Modbus 客户端实现

1：MODBUS\_TCP类实现--继承接口Modbus\_name与Modbus\_tru类

代码：

class Modbus\_TCP: Modbus\_RTU, Modbus\_name

{

Modbus\_name modbus\_Name;

public Modbus\_TCP()

{

modbus\_Name = new Modbus\_RTU();

}

byte[] Modbus\_name.GenerateH01(function function, ushort Station, ushort start, ushort number)

{

return message(modbus\_Name.GenerateH01(function, Station, start, number).ToList());

}

byte[] Modbus\_name.GenerateH02(function function, ushort Station, ushort start, ushort number)

{

return message(modbus\_Name.GenerateH02(function, Station, start, number).ToList());

}

byte[] Modbus\_name.GenerateH03(function function, ushort Station, ushort start, ushort number)

{

return message(modbus\_Name.GenerateH03(function, Station, start, number).ToList());

}

byte[] Modbus\_name.GenerateH05(function function, ushort Station, ushort start, coil coil)

{

return message(modbus\_Name.GenerateH05(function, Station, start, coil).ToList());

}

byte[] Modbus\_name.GenerateH06(function function, ushort Station, ushort start, ushort number)

{

return message(modbus\_Name.GenerateH06(function, Station, start, number).ToList());

}

byte[] Modbus\_name.GenerateH15(function function, ushort Station, ushort start, ushort number, byte coil)

{

return message(modbus\_Name.GenerateH15(function, Station, start, number,coil).ToList());

}

byte[] Modbus\_name.GenerateH16(function function, ushort Station, ushort start, ushort number, byte[] content)

{

return message(modbus\_Name.GenerateH16(function, Station, start, number, content).ToList());

}

//这个方法是吧rtu转TCP

public static byte[] message(List<byte> Data)

{

//实例化随机数--获取标头

Random random = new Random();

//由于TCP不需要CRC 所以需要移除CRC二个字节

Data.RemoveAt(Data.Count - 1);

Data.RemoveAt(Data.Count - 1);

byte[] vs = new byte[Data.Count + 6];//定义要发送的长度

for (int i = 0; i < vs.Length; i++)

{

if (i < 5) vs[i] = Convert.ToByte(00);//填充标头5个字节

if (i == 5) vs[i] = Convert.ToByte(Data.Count);//填充rtu字节长度

if (i > 5) vs[i] = Data[i - 6];

}

return vs;//返回报文

}

}

2：主类实现MODBUS\_RTU--继承接口Modbus\_name与Modbus\_crc-CRC效验类

代码：

public class Modbus\_RTU :Modbus\_crc, Modbus\_name

{

/// <summary>

/// 实现接口对下位机进行多线圈读取

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">读取起始地址</param>

/// <param name="number">读取个数</param>

/// <returns></returns>

byte[] Modbus\_name.GenerateH01(function function, ushort Station, ushort start, ushort number)

{

byte[] message = new byte[6];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

message[4 + i] = (int\_to\_byte(number)[i]);

}

return this.GetCRCDatas(message);

}

/// <summary>

/// 实现接口对下位机读取设备输入状态

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">读取起始地址</param>

/// <param name="number">读取个数</param>

/// <returns></returns>

byte[] Modbus\_name.GenerateH02(function function, ushort Station, ushort start, ushort number)

{

byte[] message = new byte[6];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

message[4 + i] = (int\_to\_byte(number)[i]);

}

return this.GetCRCDatas(message);

}

/// <summary>

/// 实现对下位机进行字(多字)读取--H03

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">读取起始地址</param>

/// <param name="number">读取个数（字）</param>

/// <returns></returns>

byte[] Modbus\_name.GenerateH03(function function, ushort Station, ushort start, ushort number)

{

byte[] message = new byte[6];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

message[4 + i] = (int\_to\_byte(number)[i]);

}

return this.GetCRCDatas(message);

}

/// <summary>

/// 实现对下位机进行线圈写入--H05

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">写入起始地址</param>

/// <param name="coil">要写入的状态</param>

/// <returns></returns>

byte[] Modbus\_name.GenerateH05(function function, ushort Station, ushort start, coil coil)

{

byte[] message = new byte[6];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

}

message[4] = Convert.ToByte(coil);

return this.GetCRCDatas(message);

}

/// <summary>

/// 实现对下位机进行寄存器写入--H06

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">写入起始地址</param>

/// <param name="number">要写入的数据</param>

/// <returns></returns>

byte[] Modbus\_name.GenerateH06(function function, ushort Station, ushort start, ushort number)

{

byte[] message = new byte[6];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

message[4 + i] = (int\_to\_byte(number)[i]);

}

return this.GetCRCDatas(message);

}

/// <summary>

/// 对下位机进行多线圈写入--H15

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">写入起始地</param>

/// <param name="number">要写入的个数(线圈个数)默认限制8个bit位</param>

/// <param name="coil">要写入的数据（把要写入的二进制转换成无符号字）</param>

byte[] Modbus\_name.GenerateH15(function function, ushort Station, ushort start, ushort number, byte coil)

{

byte[] message = new byte[8];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

message[4 + i] = (int\_to\_byte(number)[i]);

}

message[6] = 1;//此处是要写入的寄存器个数--默认8bit

message[7] = coil;//要写入的状态

return this.GetCRCDatas(message);

}

/// <summary>

/// 实现对下位机进行多寄存器写入--H16

/// </summary>

/// <param name="function">功能码枚举</param>

/// <param name="Station">访问的站号</param>

/// <param name="start">写入起始地</param>

/// <param name="number">要写入的个数</param>

/// <param name="content">要写入的数据数组</param>

/// <returns></returns>

byte[] Modbus\_name.GenerateH16(function function, ushort Station, ushort start, ushort number, byte[] content)

{

byte[] message = new byte[6+content.Length+1];

message[0] = Convert.ToByte(Station);

message[1] = Convert.ToByte(function);

for (int i = 0; i < 2; i++)

{

message[2 + i] = (int\_to\_byte(start)[i]);

message[4 + i] = (int\_to\_byte(number)[i]);

}

message[6] = Convert.ToByte(content.Length);

for (int i = 0; i < content.Length; i++)

message[7 + i] = content[i];

return this.GetCRCDatas(message);

}

/// <summary>

/// 进行int32转字节-放弃高字节

/// </summary>

/// <param name="Data"></param>

/// <returns></returns>

public static byte[] int\_to\_byte(int Data)

{

byte[] transition\_Data = new byte[2];

byte[] transition = BitConverter.GetBytes(Data);

if (transition.Length > 1)

for (int i = 0; i < transition\_Data.Length; i++)

transition\_Data[i] = transition[i];

else

transition\_Data[0] = transition[0];

Array.Reverse(transition\_Data);

return transition\_Data;

}

}

具体使用方法

class Program

{

/// <summary>

/// 测试通讯

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

Console.WriteLine("测试准备开始");

//实例化套接字

Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

Inex:

Console.WriteLine("请输入IP： 默认端口502");

if (!System.Net.IPAddress.TryParse(Console.ReadLine(), out System.Net.IPAddress address))

{

Console.WriteLine("输入IP错误请重新输入");

goto Inex;

}

IPEndPoint iPEndPoint = new IPEndPoint(address, int.Parse("502"));

Console.WriteLine("下位机是否准备好？？？？--准备开始链接");

Console.ReadLine();

try

{

socket.Connect(iPEndPoint);

}

catch(Exception e)

{

Console.WriteLine(e.Message);

goto Inex;

}

Modbus\_name modbus\_RTU = new Modbus\_RTU();

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH01(function.H01, 1, 0, 10).ToList()), socket)));//H01发送

Thread.Sleep(500);

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH02(function.H02, 1, 0, 10).ToList()), socket)));//H01发送

Thread.Sleep(500);

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH03(function.H03, 1, 0, 4).ToList()), socket)));//H03发送

Thread.Sleep(500);

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH05(function.H05, 1, 0, coil.ON).ToList()), socket)));//H05发送

Thread.Sleep(500);

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH06(function.H06, 1, 1, 12345).ToList()), socket)));//H06发送

Thread.Sleep(500);

//多线圈写入测试

for (int i = 0; i < 50; i++)

{

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH15(function.H15, 1, 0, 3, Convert.ToByte((3 \* i))).ToList()), socket)));//H06发送

Thread.Sleep(500);

}

//多寄存器写入

Console.WriteLine(BitConverter.ToString(communication(message(modbus\_RTU.GenerateH16(function.H16, 1, 0, 3,new byte[] { 0x1,0x2,0x3,0x4,0x5,0x66}).ToList()), socket)));//H06发送

Thread.Sleep(500);

Console.ReadLine();

}

public static byte[] communication(byte[] Data,Socket socket)

{

//报文测试

socket.Send(Data);//发送数据

byte[] message = new byte[20];

socket.Receive(message);

return message;

}

//这个方法是吧rtu转TCP--简单粗暴

public static byte[] message(List<byte> Data)

{

//实例化随机数--获取标头

Random random = new Random();

//由于TCP不需要CRC 所以需要移除CRC二个字节

Data.RemoveAt(Data.Count - 1);

Data.RemoveAt(Data.Count - 1);

byte[] vs = new byte[Data.Count + 6];//定义要发送的长度

for (int i = 0; i < vs.Length; i++)

{

if (i < 5) vs[i] = Convert.ToByte(00);//填充标头5个字节

if (i == 5) vs[i] = Convert.ToByte(Data.Count);//填充rtu字节长度

if (i > 5) vs[i] = Data[i - 6];

}

return vs;//返回报文

}