公安部已经大规模在全国范围内推广第二代居民身份证的换发工作，目前，许多地区二代证换发工作已经进行了很久。但是，相对应于二代证的推广，社会上许许多多的应用系统都还没有对二代证读卡开发相应的接口，为此，我写了一个通用的二代证机具读卡类，希望能对各位在各自的系统中开发这样的接口提供帮助。

本类仅仅提供读卡的用法，写卡的方式我并没有写在里面，但是，相应的写调用我已经声明在里面，希望各位大侠能独自研究应用。 

该类接口具有如下特点：

1、通用于目前市场上各个机具厂商的二代证读（写）机具。（想想，为不同厂商开发不同的接口确实令人头疼的，且由于接口不兼容，容易被客户骂死！  ）

2、自动适应串口、USB 口的各种机具

3、能读文字信息，但是，要想正确读照片，必须需要机具厂商的授权文件 Termb.Lic，且授权文件必须放在 C 盘根目录下。（我总不能不照顾机具厂商的利益吧  ）

4、除了上面第三条之外，您甚至不用安装机具的驱动程序，接上二代证机具即可使用。

废话少说，下面看实现方法：

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.Reflection ;
using System.IO;


namespace ICCard
{
    public class clsICCard
    {
        //首先，声明通用接口
        [DllImport("sdtapi.dll")]
        public static extern int SDT_OpenPort(int iPortID);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ClosePort(int iPortID);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_PowerManagerBegin(int iPortID, int iIfOpen);
        [DllImport("sdtapi.dll")]
```

```
        public static extern int SDT_AddSAMUser(int iPortID, string pcUserName, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_SAMLogin(int iPortID, string pcUserName, string pcPasswd, int iI
fOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_SAMLogout(int iPortID, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_UserManagerOK(int iPortID, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ChangeOwnPwd(int iPortID, string pcOldPasswd, string pcNewPa
sswd, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ChangeOtherPwd(int iPortID, string pcUserName, string pcNewPa
sswd, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_DeleteSAMUser(int iPortID, string pcUserName, int iIfOpen);

        [DllImport("sdtapi.dll")]
        public static extern int SDT_StartFindIDCard(int iPortID, ref int pucIIN, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_SelectIDCard(int iPortID, ref int pucSN, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ReadBaseMsg(int iPortID, string pucCHMsg, ref int puiCHMsgLe
n, string pucPHMsg, ref int puiPHMsgLen, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ReadBaseMsgToFile(int iPortID, string fileName1, ref int puiCHM
sgLen, string fileName2, ref int puiPHMsgLen, int iIfOpen);

        [DllImport("sdtapi.dll")]
        public static extern int SDT_WriteAppMsg(int iPortID, ref byte pucSendData, int uiSendLen, re
f byte pucRecvData, ref int puiRecvLen, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_WriteAppMsgOK(int iPortID, ref byte pucData, int uiLen, int iIfO
pen);

        [DllImport("sdtapi.dll")]
        public static extern int SDT_CancelWriteAppMsg(int iPortID, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ReadNewAppMsg(int iPortID, ref byte pucAppMsg, ref int puiApp
MsgLen, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ReadAllAppMsg(int iPortID, ref byte pucAppMsg, ref int puiApp
MsgLen, int iIfOpen);
        [DllImport("sdtapi.dll")]
```

```
        public static extern int SDT_UsableAppMsg(int iPortID, ref byte ucByte, int iIfOpen);

        [DllImport("sdtapi.dll")]
        public static extern int SDT_GetUnlockMsg(int iPortID, ref byte strMsg, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_GetSAMID(int iPortID, ref byte StrSAMID, int iIfOpen);

        [DllImport("sdtapi.dll")]
        public static extern int SDT_SetMaxRFByte(int iPortID, byte ucByte, int iIfOpen);
        [DllImport("sdtapi.dll")]
        public static extern int SDT_ResetSAM(int iPortID, int iIfOpen);

        [DllImport("WltRS.dll")]
        public static extern int GetBmp(string file_name, int intf);

        public delegate void De_ReadICCardComplete(clsEDZ objEDZ);
        public event De_ReadICCardComplete ReadICCardComplete;
        private clsEDZ objEDZ = new clsEDZ();
        private int EdziIfOpen = 1;             //自动开关串口
        int EdziPortID;
        public clsICCard()
        {

        }
        public bool ReadICCard()
        {
            bool bUsbPort = false;
            int intOpenPortRtn = 0;
            int rtnTemp = 0;
            int pucIIN = 0;
            int pucSN = 0;
            int puiCHMsgLen = 0;
            int puiPHMsgLen = 0;

            objEDZ = new clsEDZ();
            //检测 usb 口的机具连接，必须先检测 usb
            for (int iPort = 1001; iPort <= 1016; iPort++)
            {
                intOpenPortRtn = SDT_OpenPort(iPort);
                if (intOpenPortRtn == 144)
                {
                    EdziPortID = iPort;
                    bUsbPort = true;
                    break;
```

```
            }
        }
        //检测串口的机具连接
        if (!bUsbPort)
        {
            for (int iPort = 1; iPort <= 2; iPort++)
            {
                intOpenPortRtn = SDT_OpenPort(iPort);
                if (intOpenPortRtn == 144)
                {
                    EdziPortID = iPort;
                    bUsbPort = false;
                    break;
                }
            }
        }
        if (intOpenPortRtn != 144)
        {
            MessageBox.Show("端口打开失败，请检测相应的端口或者重新连接读卡器！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
        //在这里，如果您想下一次不用再耗费检查端口的检查的过程，您可以把 EdziPortID 保存下来，可以保存在注册表中，也可以保存在配置文件中，我就不多写了，但是，
        //您要考虑机具连接端口被用户改变的情况哦

        //下面找卡
        rtnTemp = SDT_StartFindIDCard(EdziPortID, ref pucIIN, EdziIfOpen);
        if (rtnTemp != 159)
        {
            rtnTemp = SDT_StartFindIDCard(EdziPortID, ref pucIIN, EdziIfOpen);  //再找卡
            if (rtnTemp != 159)
            {
                rtnTemp = SDT_ClosePort(EdziPortID);
                MessageBox.Show("未放卡或者卡未放好，请重新放卡！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }
        //选卡
        rtnTemp = SDT_SelectIDCard(EdziPortID, ref pucSN, EdziIfOpen);
        if (rtnTemp != 144)
        {
            rtnTemp = SDT_SelectIDCard(EdziPortID, ref pucSN, EdziIfOpen);  //再选卡
```

```
            if (rtnTemp != 144)
            {
                rtnTemp = SDT_ClosePort(EdziPortID);
                MessageBox.Show("读卡失败！", "提示", MessageBoxButtons.OK, MessageBoxIcon.
Error);
                return false;
            }
        }
        //注意，在这里，用户必须有应用程序当前目录的读写权限
        FileInfo objFile = new FileInfo("wz.txt");
        if (objFile.Exists)
        {
            objFile.Attributes = FileAttributes.Normal;
            objFile.Delete();
        }
        objFile = new FileInfo("zp.bmp");
        if (objFile.Exists)
        {
            objFile.Attributes = FileAttributes.Normal;
            objFile.Delete();
        }
        objFile = new FileInfo("zp.wlt");
        if (objFile.Exists)
        {
            objFile.Attributes = FileAttributes.Normal;
            objFile.Delete();
        }
        rtnTemp = SDT_ReadBaseMsgToFile(EdziPortID, "wz.txt", ref puiCHMsgLen, "zp.wlt", re
f puiPHMsgLen, EdziIfOpen);
        if (rtnTemp != 144)
        {
            rtnTemp = SDT_ClosePort(EdziPortID);
            MessageBox.Show("读卡失败！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Err
or);
            return false;
        }
        //下面解析照片，注意，如果在 C 盘根目录下没有机具厂商的授权文件 Termb.Lic，照
片解析将会失败
        if (bUsbPort)
            rtnTemp = GetBmp("zp.wlt", 2);
        else
            rtnTemp = GetBmp("zp.wlt", 1);
        switch (rtnTemp)
        {
```

```
            case 0:
                MessageBox.Show("调用 sdtapi.dll 错误！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            case 1:  //正常
                break;
            case -1:
                MessageBox.Show("相片解码错误！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            case -2:
                MessageBox.Show("wlt 文件后缀错误！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            case -3:
                MessageBox.Show("wlt 文件打开错误！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            case -4:
                MessageBox.Show("wlt 文件格式错误！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            case -5:
                MessageBox.Show("软件未授权！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            case -6:
                MessageBox.Show("设备连接错误！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
        }
        rtnTemp = SDT_ClosePort(EdziPortID);
        FileInfo f = new FileInfo("wz.txt");
        FileStream fs = f.OpenRead();
        byte[] bt = new byte[fs.Length];
        fs.Read(bt, 0, (int)fs.Length);
        fs.Close();

        string str = System.Text.UnicodeEncoding.Unicode.GetString(bt);

        objEDZ.Name = System.Text.UnicodeEncoding.Unicode.GetString(bt, 0, 30).Trim();
        objEDZ.Sex_Code = System.Text.UnicodeEncoding.Unicode.GetString(bt, 30, 2).Trim();
        objEDZ.NATION_Code = System.Text.UnicodeEncoding.Unicode.GetString(bt, 32, 4).Trim();
```

```
            string strBird = System.Text.UnicodeEncoding.Unicode.GetString(bt, 36, 16).Trim();
            objEDZ.BIRTH = Convert.ToDateTime(strBird.Substring(0, 4) + "年" + strBird.Substring
(4, 2) + "月" + strBird.Substring(6) + "日");
            objEDZ.ADDRESS = System.Text.UnicodeEncoding.Unicode.GetString(bt, 52, 70).Trim();
            objEDZ.IDC = System.Text.UnicodeEncoding.Unicode.GetString(bt, 122, 36).Trim();
            objEDZ.REGORG = System.Text.UnicodeEncoding.Unicode.GetString(bt, 158, 30).Trim();
            string strTem = System.Text.UnicodeEncoding.Unicode.GetString(bt, 188, bt.GetLength
(0) - 188).Trim();
            objEDZ.STARTDATE = Convert.ToDateTime(strTem.Substring(0, 4) + "年" + strTem.Subst
ring(4, 2) + "月" + strTem.Substring(6, 2) + "日");
            strTem = strTem.Substring(8);
            if (strTem.Trim() != "长期")
            {
                objEDZ.ENDDATE = Convert.ToDateTime(strTem.Substring(0, 4) + "年" + strTem.Subst
ring(4, 2) + "月" + strTem.Substring(6, 2) + "日");
            }
            else
            {
                objEDZ.ENDDATE = DateTime.MaxValue;
            }
            objFile = new FileInfo("zp.bmp");
            if (objFile.Exists)
            {
                Image img = Image.FromFile("zp.bmp");
                objEDZ.PIC_Image=(Image)img.Clone();
                System.IO.MemoryStream m = new MemoryStream();
                img.Save(m, System.Drawing.Imaging.ImageFormat.Jpeg);
                objEDZ.PIC_Byte = m.ToArray();
                img.Dispose();
                img = null;
            }
            ReadICCardComplete(objEDZ);
            return true;
        }


    }
    public class clsEDZ
    {
        private System.Collections.SortedList lstMZ = new SortedList();
        private string _Name;   //姓名
        private string _Sex_Code;   //性别代码
        private string _Sex_CName;   //性别
        private string _IDC;     //身份证号码
```

```
        private string _NATION_Code;  //民族代码
        private string _NATION_CName;  //民族
        private DateTime _BIRTH;    //出生日期
        private string _ADDRESS;  //住址
        private string _REGORG;    //签发机关
        private DateTime _STARTDATE;  //身份证有效起始日期
        private DateTime _ENDDATE;  //身份证有效截至日期
        private string _Period_Of_Validity_Code;  //有效期限代码，许多原来系统上面为了一代证考虑，常常存在这样的字段，二代证中已经没有了
        private string _Period_Of_Validity_CName;  //有效期限
        private byte[] _PIC_Byte;  //照片二进制
        private Image _PIC_Image;  //照片

        public clsEDZ()
        {
            lstMZ.Add("01", "汉族");
            lstMZ.Add("02", "蒙古族");
            lstMZ.Add("03", "回族");
            lstMZ.Add("04", "藏族");
            lstMZ.Add("05", "维吾尔族");
            lstMZ.Add("06", "苗族");
            lstMZ.Add("07", "彝族");
            lstMZ.Add("08", "壮族");
            lstMZ.Add("09", "布依族");
            lstMZ.Add("10", "朝鲜族");
            lstMZ.Add("11", "满族");
            lstMZ.Add("12", "侗族");
            lstMZ.Add("13", "瑶族");
            lstMZ.Add("14", "白族");
            lstMZ.Add("15", "土家族");
            lstMZ.Add("16", "哈尼族");
            lstMZ.Add("17", "哈萨克族");
            lstMZ.Add("18", "傣族");
            lstMZ.Add("19", "黎族");
            lstMZ.Add("20", "傈僳族");
            lstMZ.Add("21", "佤族");
            lstMZ.Add("22", "畲族");
            lstMZ.Add("23", "高山族");
            lstMZ.Add("24", "拉祜族");
            lstMZ.Add("25", "水族");
            lstMZ.Add("26", "东乡族");
            lstMZ.Add("27", "纳西族");
            lstMZ.Add("28", "景颇族");
            lstMZ.Add("29", "柯尔克孜族");
```

```
            lstMZ.Add("30", "土族");
            lstMZ.Add("31", "达翰尔族");
            lstMZ.Add("32", "仫佬族");
            lstMZ.Add("33", "羌族");
            lstMZ.Add("34", "布朗族");
            lstMZ.Add("35", "撒拉族");
            lstMZ.Add("36", "毛南族");
            lstMZ.Add("37", "仡佬族");
            lstMZ.Add("38", "锡伯族");
            lstMZ.Add("39", "阿昌族");
            lstMZ.Add("40", "普米族");
            lstMZ.Add("41", "塔吉克族");
            lstMZ.Add("42", "怒族");
            lstMZ.Add("43", "乌孜别克族");
            lstMZ.Add("44", "俄罗斯族");
            lstMZ.Add("45", "鄂温克族");
            lstMZ.Add("46", "德昂族");
            lstMZ.Add("47", "保安族");
            lstMZ.Add("48", "裕固族");
            lstMZ.Add("49", "京族");
            lstMZ.Add("50", "塔塔尔族");
            lstMZ.Add("51", "独龙族");
            lstMZ.Add("52", "鄂伦春族");
            lstMZ.Add("53", "赫哲族");
            lstMZ.Add("54", "门巴族");
            lstMZ.Add("55", "珞巴族");
            lstMZ.Add("56", "基诺族");
            lstMZ.Add("57", "其它");
            lstMZ.Add("98", "外国人入籍");
        }

        public string Name
        {
            get { return _Name; }
            set { _Name = value; }
        }
        public string Sex_Code
        {
            get { return _Sex_Code; }
            set
            {
                _Sex_Code = value;
                switch (value)
                {
```

```
                case "1":
                    Sex_CName = "男";
                    break;
                case "2":
                    Sex_CName = "女";
                    break;
            }
        }
    }
    public string Sex_CName
    {
        get { return _Sex_CName; }
        set { _Sex_CName = value; }
    }
    public string IDC
    {
        get { return _IDC; }
        set { _IDC = value; }
    }
    public string NATION_Code
    {
        get { return _NATION_Code; }
        set
        {
            _NATION_Code = value;
            if (lstMZ.Contains(value))
                NATION_CName = lstMZ[value].ToString();
        }
    }
    public string NATION_CName
    {
        get { return _NATION_CName; }
        set { _NATION_CName = value; }
    }
    public DateTime BIRTH
    {
        get { return _BIRTH; }
        set { _BIRTH = value; }
    }
    public string ADDRESS
    {
        get { return _ADDRESS; }
        set { _ADDRESS = value; }
    }
```

```
        public string REGORG
        {
            get { return _REGORG; }
            set { _REGORG = value; }
        }
        public DateTime STARTDATE
        {
            get { return _STARTDATE; }
            set { _STARTDATE = value; }
        }
        public DateTime ENDDATE
        {
            get { return _ENDDATE; }
            set
            {
                _ENDDATE = value;
                if (_ENDDATE == DateTime.MaxValue)
                {
                    _Period_Of_Validity_Code = "3";
                    _Period_Of_Validity_CName = "长期";
                }
                else
                {
                    if (_STARTDATE != DateTime.MinValue)
                    {
                        switch (value.AddDays(1).Year - _STARTDATE.Year)
                        {
                            case 5:
                                _Period_Of_Validity_Code = "4";
                                _Period_Of_Validity_CName = "5 年";
                                break;
                            case 10:
                                _Period_Of_Validity_Code = "1";
                                _Period_Of_Validity_CName = "10 年";
                                break;
                            case 20:
                                _Period_Of_Validity_Code = "2";
                                _Period_Of_Validity_CName = "20 年";
                                break;
                        }
                    }
                }

            }
```

```
    }
    public string Period_Of_Validity_Code
    {
        get { return _Period_Of_Validity_Code; }
        set { _Period_Of_Validity_Code = value; }
    }
    public string Period_Of_Validity_CName
    {
        get { return _Period_Of_Validity_CName; }
        set { _Period_Of_Validity_CName = value; }
    }
    public byte[] PIC_Byte
    {
        get { return _PIC_Byte; }
        set { _PIC_Byte = value; }
    }
    public Image PIC_Image
    {
        get { return _PIC_Image; }
        set { _PIC_Image = value; }
    }


  }
}
```

类中使用到了两个 dll 文件(sdtapi.dll 和 WltRS.dll)，点击这里下载！