位

机的发出的指令，转换为步进电机驱动器可以接受的脉冲和方向电平信号，实现对步进电机的运动控制。

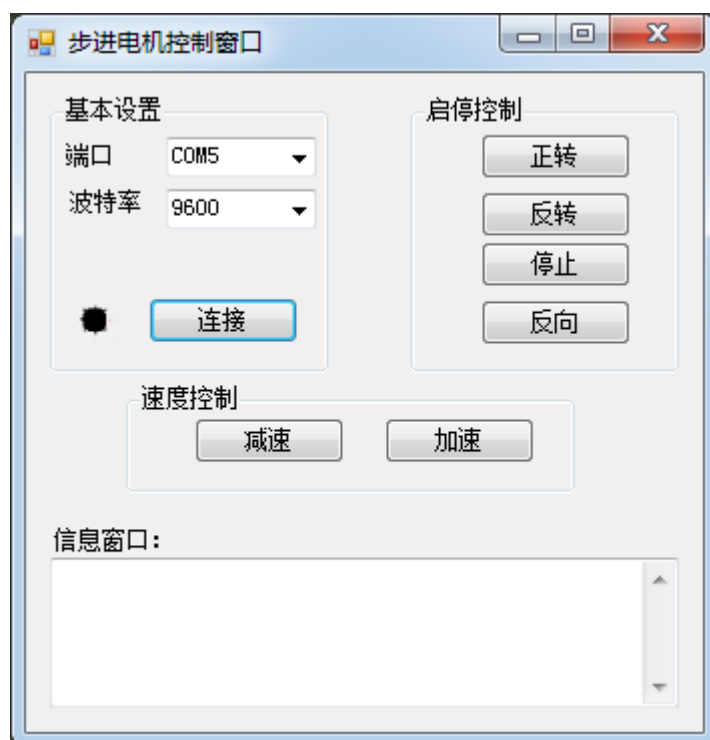
三、系统实施方案

(一)上位机软件设计

1. 功能设置

根据实验要求，步进电机教学演示系统需要实现加速、减速、步进、反向、停止、正转、反转等功能。此外，为了更好地说明步进电机的运动过程，还需添加显示步进电机信息反馈窗口。

设计窗体如下：



2. VB 程序代码

```
Public Class Form1

    Dim x As String

    Dim y As String

    Private Sub open_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles open.Click

        If (open.Text = "连接") Then

            sp.BaudRate = ComboBox2.Text

            sp.PortName = ComboBox1.Text

            open.Text = "关闭"

            sp.Open()

            pic.ImageLocation = "H:\xx\2013 下学期\机电传动
与控制\实验\我的数据
\WindowsApplication1\WindowsApplication1\Resources\hong.png"

            TextBox1.Text = "串口已连接..."

            Timer1.Start()

        Else

            open.Text = "连接"

            pic.ImageLocation = "H:\xx\2013 下学期\机电传动
与控制\实验\我的数据
```

\WindowsApplication1\WindowsApplication1\Resources\hei.png"

TextBox1.Text = "串口已关闭..."

x = "2"

sp.Write(x)

sp.Close()

Timer1.Stop()

End If

End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

x = "0"

sp.Write(x)

' TextBox1.Text = TextBox1.Text + vbCrLf + sp.ReadExisting

End Sub

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click

x = "1"

sp.Write(x)

Call Delay(1000)

```
        ' TextBox1.Text = TextBox1.Text + vbCrLf + sp.ReadExisting  
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button3.Click  
  
    x = "2"  
  
    sp.Write(x)  
  
    Call Delay(2000)  
  
    'TextBox1.Text = TextBox1.Text + vbCrLf + sp.ReadExisting  
End Sub
```

```
Private Sub Button6_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button6.Click  
  
    x = "9"  
  
    sp.Write(x)  
  
    Call Delay(1000)  
  
    'TextBox1.Text = TextBox1.Text + vbCrLf + sp.ReadExisting  
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button5.Click  
  
    x = "-"
```

```
sp.Write(x)

Call Delay(1000)

'TextBox1.Text = TextBox1.Text + vbCrLf + sp.ReadExisting

End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click

    x = "+"

    sp.Write(x)

    Call Delay(1000)

    'TextBox1.Text = sp.ReadExisting+ vbCrLf +  TextBox1.Text

End Sub
```

```
Private Sub sp_datareceived(ByVal sender As Object, ByVal e
As System.IO.Ports.SerialDataReceivedEventArgs) Handles
sp.DataReceived
```

```
End Sub
```

```
Public Function Delay(ByVal ms As Long)
```

```
    Dim i As Integer
```

```
    Do Until i > Val(100 * ms)
```

```
        i = i + 1
```

```
        Loop
    End Function

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Timer1.Tick

        y = sp.ReadExisting

        If (y <> "") Then

            TextBox1.Text = y + vbCrLf + TextBox1.Text

        End If

    End Sub

End Class
```

(二)通讯协议

1. 概述

考虑到输送带控制模块的核心技术是与 PLC 的串口通讯，在 Visual Studio 6.0 中编写串口通讯程序，一般都使用 Microsoft Communication Control（简称 MSComm）的通讯控件，本小组通过对此控件的属性和事件进行相应编程操作，就可以轻松地实现串口通讯。

2. 通讯串口参数

串口通讯最常用的参数就是通讯端口号及通讯格式(波特率、数据位、停止位和校验位)，在 MSComm 中相关的属性是 CommPort 和

Settings。SerialPort 类与 MSComm 有一些区别:

- 通讯端口:

[PortName]属性获取或设置通信端口，包括但不限于所有可用的 COM 端口，请注意该属性返回类型为 String，不是 Mscomm.CommPort 的 short 类型。通常情况下，PortName 正常返回的值为 COM1、COM2……，SerialPort 类最大支持的端口数突破了 CommPort 控件中 CommPort 属性不能超过 16 的限止，大大方便了用户串口设备的配置。

- 通讯格式:

SerialPort 类对分别用[BaudRate]、[Parity] 、 [DataBits]、[StopBits] 属性设置通讯格式中的波特率、校验位、数据位和停止位，其中[Parity]和[StopBits]分别是枚举类型 Parity、StopBits，Parity 类型中枚举了 Odd(奇)、Even(偶)、Mark、None、Space，Parity 枚举了 None、One、OnePointFive、Two。 SerialPort 类提供了七个重载的构造函数，既可以对已经实例化的 SerialPort 对象设置上述相关属性的值，也可以使用指定的端口名称、波特率和奇偶校验位数据位和停止位直接初始化 SerialPort 类的新实例。

- 串口的打开关闭

SerialPort 类没有采用 MSComm.PortOpen=True/False 设置属性值打开关闭串口，相应的是调用类的 Open()和 Close()方法。

- 数据的发送读取

SerialPort 类调用重载的 Write 和 WriteLine 方法发送数据，其中

WriteLine 可发送字符串并在字符串末尾加入换行符，读取串口缓冲区的方法有许多，其中除了 **ReadExisting**（读取 **SerialPort** 对象的流和输入缓冲区中所有立即可用的字节）和 **ReadTo**（一直读取到输入缓冲区中的指定 **value** 的字符串），其余的方法都是同步调用，线程被阻塞直到缓冲区有相应的数据或大于 **ReadTimeout** 属性设定的时间值后，引发 **ReadExisting** 异常。

(三)下位机软件设计

```
int cp = 9;

int cw1 = 5;

int cw2 = 6;

int delaytime=4000;

int interval=100;

int State='2';

int val='2';

void setup()

{

    pinMode(cp,OUTPUT);

    pinMode(cw1,OUTPUT);

    pinMode(cw2,OUTPUT);

    Serial.begin(9600);

}
```

```
void ForwardInit()           //forward
{
    digitalWrite(cp,LOW);
    digitalWrite(cw2,LOW);
    digitalWrite(cw1,HIGH);
}

void BackwardInit()          //backward
{
    digitalWrite(cp,LOW);
    digitalWrite(cw1,LOW);
    digitalWrite(cw2,HIGH);
}

void Stop()                  //backward
{
    digitalWrite(cp,LOW);
    digitalWrite(cw1,LOW);
    digitalWrite(cw2,LOW);
}

void loop()
{
    while(Serial.available())
    {
```

```
    val=Serial.read();

    delay(2);

    if(val=='0')

    {

        Serial.println("Forward");

    }

    else if(val=='1')

    {

        Serial.println("Backward");

    }

    else if(val=='2')

    {

        Serial.println("Stop");

    }

    else if(val=='9')

    {

        Serial.println("Reverse");

    }

    else if(val=='+')

    {

        Serial.println("Accelerate");

    }
```

```
else if(val=='-')
{
    Serial.println("Decelerate");
}
else
{
    Serial.println("Invalid Instruction!");
}
}

if(val=='0')                                //forward
{
    State=val;
    ForwardInit();
    digitalWrite(cp,LOW);
    delayMicroseconds(delaytime);
    digitalWrite(cp,HIGH);
    delayMicroseconds(delaytime);
}

if(val=='1')                                //backward
{
    State=val;
    BackwardInit();
```

```
    digitalWrite(cp,LOW);

    delayMicroseconds(delaytime);

    digitalWrite(cp,HIGH);

    delayMicroseconds(delaytime);

}

if(val=='2')          //stop
{
    State=val;

    Stop();
}

if(val=='9')          //reverse
{
    if(State=='1')
    {
        val='0';

        ForwardInit();

        digitalWrite(cp,LOW);

        delayMicroseconds(delaytime);

        digitalWrite(cp,HIGH);

        delayMicroseconds(delaytime);

    }

    if(State=='0')
```

```
{  
    val='1';  
    BackwardInit();  
    digitalWrite(cp,LOW);  
    delayMicroseconds(delaytime);  
    digitalWrite(cp,HIGH);  
    delayMicroseconds(delaytime);  
}  
}  
  
if (val=='+')                //accelerate  
{  
    val=State;  
    delaytime=delaytime-interval;  
}  
  
if (val=='-')                //decelerate  
{  
    val=State;  
    delaytime=delaytime+interval;  
}  
}
```


四、实物连接

