Click to prove you're human

百度知道>提示信息 知道宝贝找不到问题了> _ byte //word => ushort //DWord => uint //Int => short //Dint => int //Real => float //LReal => double //String => //DateTime //s7wstring=>string 数据类涉及到最后的转换。 需要留意 以上一篇文章博途数据块为例： 点击转至在线》点击第二个点。 监视数据块中的值的变化 图中偏移量为所在数据块的位置 ，与下一个数据相隔的偏移量为占用的字节 以第一个bool值举例： var a = plc.Read("DB1.DBX0.0");//会重复发送tcp请求 ，效率较低 ，非特殊情况不建议使用 plc.Write("DB1.DBX0.0", true);//与以上同理 //read方法的参数为 ，数据库类型默认是数据库 : DataBlock。 第几个数据库 ，偏移量 ，数据类型 ，读取多少位 可以看到a值读取到设置的初始值false 在第2步中修改了bool值之后 ，博途软件中监视重新建立一个tcp连接 ，非常消耗资源。 在实际开发中不建议使用 可以用以下方案 : int db = 1; var boolDemo = plc.Read("DB1.DBX0.0"); //会重复发送tcp请求 ，效率较低 ，非特殊情况不建议使用 plc.Write("DB1.DBX0.0", true);//与以上同理 //read方法的参数为 ，数据库类型默认是数据库 : DataBlock。 第几个数据库 ，偏移量 ，数据类型 ，读取多少位 可以看到a值读取到设置的初始值false 在第2步中修改了bool值之后 ，博途软件中监视重新建立一个tcp连接 ，非常消耗资源。 在实际开发中不建议使用 可以用以下方案 : int db = 1; var boolDemo = plc.Read("DB1.DBX0.0"); //会重复发送tcp请求 ，效率较低 ，非特殊情况不建议使用 plc.Write("DB1.DBX0.0", true);//与以上同理 //read方法的参数为 ，数据库类型默认是数据库 : DataBlock。 第几个数据库 ，偏移量 ，数据类型 ，读取多少位 可以看到 bollDemoRead = plc.Read(DataType.DataBlock, db, 0, VarType.Bit, 1); Console.WriteLine("修改后bool值打印 : " + bollDemoRead); var boolDemoRead = plc.Read(DataType.DataBlock, db, 2, VarType.Int, 1); Console.WriteLine("word值打印 : " + wordDemoRead); var wordDemoRead = plc.Read(DataType.DataBlock, db, 2, VarType.Int, 1); Console.WriteLine("修改后 + ""，0, 1); plc.Open(); if (plc.IsConnected) { Console.WriteLine("PLC连接成功"); } var a = plc.Read("DB1.DBX0.0", true); int db = 1; var boolDemo = plc.Read("DB1.DBX0.0"); plc.Write("DB1.DBX0.0", false); var boolDemoRead = VarType.Int, 1); Console.WriteLine("word值打印 : " + wordDemoRead); var wordDemo = plc.Read(DataType.DataBlock, db, 4, 8); var DwordDemoRead = plc.Read(DataType.DataBlock, db, 4, VarType.Int, 1); Console.WriteLine("Dword值打印 : " + DwordDemo); plc.Write(DataType.DataBlock, db, 4, 8); var DwordDemoRead = plc.Read(DataType.DataBlock, db, 4, VarType.Int, 1); Console.WriteLine("修改后Dword值打印 : " + DwordDemoRead); var intDemo = plc.Read(DataType.DataBlock, db, 8, VarType.Int, 1); Console.WriteLine("Int值打印 : " + intDemo); short readint = 9; plc.Write(DataType.DataBlock, db, 8, readint); var intDemoRead = plc.Read(DataType.DataBlock, db, startByteAdr: 8, VarType.Int, 1); Console.WriteLine("修改后Int值打印 : " + intDemoRead); var DintDemo = plc.Read(DataType.DataBlock, db, 10, VarType.Int, 1); Console.WriteLine("Dint值打印 : " + DintDemo); plc.Write(DataType.DataBlock, db, 10, 100); var DintDemoRead = plc.Read(DataType.DataBlock, db, 10, VarType.Int, 1); Console.WriteLine("修改后Dint值打印 : " + DintDemoRead); var RealDemo = plc.Read(DataType.DataBlock, db, 14, VarType.Int, 1); Console.WriteLine("Real值打印 : " + RealDemo); plc.Write(DataType.DataBlock, db, 14, (float)99.9); var DrealDemoRead = plc.Read(DataType.DataBlock, db, 14, VarType.Int, 1); Console.WriteLine("修改后Real值打印 : " + DrealDemoRead); var LRealDemo = plc.Read(DataType.DataBlock, db, 18, VarType.Int, 1); Console.WriteLine("LReal值打印 : " + LRealDemo); plc.Write(DataType.DataBlock, db, 18, 88.88); var LrealDemoRead = plc.Read(DataType.DataBlock, db, 18, VarType.Int, 1); Console.WriteLine("修改后Lreal值打印 : " + LrealDemoRead); var bytedemo = plc.Read(DataType.DataBlock, db, 290, VarType.Int, 1); Console.WriteLine("修改后bytedemo值打印 : " + bytedemo); plc.Write(DataType.DataBlock, db, 290, (byte)2); var bytedemoRead = plc.Read(DataType.DataBlock, db, 290, VarType.Byte, 1); Console.WriteLine("修改后byte值打印 : " + bytedemoRead); var datedemo = plc.Read(DataType.DataBlock, db, 282, VarType.DateTime, 1); Console.WriteLine("date值打印 : " + datedemo); plc.Write(DataType.DataBlock, 2, 282, System.DateTime.Now); var datedemoRead = plc.Read(DataType.DataBlock, db, 282, VarType.DateTime, 1); Console.WriteLine("修改后date值打印 : " + datedemoRead); var charDemo = plc.Read(DataType.DataBlock, db, 4, VarType.String, 1); Console.WriteLine("char值打印 : " + charDemo); plc.Write(DataType.DataBlock, db, 4, "a"); var charDemoRead = plc.Read(DataType.DataBlock, db, 4, VarType.Int, 1); Console.WriteLine("修改后char值打印 : " + charDemoRead); 比较特殊的 : string : 需要先获取string值的所占长度。 再拿到具体bytelen。 转换为utf8格式的bascii码 具体代码中有体现 +2表示获取到长度 +1表示获取到跳过偏移长度的字符 string类型只能存储ascci码 , 需要注意 , 不能存储中文 #pragma warning disable var count = (byte)plc.Read(DataType.DataBlock, db, 26 + 1, VarType.Byte, 1); byte[] stringDemo = plc.ReadBytes(DataType.DataBlock, db, 26 + 2, count); string stringValue = Encoding.Default.GetString(stringDemo); Console.WriteLine("string值打印 : " + stringValue); plc.WriteDataType.DataBlock, db, 26 + 1, (byte)"nihao".Length); plc.WriteDataType.DataBlock, db, 26 + 2, "nihao"); stringDemo = plc.ReadBytes(DataType.DataBlock, db, 26 + 2, count); string stringDemoRead = Encoding.Default.GetString(stringDemo); Console.WriteLine("修改后string值打印 : " + stringDemoRead); wstring : wstring类型 , 需要注意的是获取的字符需要为254个。 因为符号占用了4个字节 var wstringDemo = plc.Read(DataType.DataBlock, db, startByteAdr: 292, VarType.S7WString, varCount: 254); array : array区别于其他方法的原因是 : 在读取时需要进行转换 , 例如使用float[] 将博途的real类型转换为float数组类型。 在读取时count值也需要和plc中设置的长度一致 在写入时 , 可以在读取的值上做修改如 : arrydemo。 也可以自定义一个新数组。 注意新数组的长度可以少于plc中设定的长度。 那么长度只能的会写入生效。 例如plc中设置长度为2 , 我写入的长度只有1 , 那么数组第一个值会写入 , plc中第一个值则不变。 如果新数组的长度大于plc中设置的长度 , 则抛出异常 具体可以看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(DataType.DataBlock, db, 804 VarType.Real, 10);//array转为特殊 , 他设定的范围值可能有多种类型 , real , int , word , bool等 , 需要根据不同类型做处理 Console.WriteLine("array值打印 : " + SystemString.Join(",", arrayDemo)); //写入array并打印 for (int i = 0; i < arrayDemo.Length; i++) { arrayDemo[i] += 0.1F; } plc.Write(DataType.DataBlock, db, 804, arrayDemo);//在原有的基础上修改 float[] arrayDemo2 = { 0.11F, 0.22F };//如果只改变其中一个帮助方法修改传值的array plc.Write(DataType.DataBlock, db, 804, arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印 : " + SystemString.Join(",", arrayDemoRead)); 实现了一个帮助类 , 导入即用 : 资源链接： 文章来源： 知识共享原创文章 , 遵循CC 4.0 BY-SA 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不帮助类 , 导入即用 : 资源链接： 文章来源： 知识共享原创文章 , 遵循CC 4.0 BY-SA 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不声明。 百度知道>提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不帮助类 , 导入即用 : 资源链接： 文章来源： 知识共享原创文章 , 遵循CC 4.0 BY-SA 知道宝贝找不到问题了> _ 提示信息 知道宝贝找不声明。