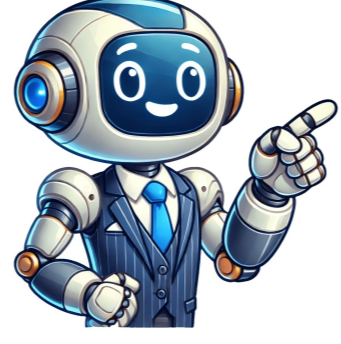


Click to prove  
you're human





























```

scoredhigh 2015-04-13 04:40:01 为什么System.Diagnostics.Debug.WriteLine不能在输出窗口显示了？本人有个项目，以前一直都是在输出窗口显示调试信息的，突然有一天就显示不出了，可能我做了什么操作，但是实在想不起来，后来无论怎么设置，System.Diagnostics.Debug.WriteLine都不能在输出窗口显示了...全文
xiaowulai225 2016-04-07 07:31:19 如题，c写的一个socket网关服务器和内网的一个逻辑服进行通讯。系统版本是centos6.5_x64 观察下来如果经常性的有客户端和网关交互（网关把客户端消息传给内网服务器，或是内网服通过网关发送消息给客户端），无论消息量的大小，网关服和内网服的socket连接在read或是write时 经常性会出现 errno=5的错误，观察下来基本2~3天就会出现一次。请问下这个错误码表示什么意思？大概是什么原因引起的 因为观察到如果几天没有任何客户端消息的情况下网关基本不会出现这个错误的。感觉上应该和代码逻辑有点关系。在论坛里搜到个类似的问题，有人回答可能是因为网卡等设备引起的。求救大神！万分感谢！...全文
Ma新新赫赫赫 2019-07-22 10:50:36 ...全文
百度知道>提示您:问题宝贝找不到问题!> byte /word => ushort /DWord => uint /Int => short /DInt => int /Real => float /LReal => double /SString => string /Date/TimeLong=>datetime //&vstring->string 数据涉及到底层的转换。需要留意 以上一篇文档将数据类型转换为二进制格式的数据块为例：点击左在在线 点击第二个点。监视数据块中值的变化 图中偏移量为所存放数据块的位置。与下一个数据块间的偏移量为占用的字节 以一个bool值为例： var a = plc.Read("D1.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+是数据库。DataBlock。第几个数据块。偏移量。数据类型。读取多少位 可以看到a读取取设置的初始值为false 在第二步中修改了bool值之后，精途软件中监视值变成了true 以上代码读取一次会重新建立一个tcp连接。非常消耗资源。在实际开发中不建议使用 可以以下方案： int db = 1; var boolDemo = plc.Read(DataType.DataBlock, db, 0, VarType.Bit, 1); Console.WriteLine("bool值打印："+ boolDemo); plc.Write(DataType.DataBlock, db, 0, false); var boolDemoRead = plc.Read(DataType.DataBlock, db, 0, VarType.Bit, 1); Console.WriteLine("修改后bool值打印："+ boolDemoRead); 在途中第一个参数，标识数据块类型。第二个标识数据块。我们是db1，所以写：1 偏移量为0 类型为bit 获取长度（除了字符类型和数组其他都默认1）以下代码是其他值类型的示例： Plc plc = new Plc(CpuType.S71500, "192.168.43.14" + "", 0, 1); plc.Open(); if (plc.IsConnected) { Console.WriteLine("PLC连接成功"); } var a = plc.Read("DB1.DBX0.0"); plc.Write("DB1.DBX0.0", true); int db = 1; var boolDemo = plc.Read(DataType.DataBlock, db, 0, VarType.Bit, 1); Console.WriteLine("bool值打印："+ boolDemo); plc.Write(DataType.DataBlock, db, 0, false); var boolDemoRead = plc.Read(DataType.DataBlock, db, 0, VarType.Bit, 1); Console.WriteLine("修改后bool值打印："+ boolDemoRead); var wordDemo = plc.Read(DataType.DataBlock, db, 2, VarType.Int, 1); Console.WriteLine("word值打印："+ wordDemo); plc.Write(DataType.DataBlock, db, 2, (ushort)9); var wordDemoRead = plc.Read(DataType.DataBlock, db, 4, VarType.Int, 1); Console.WriteLine("修改后word值打印："+ wordDemoRead); var DwordDemo = 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 readDemo = plc.Read(DataType.DataBlock, db, 8, VarType.Int, 1); Console.WriteLine("short值打印："+ intDemo); string stringDemo = Encoding.Default.GetString(stringDemo); Console.WriteLine("string值打印："+ stringValue); plc.Write(DataType.DataBlock, db, 26 + 1, (byte)"nihao"); plc.Write(DataType.DataBlock, db, 26 + 2, "nihao"); stringDemo = plc.ReadBytes(DataType.DataBlock, db, 26 + 2, count); string stringDemoRead = plc.Read(DataType.DataBlock, db, 10, VarType.Int, 1); Console.WriteLine("修改后Dint值打印："+ DwordDemo); plc.Write(DataType.DataBlock, db, 10, (float)99.9); var RealDemoRead = 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, 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.Byte, 1); Console.WriteLine("byte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, 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, "a"); var charDemoRead = plc.Read(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); string = 需要先从获取string所占字节。再拿到具体byte值。转换为int格式+ascii码 具体代码中有体现 +1表示获取到长度 +2表示获取到跳过长度的字符 string类型只能存ascii码，需要注意，不能存中文+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 stringDemoRead = Encoding.Default.GetString(stringDemo); Console.WriteLine("修改后string值打印："+ stringDemoRead); wstring = wstring的读取比较简单，需要知道的是获取的字符需要为254个。因为符号占用了4个字节 所以写wstringDemo = plc.Read(DataType.DataBlock, db, startByteAddr: 292, VarType.S7String, 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[] 将传递的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：arrayDemo。也可以自定义一个新数组。注意新数组的长度至少少于plc中设置的长度。那么长度只值会写入生效。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，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //发送数据 for (i=8; i>=0; i--) { SCLK = 0; temp = dat; DIO = (bit)(temp&0x01); dat >>= 1; SCLK = 1; } CE = 0; } //数据读取子程序 unsigned char Read1302 ( unsigned char addr ) { unsigned char i,temp,dat1,dat2; CE=0; SCLK=0; CE = 1; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成二一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS1302片选5脚 //地址。数据发送子程序 void Write1302 ( unsigned char addr,dat ) { unsigned char i,temp; CE=0; /CE引脚为低，数据发送中止 SCLK=0; //清零时总线 CE = 1; /CE引脚为高，逻辑控制有效; //发送地址 for (i=8; i>=0; i--) //循环右移位 ( SCLK = 0, temp = addr, DIO = (bit)(temp&0x01) ) /每次传输低字节 addr >>= 1; /右移一位 SCLK = 1; } //读取数据 for (i=8; i>=0; i--) { ACC = DIO; SCLK = 0; ACC >>= 1; SCLK = 1; } CE=0; dat1=ACC; dat2=dat1/16; //数据进制转换 dat1=dat1%16; //十六进制转二进制 dat1=dat1*2+10; return (dat1); } //初始化DS1302 void Initial(void) { Write1302 (WRITE_PROTECT,0x00); //禁止写保护 Write1302 (WRITE_SECOND,0x56); //秒位初始化 Write1302 (WRITE_MINUTE,0x34); //分种初始化 Write1302 (WRITE_HOUR,0x12); //小时初始化 Write1302 (WRITE_PROTECT,0x80); //允许写保护 } 下面给大家分享我使用的单片机的(P1.6 (CLK ), P1.7 (CE) ),可是时间读出来总是85，很郁闷，把1302拔掉，读出来是85，知道是没有读到1302。没有办法，自己研究E文Datashet，觉得上边下的程序在读取数据的时候似乎有点问题，似乎不是下降沿读取数据而是上升沿，就自己改了程序，如下 //读取数据 for (i=8; i>=0; i--) { SCLK=0; ACC = DIO; SCLK = 0; ACC >>= 1; /SCLK = 1; } 结果还是一样，读出来是85。（其实人总是喜欢去怀疑别人，而愿意相信自己是对的，大家注意了我改的是错的，最上面的程序是对的）这个时候就像了，不知道哪里有问题，也不知道为什么什么都读不到，第一步“读取”都失败了，下面还有写入，还有存储数据根本就无法谈起，最怕就是这种问题，明明觉得没有问题，却总是得不到想要的结果，现在直接读不到任何结果。在无奈和苦闷中，我把CE接到了P1.5，（不管是不是这个问题，试试看啦，因为真的想不到其他的了），想不到奇迹竟然出现了，还真的瞎猫撞到了死耗子，居然能读出数据了，能读到初始化后的分，秒，秒的数值了。兴奋啊，赶紧趁热打铁，又写了一个定时程序，一秒钟把DS1302的“秒”读出来用数码管显示，呵呵，真的像我预料的一样，每一秒钟数码管的数值都会变化，现在更加兴奋了，简单的时钟已经实现了，可是没过几分钟就发现，秒钟跳动极不稳定，有时候正常，有时候会乱跳，从20跳到40，从5跳到18... 现在我开始怀疑自己了，怀疑自己别人的时序代码引起的，于是又改了回去，终于自己改了！ 接着又写了“修改时间”程序，因为到这一步，基本已经搞定了（这里写的程序都是已经成功的，）下面开始写DS1302的14-BYTES的RAM 程序就贴出来了，还用最上面的程序，只是寄存器地址用相应的RAM地址了。我的程序显示把秒写入进去，然后在把秒读出来数码管显示，按照预期，读出来的应该跟写进去的一样才对，因为是对同一个寄存器进行读写啊，可是读出来的值跟写进去的值死活就不一样，写进去是一个值，读出来始终都是另外一个值，试过N多次都是同样的结果。直到写入A222（我把前两位后两位分别写到两个寄存器），读出来是1616，显然不是我要找的结果。但是我让我兴奋，因为我觉得这两者之间肯定存在着某种联系，这是我想到了BCD码转换。（大家都知道DS1302的时钟寄存器读出来是BCD），所以在上面的程序中读1302的秒数读Read1302的的最后是把BCD转换成一的16进制。大家把22换成16进制看看，是不是正好是16啊！所以我认为读1302的时钟寄存器需要转换，而读RAM不需要。就在Read1302（）把转换的部分代码注释掉，哈哈，读出来果然就正确啦！（这下OK啦，后来自己写了一个很大的程序，用键盘修改时间和RAM数据，则超出异常 具体可看代码实现 //读取array float[] arrayDemo = (float[])plc.Read(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的我自己写一个帮助方法去修改值的array.plc.Write(DataType.DataBlock, db, 804, arrayDemo2); //可以修改少数几个。如果只修改其中几个则修改后完整修改 指针不能超过长度 float[] arrayDemoRead = (float[])plc.Read(DataType.DataBlock, db, 804, VarType.Real, 10); Console.WriteLine("修改后array值打印："+ System.String.Join(", ", arrayDemoRead)); 实现了一个帮助类，导入即用：资源链接：文源来源：◆◆版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请注明出处及本声明。 首先把程序的程序贴出来，是网上的，我未经过硬件测试，绝对没有用 #define WRITE_SECOND 0x80 #define WRITE_MINUTE 0x82 #define WRITE_HOUR 0x84 #define READ_SECOND 0x81 #define READ_MINUTE 0x83 #define READ_HOUR 0x85 #define WRITE_PROTECT 0x8E //位地址寄存器定义 sbit ACC = P3^5; //DS1302时钟信号7脚 sbit DIO = P3^6; //DS1302数据信号6脚 sbit CE = P3^7; //DS
```