浙江理工大学

机械与自动控制学院

计算机控制系统与接口技术实验报告

专业： 机械工程

班级： 18级（4）班

姓名： 栗硕

学号： 2018G0505021

题目

1. 编写Arduino/单片机程序，实现以下功能：
   1. Arduino上运行PID控制算法，控制PC中的虚拟模型
   2. 开发PC中的虚拟模型，可以为电机，或者是加热器等，自定。
   3. 在上位机上绘制波形

配置环境

本次报告采用的下位机是Arduino UNO单片机，如图1所示：

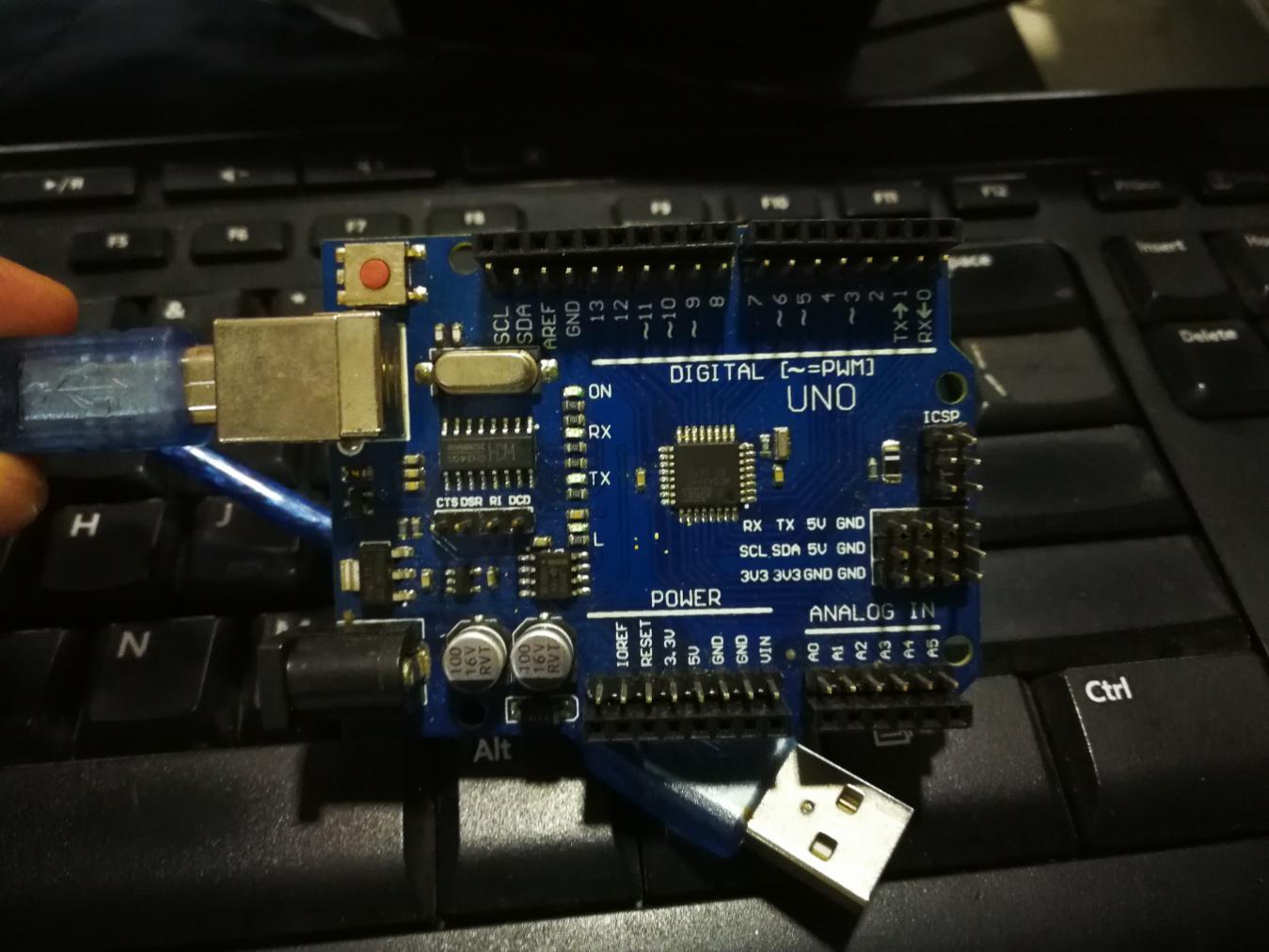


图1

PC端的虚拟模型是采用C#语言编写，使用Visual Studio 2012进行编译，如图2所示：

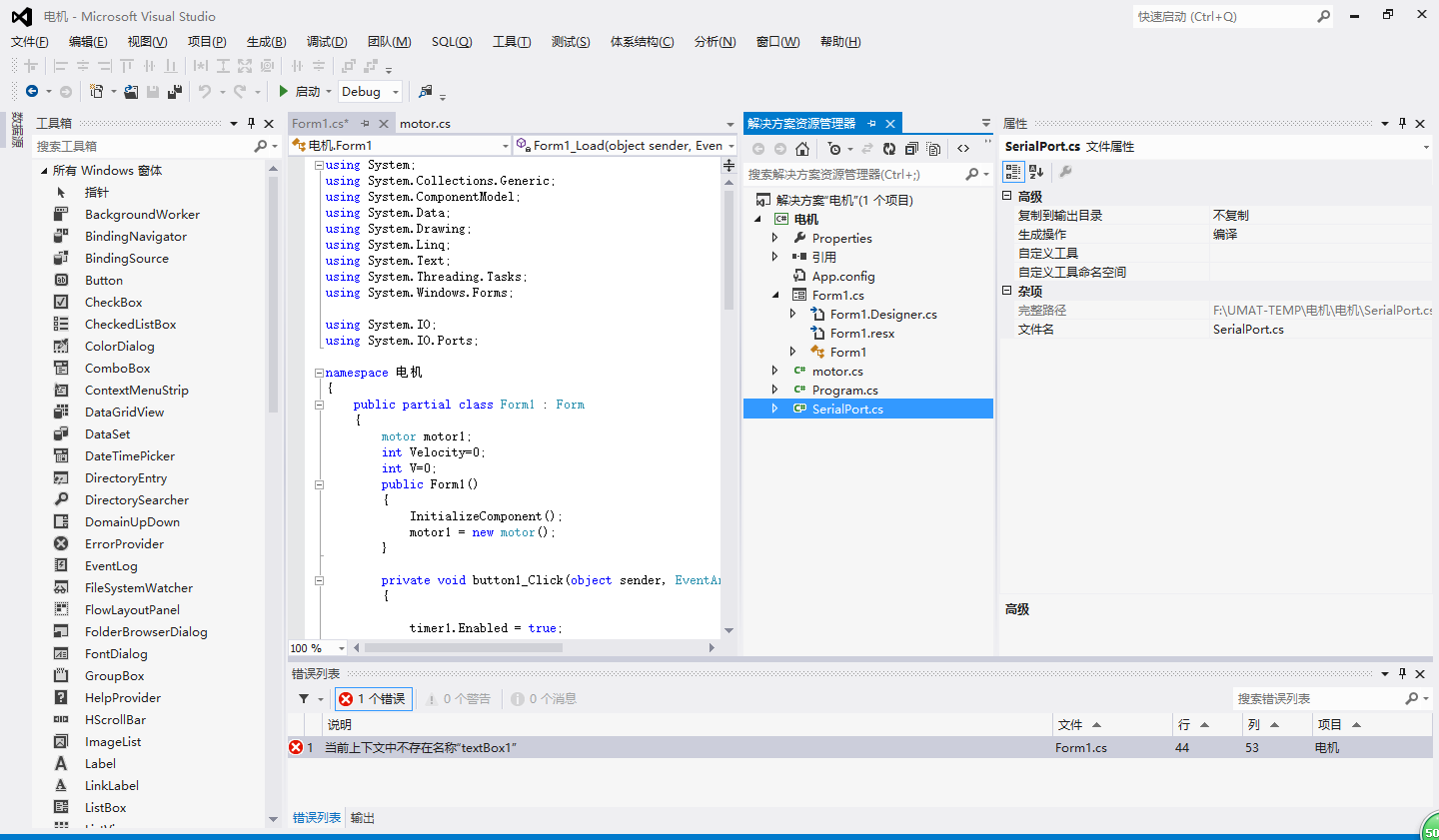


图2

调试过程中使用到串口调试助手，如图3所示：

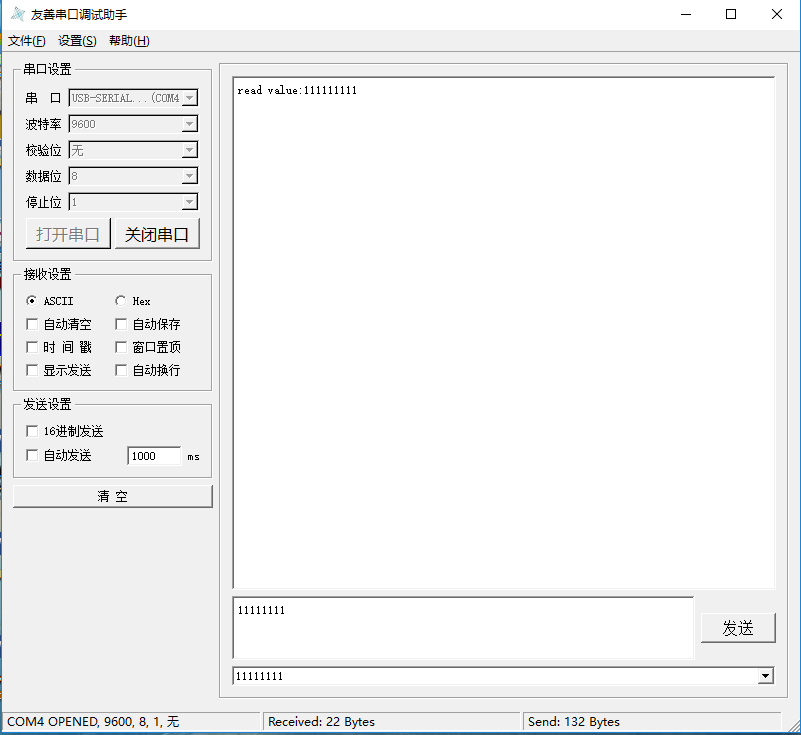


图3

Arduino上运行PID控制算法

PID算法，是一种在工程应用领域被使用最为广泛的负反馈调节方法，通过PID算法中比例、积分、微分三个部分的作用，达到使系统稳定的效果。本报告采用的是Arduino UNO单片机实现。

单片机控制程序分为三部分，即void setup() {}和void loop() {}以及定义变量。定义变量及void setup() {}都属于初始化部分，具体程序及注释如下：

char flag\_run; //电机运行标志

int Velocity\_goal = 2500; //目标速度

int PID\_P = 2, PID\_I = 1, PID\_D = 0, T = 1; //PID参数

int Velocity\_Integral = 0,last\_Velocity\_bias = 0,Velocity\_bias = 0,Velocity\_PWM = 0,Velocity\_Least = 0;

int offset; //补偿值

String comdata = ""; //存放串口读取的值

char temp\_char=0;

bool Find\_head = false;

void setup()

{

Serial.begin(9600); //定义串口通讯波特率为9600 bit/s

}

循环的主函数void loop() {}部分：

void loop() {

while (Serial.available() > 0)

{

temp\_char=char(Serial.read());

if(temp\_char=='\n') //发现包头

{

Find\_head=~Find\_head;

}

if(Find\_head==true)

{

comdata += temp\_char; //读取数据

delay(2);

}

if(comdata!= NULL)

{

char\* end;

Velocity\_Least = static\_cast<int>(strtol(comdata.c\_str(),&end,10));

comdata ="" ; //将读入的字符串转化为整形数值

}

}

last\_Velocity\_bias = Velocity\_bias; //保存上一次的误差值

Velocity\_bias = Velocity\_goal - Velocity\_Least;

Velocity\_Integral += Velocity\_bias; //===积分出位移

if (Velocity\_Integral > 3600000) Velocity\_Integral = 360000; //===积分限幅

if (Velocity\_Integral < -3600000) Velocity\_Integral = -360000; //===积分限幅

offset = Velocity\_bias \* PID\_P + Velocity\_Integral \* PID\_I + (Velocity\_bias - last\_Velocity\_bias) \* PID\_D; //===速度PID控制器

if (flag\_run == 0)

{

Velocity\_Integral = 0; //===电机关闭后清除积分

Velocity\_PWM = 0;

}

Serial.println('V'+offset);

delay(500);

}

C#编写的虚拟模型

本报告使用C#编写一个电机模型，包含启动和停止两个控制命令，界面如图4所示：

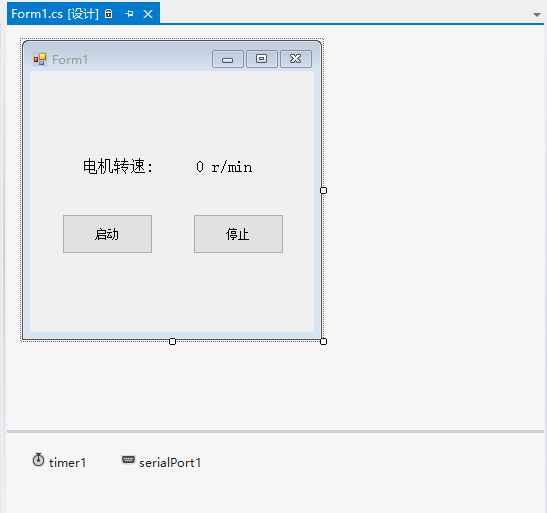


图4

整个模型的具体代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.IO.Ports;

namespace 电机

{

public partial class Form1 : Form

{

motor motor1;

int Velocity=0;

int V=0;

public Form1()

{

InitializeComponent();

motor1 = new motor();

}

private void button1\_Click(object sender, EventArgs e)

{

timer1.Enabled = true;

}

private int Com\_to\_UNO(int Velocity)

{

serialPort1.PortName = "COM4";

serialPort1.BaudRate = 9600;

serialPort1.Parity = Parity.Odd;

serialPort1.StopBits = StopBits.Two;

serialPort1.ReadTimeout = 1000;

if (serialPort1.IsOpen == false)

serialPort1.Open();

byte[] data = Encoding.Unicode.GetBytes(Velocity);

string V = Convert.ToBase64String(data);

serialPort1.WriteLine(V);

MessageBox.Show("数据发送成功！", "系统提示");

return V;

}

private void timer1\_Tick(object sender, EventArgs e)

{

V = Com\_to\_UNO(Velocity);

Velocity = motor1.motor\_run(V);

label2.Text = Velocity.ToString() + "r/min";

}

private void button2\_Click(object sender, EventArgs e)

{

timer1.Enabled = false;

motor1.motor\_stop();

}

}

}

电机的具体定义函数为：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace 电机

{

static class Program

{

/// <summary>

/// 应用程序的主入口点。

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

模型的运行情况如下图5：

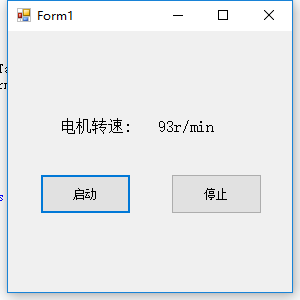


图5

串口通讯

单片机Arduino UNO在与虚拟模型通讯时，需要将串口中的字符串转换为整形数进行读取，先控制虚拟模型向单片机输出值为一定值，调试串口，可以发现com4端口能不停地读取出端口中的值，如图6所示：

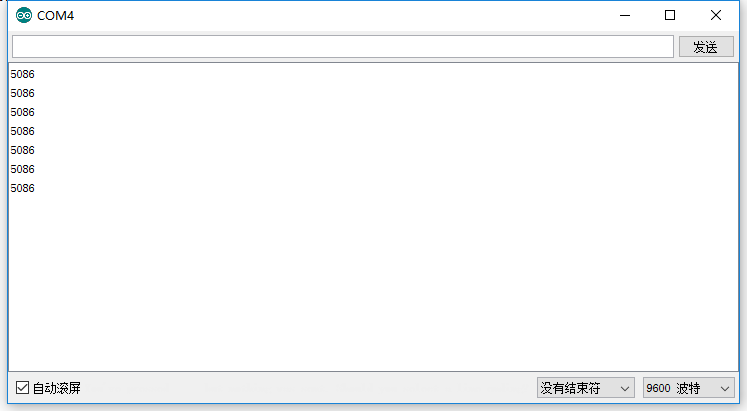


图6

在虚拟模型与单片通讯时，使用C#自带的SerialPor控件。在“工具箱”的“组件”中选择SerialPort控件添加。设置串口并打开

serialPort1.PortName = "COM4";

serialPort1.BaudRate = 9600;

serialPort1.Open();

写入数据时可以使用Write或者函数serialPort1.WriteLine(str);

添加数据接收的事件使用private void serialPort1\_DataReceived (object sender, SerialDataReceivedEventArgs e)

读取串口数据，如下图7：

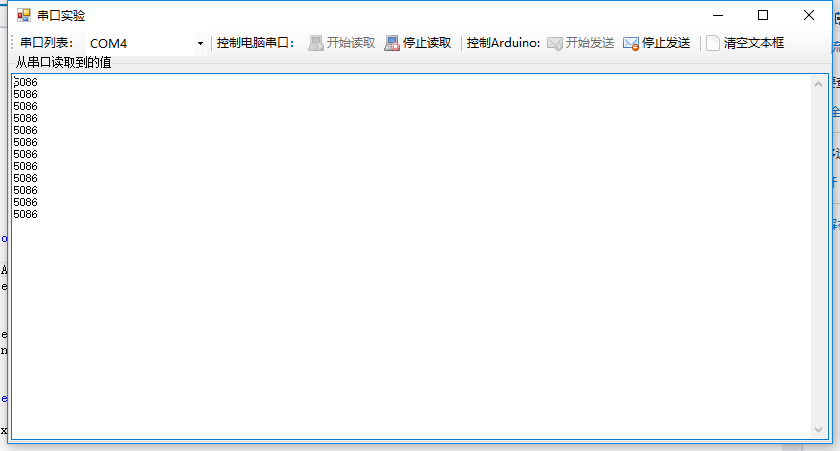


图7

输出结果

通过调试，得到以下输出曲线图8：

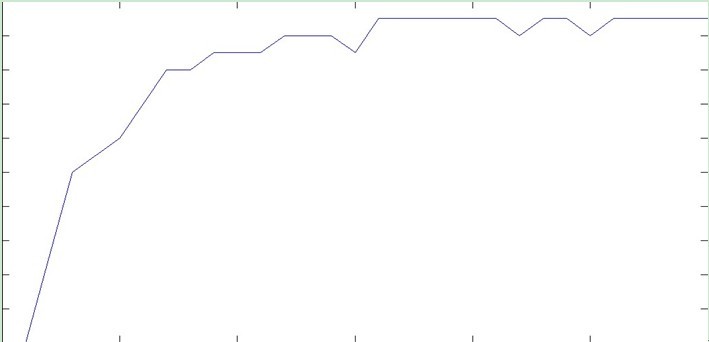


图8