

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Threading;
using System.Text;
using JustInIO;
using SMS;
using System.IO;
using System.Timers;

    public struct message //定期发送有关信息
    {
        public string number;
        public string center;
        public string mess;
    };

    message message1 = new message();
    message message2 = new message();
    message message3 = new message();
    string state = ""; //存储发送状况

```

```

        int x = 0, y = 0; //发送计数

        ///

```

```

        {
            ss_port.Close();
            ss_port.Open();
        }
else
    {
        ss_port.Open();//打开串口
    }
return true;
}
catch
    (Exception)
    {
        MessageBox.Show("错误:" + e.Message);
        return false;
    }
}
//初始化代码, 并获取有关信息
private void btnConnect_Click(object sender, System.EventArgs e)
    {
        bool opened =

```

```

InitCom(ConnectPort.SelectedItem.ToString() , Convert
t.ToInt32(ConnectBaudRate.SelectedItem.ToString()));

    bool Connected = false;

    if (opened)
    {
        ss_port.Write(Encoding.ASCII.GetBytes
es("AT+CGMI\r")); //获取品牌

        string response = Encoding.ASC
II. GetString(ss_port.Read(128)) ;

        if(response.Length > 0)
        {
            ConnectState.Text = respo
nse. Substring (2, 16);

            Connected = true;
        }
        else
        {
            ConnectState.Text = " 连接不成功";

            Connected = false;
        }

        ss_port.Write (Encoding.ASCII.GetBy
tes("AT+CGMM\r")); //获取型号

```

```

        response = Encoding.ASCII.GetString(ss_port.Read(128));
        if (response.Length > 0)
        {
            ConnectState.Text =
ConnectState.Text + " " + response.Substring(2, 6) + "
已连接";

            Connected = true;
        }
        else
        {
            ConnectState.Text = "连接不成功";
            Connected = false;
        }

        ss_port.Write(Encoding.ASCII.GetBytes("AT+CSCA?\r")); //获取短信中心号

        response = Encoding.ASCII.GetString(ss_port.Read(128));
        if (response.Length > 0)
        {
            CenterNumber.Text = response.Substring(11, 13);

```

Connected = true ;

```

    }
else
    {
        Connected = false;
    }
    if (Connected == true)
    {
        btnConnect.Enabled = false;
        btnSend.Enabled = true;
        receive.Enabled = true;
        unread.Enabled = true;
        sendstate.Enabled = true;
        clearsend.Enabled = true;
    }
else
    {
        btnConnect.Enabled = true;
        btnSend.Enabled = false;
        receive.Enabled = false;
        unread.Enabled = false;
        sendstate.Enabled = false;
        clearsend.Enabled = false;
    }
}

```

e;



```

    }

    timer2.Start(); //连接成功后，每1秒
自动读取一次未读信息，若有未读则显示

    }
}

//发送短信

private void btnSend_Click(object sender,
System.EventArgs e)
{
    //输入对方手机号码局限性或短信为空时报错并重启
    string num = targetNumber.Text;
    num = num.Substring(0, 2);
    bool result = string.Equals(num,
"13");
    if (!result)
    {
        result = string.Equals(num, "14");
        if (!result)
        {
            result = string.Equals(num, "15");
            if (!result)
                result =

```

```

string.Equals (num, "18");
        }
    }
    if ((!result) || (targetNumber. TextLength != 11) || (smsContent. TextLength == 0) )
    {
        MessageBox.Show("请检核对方手机号且
短信内容不得为空!", "警告");
        return;
    }
    //选择定期则启动定期器
    if (radioButton1.Enabled & hour.Enabled)//
定期1
    {
        message1.number = targetNumber. Text;
        message1.center = CenterNumber. Text;
        message1.message = smsContent. Text;
        int second = (int.Parse(hour. Text) * 60
+ int.Parse (min.Text) ) * 60+1; //若定期为0,避免定期出错,

```

默觉得1秒

```
timer3.Interval = 1000 * second;
```

```

        smsState.Text = "信息将在" + hour.Text
        + "小时" + min.Text + "分钟之后发出。 \r\n ";
        timer3.Start ();
        radioButton1.Enabled = false;
        return;
    }
    else if (radioButton2.Enabled & hour2.Enabled) //定期2
    {
        message2.number = targetNumber.Text;
        message2.center = CenterNumber.Text;
        message2.mess = smsContent.Text;
        int second = (int.Parse(hour2.Text) * 60 + int.Parse(min2.Text) ) * 60 + 1;
        timer4.Interval = 1000 * second;
        smsState.Text = "信息将在" + hour2.Text + "小时" + min2.Text + "分钟之后发出。 \r\n ";
        timer4.Start();
        radioButton2.Enabled = false;
        return;
    }
}

```

```

    }

    else if (radioButton4.Enabled &
hour3.Enabled) //定期3
    {
        message3.number = targetNumber.Text;
        message3.center = CenterNumber
er.Text;

        message3.message = smsContent.Text;

        int second = (int.Parse(hour3.Text)
t) * 60 + int.Parse(min3.Text) * 60 + 1;

        timer5.Interval = 1000 * second;

        smsState.Text = "信息将在 " +
hour3.Text + "小时" + min3.Text + "分钟之后发出。 \r\n";

        timer5.Start();

        radioButton4.Enabled = false;

        return;
    }

    Thread.Sleep ();//目前线程延迟2秒,提高发送成
功率

    string decodedSMS = sms.smsD
ecodedsms(CenterNumber.Text, targetNumber.Text

```

t, smsContent.Text);

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/236141151203010240>