蓝牙虚拟串口操作

蓝牙基本操作：搜索-连接-发送-接收

主要注意的地方就是串口连接再断开再连接的问题，要有个类似于注册表中注销串口的操作，才能下次继续申请串口。

namespace BluetoothManager

{

    /// <summary>

    /// 采用Platform/Invoke方式进行蓝牙操作类

    /// </summary>

    public static class Win32Bluetooth

    {

        //获取本地蓝牙设备的设备地址

        [DllImport("Btdrt.dll", SetLastError = true)]

        public static extern int BthReadLocalAddr(byte[] PBa);

        public enum RadioMode

        {

            BTH\_POWER\_OFF = 0,

            BTH\_CONNECTABLE = 1,

            BTH\_DISCOVERABLE = 2

        }

        [DllImport("BthUtil.dll", SetLastError = true)]

        public static extern int BthGetMode(out RadioMode dwMode);

        [DllImport("BthUtil.dll", SetLastError = true)]

        public static extern int BthSetMode(RadioMode dwMode);

        //是否存在蓝牙设备

        //如果获取到的蓝牙地址是空，或者"00:00:00:00:00:00"

        //就假定本设备没有蓝牙

        public static bool ExistBluetooth()

        {

            string address = GetDeviceAddress();

            if (string.IsNullOrEmpty(address) || address == "00:00:00:00:00:00")

            {

                return false;

            }

            return true;

        }

        /// <summary>

        /// 获取蓝牙地址

        /// </summary>

        /// <returns></returns>

        public static string GetDeviceAddress()

        {

            byte[] data = new byte[6];

            string address = string.Empty;

            try

            {

                int d=BthReadLocalAddr(data);

                for (int i = 5; i >0; i--)

                {

                    address += data[i].ToString("X2") + ":";

                }

                address += data[0].ToString("X2");

                return address;

            }

            catch

            {

                address = string.Empty;

                return address;

            }

        }

        //打开蓝牙

        public static bool Open()

        {

            bool res = false;

            try

            {

                RadioMode mode = RadioMode.BTH\_POWER\_OFF;

                int ret = 0;

                //检查蓝牙状态，如果不是"连接"状态，则设为"连接"状态

                ret = BthGetMode(out mode);

                if (mode != RadioMode.BTH\_CONNECTABLE)

                    BthSetMode(RadioMode.BTH\_CONNECTABLE);

                res = true;

                return res;

            }

            catch

            {

                res = false;

                return res;

            }

        }

        //关闭蓝牙

        public static void Close()

        {

            int i = BthSetMode(RadioMode.BTH\_POWER\_OFF);

        }

        /// <summary>

        /// 搜索附近的蓝牙设备

        /// </summary>

        /// <returns></returns>

        public static BluetoothDeviceInfo[] SearchBluetooth()

        {

            BluetoothClient Client = new BluetoothClient();

            BluetoothDeviceInfo[] BthDevices = Client.DiscoverDevices();

            return BthDevices;

        }

    }

}

搜索蓝牙：

          bool canSearch = Win32Bluetooth.ExistBluetooth();

            if (!canSearch)

            {

                this.lblMsg.Text = "本设备没有蓝牙,请确认!";

                return;

            }

            this.lblMsg.Text = "正在搜索蓝牙设备,请稍候...";

                    if (Win32Bluetooth.Open())

                    {

                        exe.ChangeMessage( "正在搜索蓝牙设备,请稍候...");

                        BluetoothDevice = Win32Bluetooth.SearchBluetooth();

                        DevicesNameAddress.Clear();

                        if (BluetoothDevice != null)

                        {

                          //  XmlUtility.DevicesRecord.Clear();

                            foreach (BluetoothDeviceInfo ent in BluetoothDevice)

                            {

                                string name = ent.DeviceName;

                                string address = ent.DeviceAddress.ToString();

                                DevicesNameAddress.Add(address, name);

                           //     XmlUtility.DevicesRecord.Add(address, name);

                            }

                        }

                    }

连接蓝牙：

   [DllImport("CoreDll.dll")]

        private static extern bool DeregisterDevice(

            IntPtr handle);//取消注册用

定义：

        private BluetoothEndPoint endPoint;

        SerialPort serialPort = new SerialPort();

        BluetoothSerialPort bluetoothSerialPort = null;

        BluetoothClient client = null;

        string portName = string.Empty;

连接：

                      try

                       {

                           if (IsSelectPort)

                           {

                               if (serialPort != null)

                               {

                                   if (serialPort.IsOpen)

                                   {

                                       serialPort.Close();

                                   }

                               }

                               if (bluetoothSerialPort != null)

                               {

                                   bluetoothSerialPort.Close();

                                   bluetoothSerialPort.Dispose();

                                   bluetoothSerialPort = null;

                               }

                           }

                           else

                           {

                               if (client != null)

                               {

                                   client.Close();

                                   client.Dispose();

                                   client = null;

                               }

                           }

                       }

                       catch (Exception ex)

                       {

                       }

                            byte[] byteArray = ByteHexConvertUtility.StringToByteArray(strAddress, 0, strAddress.Length, true);

                           BluetoothAddress address = new BluetoothAddress(byteArray);

                           endPoint = new BluetoothEndPoint(address, BluetoothService.SerialPort);

                           if (IsSelectPort)//串口通讯

                           {

                               try

                               {

                                   bluetoothSerialPort = BluetoothSerialPort.CreateClient(endPoint);

                                   portName = bluetoothSerialPort.PortName;

                               }

                               catch (Exception ex)

                               {

                                   for (int i = 0; i <= 9; i++)

                                   {

                                       temp = string.Format("COM{0}", i);

                                       try

                                       {

                                           bluetoothSerialPort = BluetoothSerialPort.CreateClient(temp, endPoint);

                                           portName = temp;

                                           break;

                                       }

                                       catch

                                       {

                                           continue;

                                       }

                                   }

                                   if (string.IsNullOrEmpty(portName))

                                   {

                                       portName = "COM9";

                                   }

                                   System.Threading.Thread.Sleep(1500);

                               }

                               #endregion

                               try

                               {

                                   //port = new SerialPort(portName, 38400);

                                   SelectParity();//波特率设置

                                   serialPort.PortName = portName;

                                   serialPort.BaudRate = tempbaudrate;

                                   serialPort.Parity = tempParity;

                                   serialPort.DataBits = tempdatabits;

                                   serialPort.StopBits = tempStopBits;

                                   serialPort.WriteTimeout = 2000;

                                   serialPort.Open();

                                   txtCom.Invoke(new EventHandler(delegate

                                   {

                                       txtCom.Text = serialPort.PortName;

                                   }));

                               }

                               catch (Exception ex)

                               {

                                   string temp = string.Empty;

                                   for (int i = 0; i <= 9; i++)

                                   {

                                       temp = string.Format("COM{0}", i);

                                       try

                                       {

                                           SelectParity();

                                           serialPort.PortName = portName;

                                           serialPort.BaudRate = tempbaudrate;

                                           serialPort.Parity = tempParity;

                                           serialPort.DataBits = tempdatabits;

                                           serialPort.StopBits = tempStopBits;

                                           serialPort.WriteTimeout = 2000;

                                           serialPort.Open();

                                           break;

                                       }

                                       catch

                                       {

                                           continue;

                                       }

                                   }

                                   System.Threading.Thread.Sleep(1500);

                               }

                               #endregion

                           }

                           else//SOCKET

                           {

                               client = new BluetoothClient();

                               client.Connect(endPoint);

                               if (client.Connected)

                               {

                                   isConnect = true;

                                   isRun = true;

                                   if (thrReceiveUii == null)

                                   {

                                       thrReceiveUii = new System.Threading.Thread(ReceiveUiiProc);

                                       thrReceiveUii.IsBackground = true;

                                       thrReceiveUii.Start();

                                   }

                                   else

                                   {

                                       thrReceiveUii.Start();

                                   }

                               }

                           }

接收：

   #region 接收

        /// <summary>

        /// 串口接收

        /// </summary>

        /// <param name="sender"></param>

        /// <param name="e"></param>

         void serialPort\_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)

        {

            if (!isConnect)

                return;

            if (!serialPort.IsOpen)

                return;

            try

            {

                System.Threading.Thread.Sleep(200);

                int co = serialPort.BytesToRead;

                byte[] buffer = new byte[co];

                if (co == 0)

                    return;

                else

                    serialPort.Read(buffer, 0, co);

                txtRe.Invoke(new EventHandler(delegate

                {

                    txtRe.Text = ByteArrayToHexString(buffer).Replace(" ","");

                }));

            }

            catch (Exception ex)

            {

            }

        }

        /// <summary>

        /// SOCKet接收

        /// </summary>

        void ReceiveUiiProc()

        {

            try

            {

                while (isRun)

                {

                    if (isConnect)

                    {

                        GpsStream = this.client.GetStream();

                        num1 = GpsStream.Read(buffer1, 0, 0x100);

                        if (num1 == 0)

                            return;

                      //  gpsinfo = Encoding.ASCII.GetString(buffer1, 0, num1);

                       byte[] tempdata=new byte[num1];

                       Array.Copy(buffer1, tempdata, num1);

                        txtRe.Invoke(new EventHandler(delegate

                        {

                            txtRe.Text = ByteArrayToHexString(tempdata).Replace(" ","");

                        }));

                    }

                }

            }

            catch (Exception ex)

            {

            }

        }

        #endregion

关闭：

 try

                  {

                      if (serialPort != null)

                      {

                          if (serialPort.IsOpen)

                          {

                              serialPort.Close();

                          }

                      }

                      if (bluetoothSerialPort != null)

                      {

                          bluetoothSerialPort.Close();

                          DeregisterDevice(bluetoothSerialPort.Handle);

                          bluetoothSerialPort.Dispose();

                          bluetoothSerialPort = null;

                      }

                      if (client != null)

                      {

                          client.Close();

                          client.Dispose();

                          client = null;

                      }

                  }

                  catch (Exception ex)

                  {

                  }