VS2010串口通信快速开发(完整版)

[1、概述 2](#_Toc496439039)

[2、关于ComXYZ 2](#_Toc496439040)

[2.1、获取控件文件 2](#_Toc496439041)

[2.2、开发环境 2](#_Toc496439042)

[2.3、ComXYZ控件关系 2](#_Toc496439043)

[2.4、添加ComXYZ控件到工具箱 3](#_Toc496439044)

[3、第一个例子（ComY） 6](#_Toc496439045)

[3.2、编写代码 8](#_Toc496439046)

[3.3、打开串口 8](#_Toc496439047)

[3.4、关闭串口 9](#_Toc496439048)

[3.5、发送数据及显示发送数据量 9](#_Toc496439049)

[3.6、接收数据及显示接收数据量 9](#_Toc496439050)

[3.7、清空计数器 10](#_Toc496439051)

[3.8、代码量 10](#_Toc496439052)

[3.9、发送接收中文及ASCII 13](#_Toc496439053)

[4、第2个例子（ComZ） 14](#_Toc496439054)

[4.1、新建窗体工程（testComZ）及添加控件如下表 14](#_Toc496439055)

[4.2、编写代码 15](#_Toc496439056)

[4.3、发送数据及显示发送数据量 16](#_Toc496439057)

[4.4、接收数据及显示接收数据量 16](#_Toc496439058)

[4.5、清零计数器 17](#_Toc496439059)

[4.6、运行效果图 17](#_Toc496439060)

[4.7、代码量（全部仅仅35行）如下图 19](#_Toc496439061)

[5、ComXYZ接口函数 20](#_Toc496439062)

[5.1、ComY 20](#_Toc496439063)

[5.2、ComZ 22](#_Toc496439064)

[6、获取测试例子，请加QQ群321903608 23](#_Toc496439065)

1、概述

基于PC的串口通信方法通常有，WIN API，MSCOM控件，SerialPort控件等，现在这里要介绍的是ComXYZ三个控件，使用这个三个控件均能快速开发出你的串口通信上位机程序。优点1：速度快，优点2：速度快，优点3：速度快。只需简单的串口配置，或者不需配置（ComZ时），仅仅需要调用控件提供的发送接收函数即可，无需考虑是中文发送还是英文ASCII发送，也不需要考虑ASCII发送还是16进制发送，它们均已封装在控件里，你只需在调用发送函数时指定是16进制发送还是ASCII发送；接收部分也已经封装好，你只需写一个线程查询是否有接收到数据，再决定你需要的数据格式，控件提供了，16进制数组---用于数据分析（分析数据之前可强制清空缓存），16进制ASCII模式（数据之间自动插入一个空格）---用于显示，字符串模式---用于显示或者分析。

2、关于ComXYZ

2.1、获取控件文件

2.1.1、下载地址：（永久有效）

2.1.2、请加QQ群321903608（获取新版本文件---也可关注作者的网盘）

2.2、开发环境

基于VS2010/2的串口通信程序，语言：ｃ#（本人验证及开发它的环境，其他环境请自行验证）

2.3、ComXYZ控件关系

2.3.1、ComX

作为整个控件的母亲，包含了ComY与ComZ的全部代码，也是最早的一个复合控件。（不再更新，不推荐使用）

2.3.2、ComY

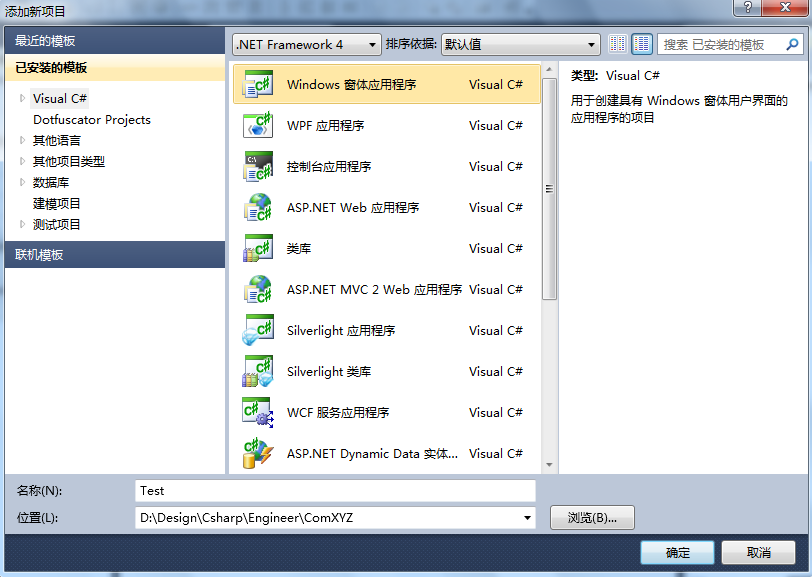
对ComX进行简化，去除界面元素，仅保留必要的通信功能，当需要个性化的串口配置显示方式时使用。仅需进行必要的串口配置即可。

2.3.3、ComZ

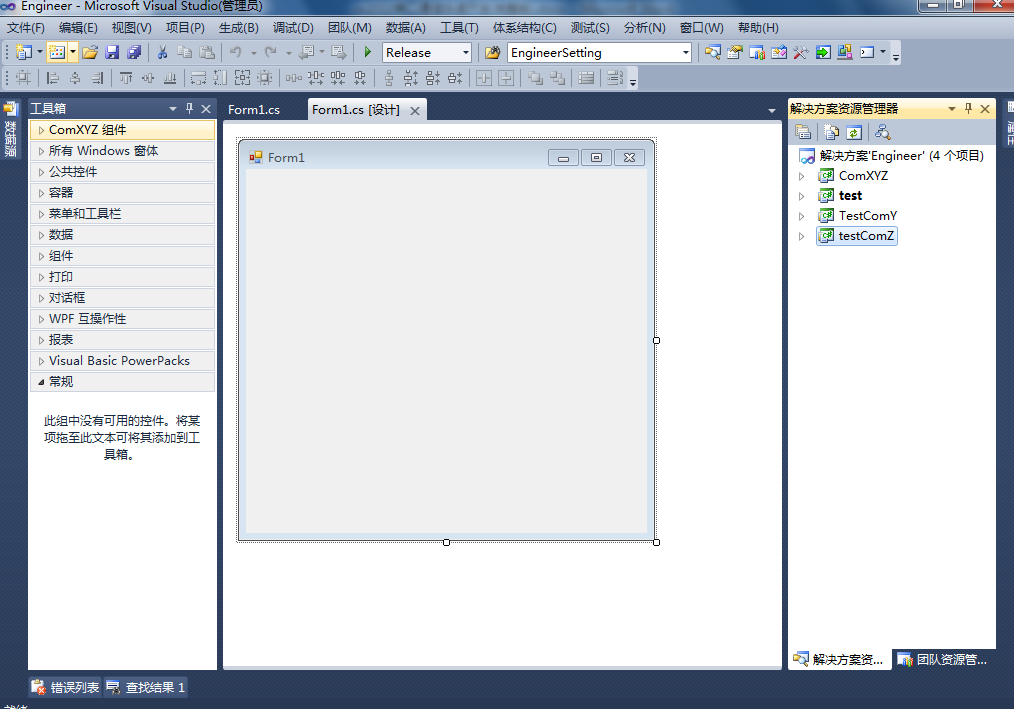
在ComY增加一个可视界面的封装，可见界面与ComX一个样子，但代码已经不一样。不需要个性化的配置界面推荐使用。

2.4、添加ComXYZ控件到工具箱

2.4.1、新建WINFORM工程（我这里是在现在解决方案里面添加一个工程）

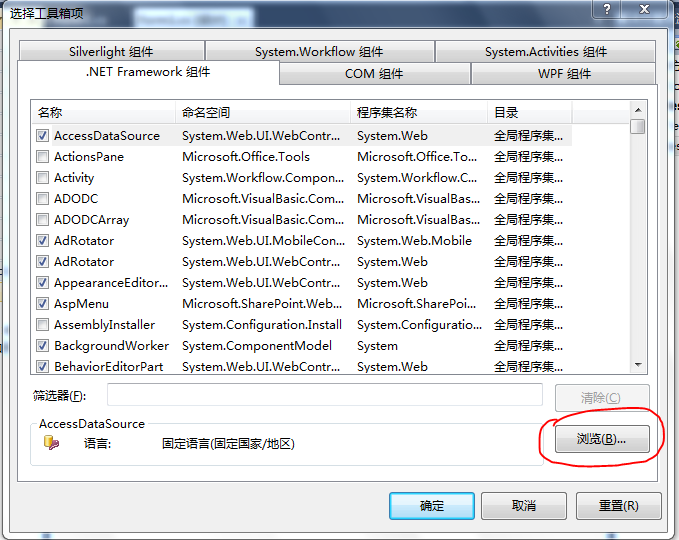


完成之后如下图

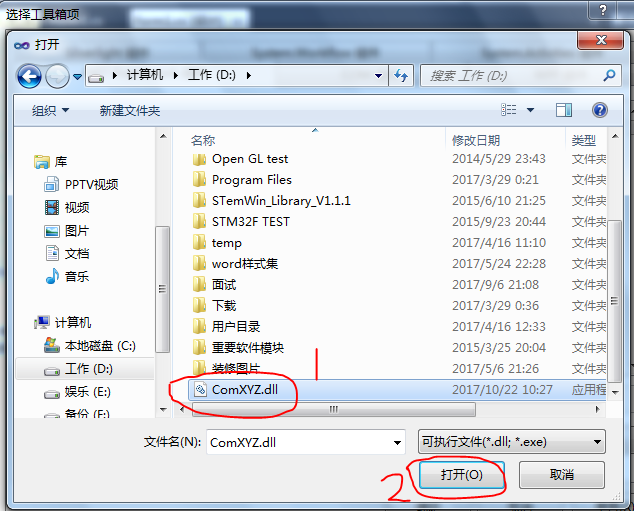


2.4.2、添加控件到工具箱中

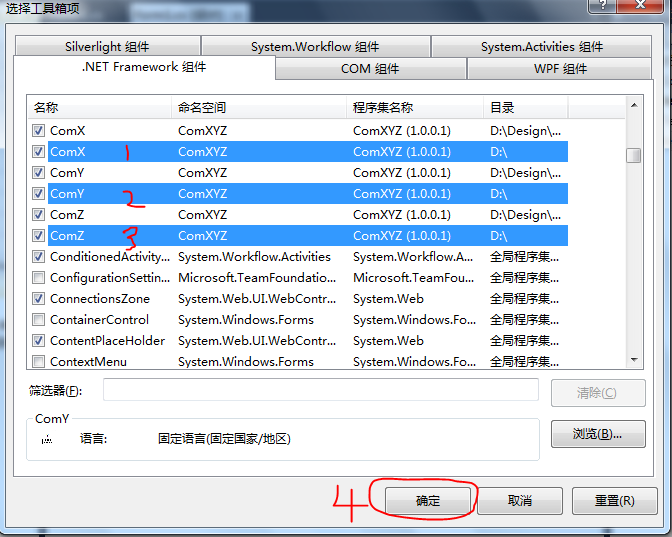
在上图工具箱的左下角常规里点击鼠标右键，选择弹出菜单的“选择项”弹出如下图对话框



点击红圈的浏览按钮，添加下载的控件文件“ComXYZ.DLL”



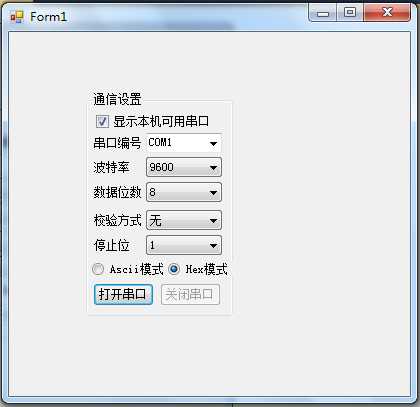
如下图所示



确保选择上图的1 2 3三个控件，之后点击确定按钮，工具箱如下图所示



由图可见3个控件已经添加到工具箱的常规选项卡中。之前它们的使用方法和软件自带的其他控件一样的使用方法。添加ComZ的运行效果如下图。

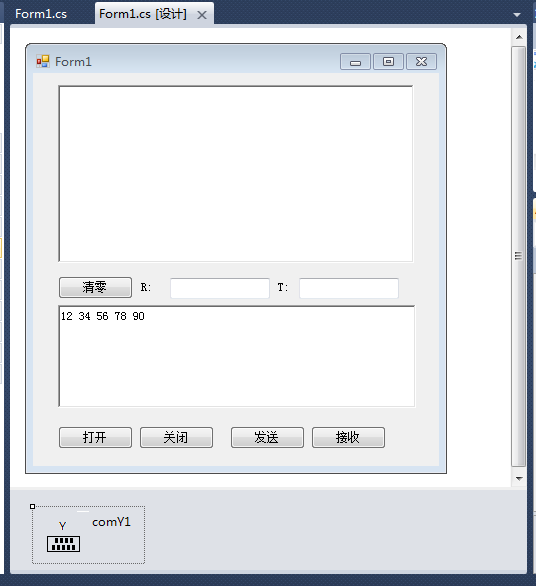


3、第一个例子（ComY）

3.1、新建项目TestComY，并拖放如下控件/组件并定义其名称如下表。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 控件类型 | 控件名称 | 用途 | 显示信息 | 备注 |
| 01 | RichTextBox | RxRichTextBox | 显示接收 |  | 接收区 |
| 02 | RichTextBox | TxRichTextBox | 输入发送数据 | 1234567890 | 发送区 |
| 03 | TextBox | RxTextBox | 显示接收数量 |  | 计数器值 |
| 04 | TextBox | TxTextBox | 显示发送数量 |  | 计数器值 |
| 05 | label | label1 | 指示接收计数区 | R | 计数器标识 |
| 06 | label | Label2 | 指示发送计数区 | T | 计数器标识 |
| 07 | Button | BtnClearCounter | 清空计数器值 | 清零 |  |
| 08 | Button | BtnComOpen | 打开串口 | 打开 |  |
| 09 | Button | BtnComClose | 关闭串口 | 关闭 |  |
| 10 | Button | BtnComTx | 发送数据区的数据 | 发送 |  |
| 11 | Button | BtnComRx | 接收并显示数据 | 接收 |  |
| 12 | comY | comY1 | 通信组件 |  | 通信服务 |

效果图如下



3.2、编写代码

双击界面中的五个按钮得到如下代码。

namespace TestComY

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void BtnComTx\_Click(object sender, EventArgs e)

{ }

private void BtnComRx\_Click(object sender, EventArgs e)

{ }

private void BtnComOpen\_Click(object sender, EventArgs e)

{ }

private void BtnComClose\_Click(object sender, EventArgs e)

{ }

private void BtnClearCounter\_Click(object sender, EventArgs e)

{ }

}

}

3.3、打开串口

private void BtnComOpen\_Click(object sender, EventArgs e)

{

comY1.SetPortPara("COM1",115200);

comY1.Open();

BtnComOpen.Enabled = false;

BtnComClose.Enabled = true;

}

3.4、关闭串口

private void BtnComClose\_Click(object sender, EventArgs e)

{

comY1.Close();

BtnComClose.Enabled = false;

BtnComOpen.Enabled = true;

}

3.5、发送数据及显示发送数据量

private void BtnComTx\_Click(object sender, EventArgs e)

{

comY1.Send(TxRichTextBox.Text);

TxTextBox.Text = comY1.TxCounterVal.ToString();

}

3.6、接收数据及显示接收数据量

private void BtnComRx\_Click(object sender, EventArgs e)

{

if (comY1.IsGotNewData() == true)

RxRichTextBox.Text += comY1.GetRxHexWithBlankString();

RxTextBox.Text = comY1.RxCounterVal.ToString();

}

3.7、清空计数器

private void BtnClearCounter\_Click(object sender, EventArgs e)

{

RxTextBox.Text = "";

TxTextBox.Text = "";

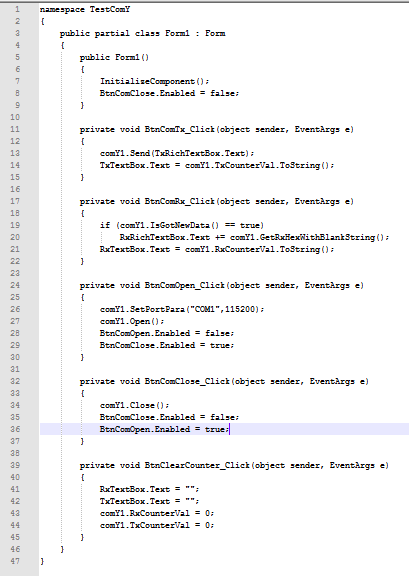
comY1.RxCounterVal = 0;

comY1.TxCounterVal = 0;

}

3.8、代码量

实现基本功能全部代码不到50行，如下图

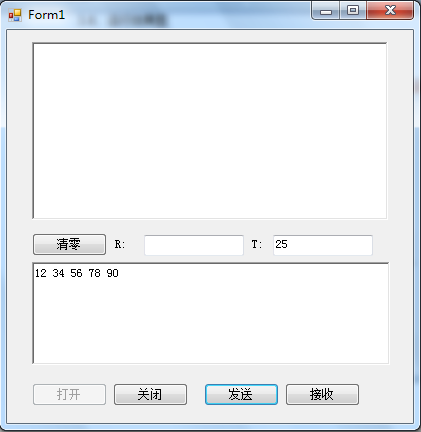


3.8、运行效果图

3.8.1、打开串口



3.8.2、发送数据



3.8.3、接收数据



3.8.4、关闭串口点发送



3.9、发送接收中文及ASCII

3.9.1、打开串口修改如下

private void BtnComOpen\_Click(object sender, EventArgs e)

{

comY1.SetPortPara("COM1",115200);

comY1.IsHexMode = false; //添加代码

comY1.Open();

BtnComOpen.Enabled = false;

BtnComClose.Enabled = true;

}

3.9.2、接收代码修改如下

private void BtnComRx\_Click(object sender, EventArgs e)

{

if (comY1.IsGotNewData() == true)

{

if (comY1.IsHexMode==true) //添加代码

RxRichTextBox.Text += comY1.GetRxHexWithBlankString();

else

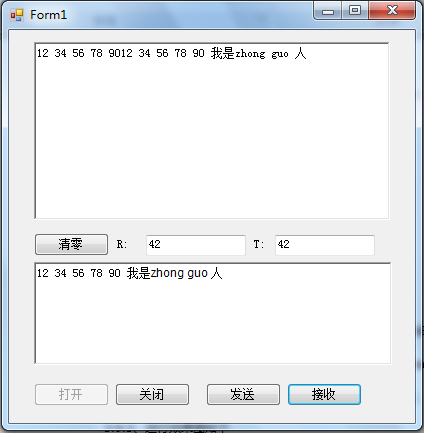
RxRichTextBox.Text += comY1.GetRxAsciiString(); //添加代码

}

RxTextBox.Text = comY1.RxCounterVal.ToString();

}

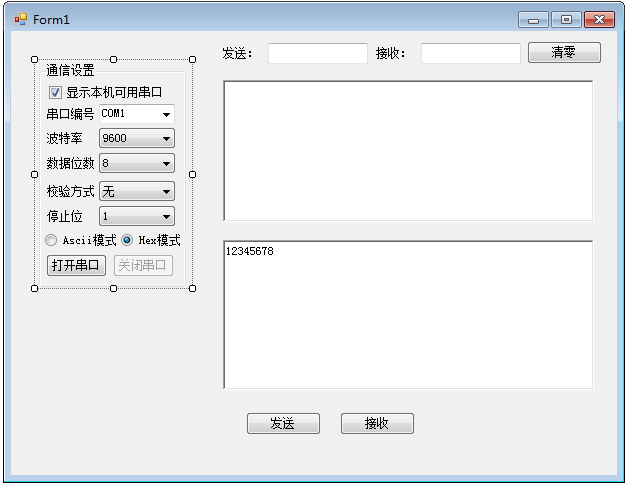
3.9.3、运行效果图如下



4、第2个例子（ComZ）

4.1、新建窗体工程（testComZ）及添加控件如下表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 控件类型 | 控件名称 | 用途 | 显示信息 | 备注 |
| 01 | RichTextBox | RxBox | 显示接收 |  | 接收区 |
| 02 | RichTextBox | TxBox | 输入发送数据 | 12345678 | 发送区 |
| 03 | TextBox | RxTextBox | 显示接收数量 |  | 计数器值 |
| 04 | TextBox | TxTextBox | 显示发送数量 |  | 计数器值 |
| 05 | label | Label2 | 指示接收计数区 | 发送 | 计数器标识 |
| 06 | label | Label3 | 指示发送计数区 | 接收 | 计数器标识 |
| 07 | Button | btnClean | 清空计数器值 | 清零 |  |
| 08 | Button | btnTx | 发送数据区的数据 | 发送 |  |
| 09 | Button | BtnRx | 接收并显示数据 | 接收 |  |
| 10 | comZ | comZ1 | 通信组件 |  | 通信服务 |

效果图如下

4.2、编写代码

双击界面上的按钮得到如下代码

namespace testComZ

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnTx\_Click(object sender, EventArgs e)

{ }

private void BtnRx\_Click(object sender, EventArgs e)

{ }

private void btnClean\_Click(object sender, EventArgs e)

{ }

}

}

4.3、发送数据及显示发送数据量

private void btnTx\_Click(object sender, EventArgs e)

{

UInt32 txcounter;

comZ1.Send(TxBox.Text);

txcounter=comZ1.TxCounterVal;

TxTextBox.Text = txcounter.ToString();

}

4.4、接收数据及显示接收数据量

private void BtnRx\_Click(object sender, EventArgs e)

{

if (comZ1.IsGotNewData() == true)

{

if (comZ1.IsHexMode == true)

RxBox.Text += comZ1.GetRxStringAndHexWithBlank();

else

RxBox.Text += comZ1.GetRxAsciiString();

RxTextBox.Text = comZ1.RxCounterVal.ToString();

}

}

4.5、清零计数器

private void btnClean\_Click(object sender, EventArgs e)

{

comZ1.TxCounterVal = 0;

comZ1.RxCounterVal = 0;

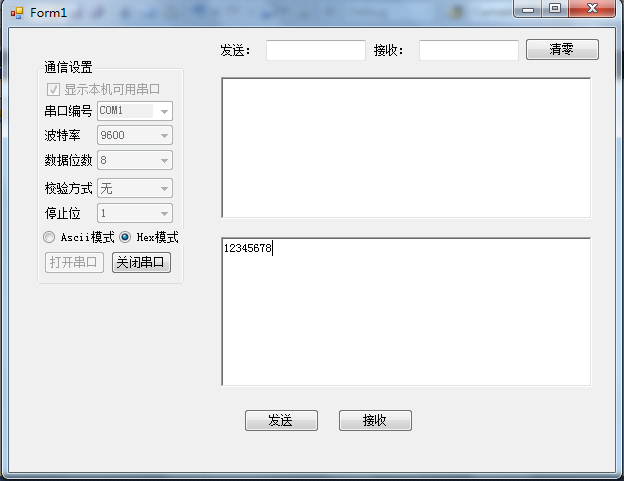
TxTextBox.Text = comZ1.TxCounterVal.ToString();

RxTextBox.Text = comZ1.RxCounterVal.ToString();

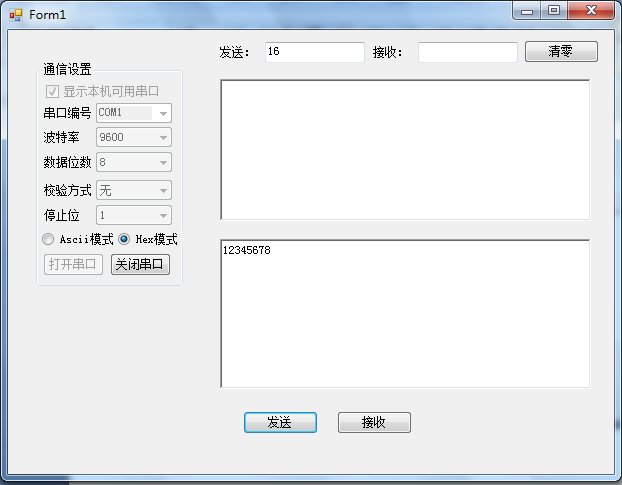
}

4.6、运行效果图

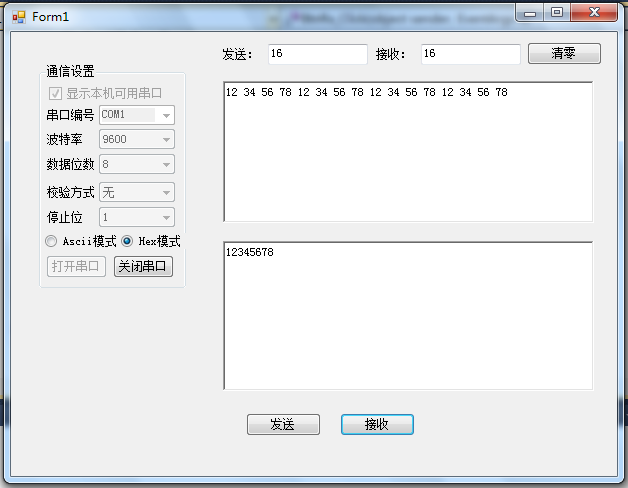
4.6.1、打开串口



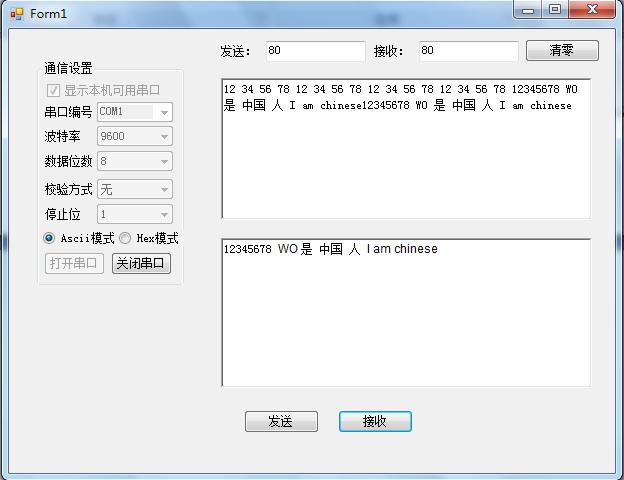
4.6.2、发送数据



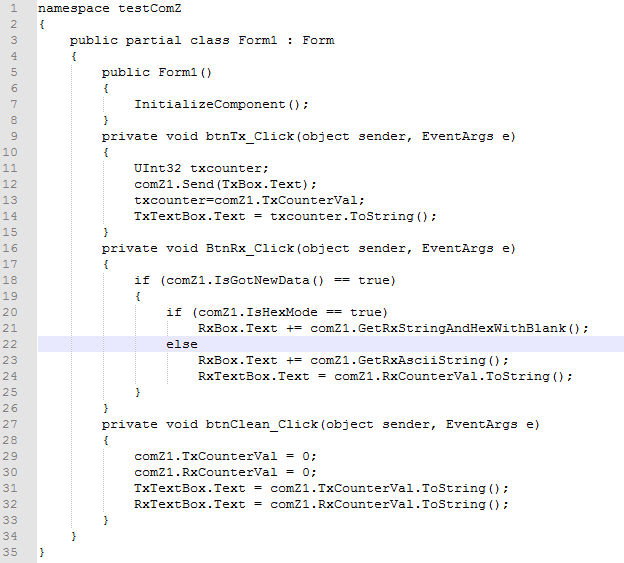
4.6.3、接收数据



4.6.4、发送中文及ASCII



4.7、代码量（全部仅仅35行）如下图



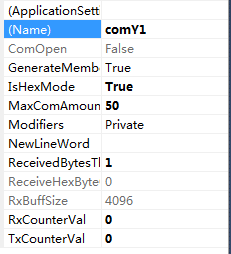
4.8、由代码量可见，整个基本功能的实现三分钟内即可完成。

5、ComXYZ接口函数

5.1、ComY

这是一个组件，基于SerialPort进行二次开发。

5.1.1、属性



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 属性名称 | 含义 | 值/类型 | 备注 |
| 1 | Name | 名称 | 字符串 | 自定义 |
| 2 | IsOpen | 串口打开 | TRUE/FALSE | 只读 |
| 3 | IsHexMode | 16进制模式 | TRUE/FALSE | 读写 |
| 4 | RxCounterVal | 接收计数器 | 32位变量 | 读写 |
| 5 | TxCounterVal | 发送计数器 | 32位变量 | 读写 |

5.1.2、方法

5.1.2.1、串口设置方法

public string[] GetComNameList()

public void SetPortName(string Name)

public void SetPortBaudRate(int BaudRate)

public void SetPortDataBits(int Databits)

public void SetPortParityBits(Parity Paritybits)

public void SetPortStopBits(StopBits Stopbits)

public void SetPortPara(string portName, int BaudRate)

public void SetPortPara(int Databits, Parity Paritybits, StopBits Stopbits)

public void SetPortPara(string portName, int BaudRate, int Databits, Parity Paritybits, StopBits Stopbits)

public void SetPortPara(string portName, int BaudRate, int Databits, int Paritybits, int Stopbits)

5.1.2.2、串口打开与关闭

public void Open()//打开失败弹窗提示

public void Close()

5.1.2.3、发送接收数据

public bool Send(string SendText)

public bool Send(bool HexMode, string SendText)

自动检测串口是否打开，两个函数均可发送16进制数与ASCII代码

5.1.2.4、接收数据

public bool GetHexArrayFromRxBuff(byte[] buff, ref int count)

提取16进制数据用于数据分析，无数据时返回false。

public string GetRxString()

提取符串形式的数据，无数据时返回空。

public string GetRxStringAndHexWithBlank()

提取字符串形式的数据，16进制数据自动插入空格，无数据时返回空。

5.1.2.5、辅助方法

public bool IsGotNewData()查询是否接收到新数据

public void ForceClearHexBuffer()强制清空16进制数据缓存

5.2、ComZ

这是一个复合组件，基于ComY进行再次开发，目的是简化ComY的使用，有一个可视的固定界面组合，如下图。



由图可见串口的基本配置已经实现。

5.2.1、属性

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 属性名称 | 含义 | 值/类型 | 备注 |
| 1 | Name | 名称 | 字符串 | 自定义 |
| 2 | IsOpen | 串口打开 | TRUE/FALSE | 只读 |
| 3 | IsHexMode | 16进制模式 | TRUE/FALSE | 读写 |
| 4 | RxCounterVal | 接收计数器 | 32位变量 | 读写 |
| 5 | TxCounterVal | 发送计数器 | 32位变量 | 读写 |
| 6 | MaxComAmount | 串口编号最大值 | 16位变量 | 读写默认50 |

5.2.2、开放方法

public void Send(string SendText)

public void Send(bool HexMode, string SendText)

public bool IsGotNewData()

public bool GetHexArrayFromRxBuff(byte[] buff, ref int count)

public string GetRxAsciiString()

public string GetRxHexString()

public string GetRxStringAndHexWithBlank()

public void ForceClearHexBuffer()

6、获取测试例子，请加QQ群321903608

201701022