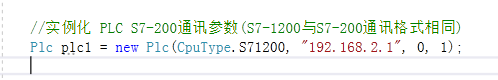
引用 S7.NET

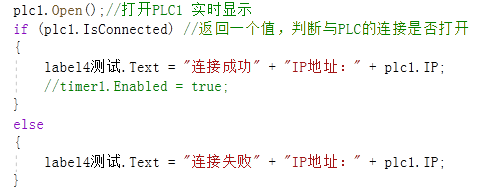


实例化 PLC S7-200通讯参数(S7-1200与S7-200通讯格式相同)



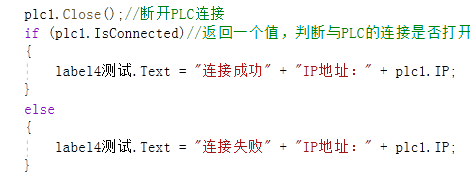
打开PLC1

返回一个值，判断与PLC的连接是否打开

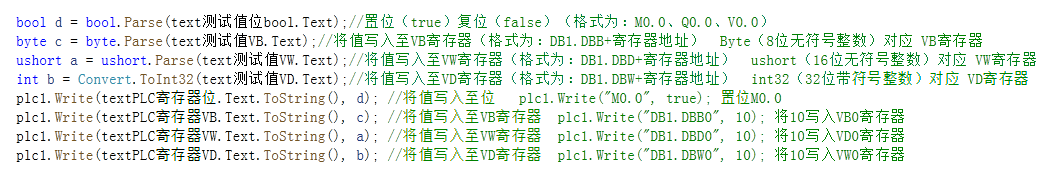


断开PLC1

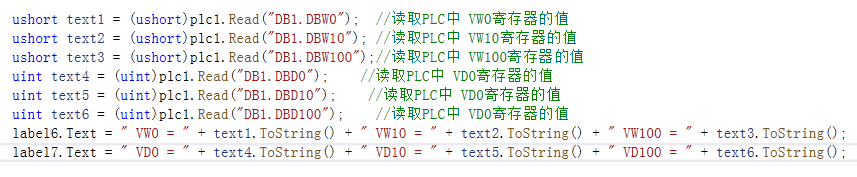
返回一个值，判断与PLC的连接是否打开



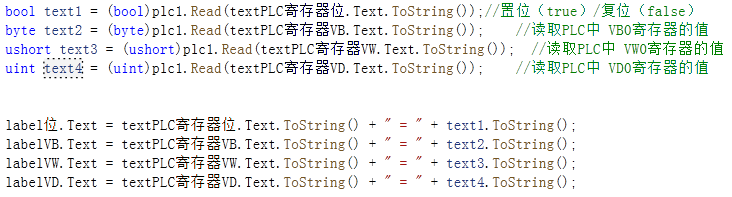
将值写入至PLC寄存器

、

读取PLC中的值



/////////



界面：



源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using S7.Net; //引用S7.NET

namespace 西门子PLC通讯测试

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

text端口.Enabled = false;

textIP.Enabled = false;

timer1.Enabled = false;

//Plc plc1 = new Plc(CpuType.S71200, string.Concat(textIP.Text), 0, 1);//实例化 PLC S7-200通讯参数

//Plc plc1 = new Plc(CpuType.S7200Smart, "192.168.2.1", 0, 0);//实例化 PLC S7-200通讯参数Plc plc1 = new Plc(CpuType.S71200, string.Concat(textIP.Text), 0, 1);//实例化 PLC S7-200通讯参数

}

//实例化 PLC S7-200通讯参数(S7-1200与S7-200通讯格式相同)

Plc plc1 = new Plc(CpuType.S71200, "192.168.2.1", 0, 1);

private void But连接\_Click(object sender, EventArgs e)

{

try

{

plc1.Open();//打开PLC1

if (plc1.IsConnected) //返回一个值，判断与PLC的连接是否打开

{

label4测试.Text = "连接成功" + "IP地址：" + plc1.IP;

//timer1.Enabled = true;

}

else

{

label4测试.Text = "连接失败" + "IP地址：" + plc1.IP;

}

//label4测试.Text = "正在连接" + "IP地址：" + plc1.IP;//显示PLC1的 IP 及 端口

//text测试值.Text = "正在连接" + "IP地址："+plc1.IP + "端口号:" + plc1.Port;//显示PLC1的 IP 及 端口

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void But断开连接\_Click(object sender, EventArgs e)

{

try

{

//Plc plc1 = new Plc(CpuType.S71200, string.Concat(textIP.Text), 0, 1);//实例化 PLC S7-200通讯参数

plc1.Close();//断开PLC连接

if (plc1.IsConnected)//返回一个值，判断与PLC的连接是否打开

{

label4测试.Text = "连接成功" + "IP地址：" + plc1.IP;

}

else

{

label4测试.Text = "连接失败" + "IP地址：" + plc1.IP;

}

timer1.Enabled = false;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void But写入位Bool\_Click(object sender, EventArgs e)

{

try

{

bool d = bool.Parse(text测试值位bool.Text);//置位（true）复位（false）（格式为：M0.0、Q0.0、V0.0）

//byte c = byte.Parse(text测试值VB.Text);//将值写入至VB寄存器（格式为：DB1.DBB+寄存器地址） Byte（8位无符号整数）对应 VB寄存器

//ushort a = ushort.Parse(text测试值VW.Text);//将值写入至VW寄存器（格式为：DB1.DBD+寄存器地址） ushort（16位无符号整数）对应 VW寄存器

//int b = Convert.ToInt32(text测试值VD.Text);//将值写入至VD寄存器（格式为：DB1.DBW+寄存器地址） int32（32位带符号整数）对应 VD寄存器

plc1.Write(textPLC寄存器位.Text.ToString(), d); //将值写入至位 plc1.Write("M0.0", true); 置位M0.0

//plc1.Write(textPLC寄存器VB.Text.ToString(), c); //将值写入至VB寄存器 plc1.Write("DB1.DBB0", 10); 将10写入VB0寄存器

//plc1.Write(textPLC寄存器VW.Text.ToString(), a); //将值写入至VW寄存器 plc1.Write("DB1.DBD0", 10); 将10写入VD0寄存器

//plc1.Write(textPLC寄存器VD.Text.ToString(), b); //将值写入至VD寄存器 plc1.Write("DB1.DBW0", 10); 将10写入VW0寄存器

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void But写入VB\_Click(object sender, EventArgs e)

{

try

{

//bool d = bool.Parse(text测试值位bool.Text);//置位（true）复位（false）（格式为：M0.0、Q0.0、V0.0）

byte c = byte.Parse(text测试值VB.Text);//将值写入至VB寄存器（格式为：DB1.DBB+寄存器地址） Byte（8位无符号整数）对应 VB寄存器

//ushort a = ushort.Parse(text测试值VW.Text);//将值写入至VW寄存器（格式为：DB1.DBD+寄存器地址） ushort（16位无符号整数）对应 VW寄存器

//int b = Convert.ToInt32(text测试值VD.Text);//将值写入至VD寄存器（格式为：DB1.DBW+寄存器地址） int32（32位带符号整数）对应 VD寄存器

//plc1.Write(textPLC寄存器位.Text.ToString(), d); //将值写入至位 plc1.Write("M0.0", true); 置位M0.0

plc1.Write(textPLC寄存器VB.Text.ToString(), c); //将值写入至VB寄存器 plc1.Write("DB1.DBB0", 10); 将10写入VB0寄存器

//plc1.Write(textPLC寄存器VW.Text.ToString(), a); //将值写入至VW寄存器 plc1.Write("DB1.DBD0", 10); 将10写入VD0寄存器

//plc1.Write(textPLC寄存器VD.Text.ToString(), b); //将值写入至VD寄存器 plc1.Write("DB1.DBW0", 10); 将10写入VW0寄存器

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Button1\_Click(object sender, EventArgs e)

{

try

{

//bool d = bool.Parse(text测试值位bool.Text);//置位（true）复位（false）（格式为：M0.0、Q0.0、V0.0）

//byte c = byte.Parse(text测试值VB.Text);//将值写入至VB寄存器（格式为：DB1.DBB+寄存器地址） Byte（8位无符号整数）对应 VB寄存器

ushort a = ushort.Parse(text测试值VW.Text);//将值写入至VW寄存器（格式为：DB1.DBD+寄存器地址） ushort（16位无符号整数）对应 VW寄存器

//int b = Convert.ToInt32(text测试值VD.Text);//将值写入至VD寄存器（格式为：DB1.DBW+寄存器地址） int32（32位带符号整数）对应 VD寄存器

//plc1.Write(textPLC寄存器位.Text.ToString(), d); //将值写入至位 plc1.Write("M0.0", true); 置位M0.0

//plc1.Write(textPLC寄存器VB.Text.ToString(), c); //将值写入至VB寄存器 plc1.Write("DB1.DBB0", 10); 将10写入VB0寄存器

plc1.Write(textPLC寄存器VW.Text.ToString(), a); //将值写入至VW寄存器 plc1.Write("DB1.DBD0", 10); 将10写入VD0寄存器

//plc1.Write(textPLC寄存器VD.Text.ToString(), b); //将值写入至VD寄存器 plc1.Write("DB1.DBW0", 10); 将10写入VW0寄存器

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void But写入VD1\_Click(object sender, EventArgs e)

{

try

{

//bool d = bool.Parse(text测试值位bool.Text);//置位（true）复位（false）（格式为：M0.0、Q0.0、V0.0）

//byte c = byte.Parse(text测试值VB.Text);//将值写入至VB寄存器（格式为：DB1.DBB+寄存器地址） Byte（8位无符号整数）对应 VB寄存器

//ushort a = ushort.Parse(text测试值VW.Text);//将值写入至VW寄存器（格式为：DB1.DBD+寄存器地址） ushort（16位无符号整数）对应 VW寄存器

int b = Convert.ToInt32(text测试值VD.Text);//将值写入至VD寄存器（格式为：DB1.DBW+寄存器地址） int32（32位带符号整数）对应 VD寄存器

//plc1.Write(textPLC寄存器位.Text.ToString(), d); //将值写入至位 plc1.Write("M0.0", true); 置位M0.0

//plc1.Write(textPLC寄存器VB.Text.ToString(), c); //将值写入至VB寄存器 plc1.Write("DB1.DBB0", 10); 将10写入VB0寄存器

//plc1.Write(textPLC寄存器VW.Text.ToString(), a); //将值写入至VW寄存器 plc1.Write("DB1.DBD0", 10); 将10写入VD0寄存器

plc1.Write(textPLC寄存器VD.Text.ToString(), b); //将值写入至VD寄存器 plc1.Write("DB1.DBW0", 10); 将10写入VW0寄存器

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void But读取\_Click(object sender, EventArgs e)

{

try

{

//Plc plc1 = new Plc(CpuType.S71200, string.Concat(textIP.Text), 0, 1);//实例化 PLC S7-200通讯参数

//plc1.Open();//打开PLC1 实时显示

bool text1 = (bool)plc1.Read(textPLC寄存器位.Text.ToString());//置位（true）/复位（false）

byte text2 = (byte)plc1.Read(textPLC寄存器VB.Text.ToString()); //读取PLC中 VB0寄存器的值

ushort text3 = (ushort)plc1.Read(textPLC寄存器VW.Text.ToString()); //读取PLC中 VW0寄存器的值

uint text4 = (uint)plc1.Read(textPLC寄存器VD.Text.ToString()); //读取PLC中 VD0寄存器的值

label位.Text = textPLC寄存器位.Text.ToString() + " = " + text1.ToString();

labelVB.Text = textPLC寄存器VB.Text.ToString() + " = " + text2.ToString();

labelVW.Text = textPLC寄存器VW.Text.ToString() + " = " + text3.ToString();

labelVD.Text = textPLC寄存器VD.Text.ToString() + " = " + text4.ToString();

//timer1.Enabled = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Timer1\_Tick(object sender, EventArgs e)

{

try

{

//ushort a;

//a = 0;

//a = a++;

//a = ushort.Parse(text测试值.Text);

//plc1.Write("DB1.DBW0", a);

//ushort text3 = (ushort)plc1.Read("DB1.DBW0");

//label6.Text = text3.ToString();

bool text1 = (bool)plc1.Read(textPLC寄存器位.Text.ToString());//置位（true）/复位（false）

byte text2 = (byte)plc1.Read(textPLC寄存器VB.Text.ToString()); //读取PLC中 VB0寄存器的值

ushort text3 = (ushort)plc1.Read(textPLC寄存器VD.Text.ToString()); //读取PLC中 VW0寄存器的值

uint text4 = (uint)plc1.Read(textPLC寄存器VW.Text.ToString()); //读取PLC中 VD0寄存器的值

label位.Text = textPLC寄存器位.Text.ToString() + " = " + text1.ToString();

labelVW.Text = textPLC寄存器VB.Text.ToString() + " = " + text2.ToString();

labelVD.Text = textPLC寄存器VD.Text.ToString() + " = " + text3.ToString();

labelVB.Text = textPLC寄存器VW.Text.ToString() + " = " + text4.ToString();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}