

## C#使用 COM 组件接口进行串口操作

成都英创信息技术有限公司

2007 年 09 月

英创 ARM9 工控主板均提供了多个串口供客户进行通讯控制使用，包括标准的 RS232 和 TTL 电平的 COM 接口(经过简单的电平转换即可成为标准的 RS232/RS485 或者 RS422)。由于英创 ARM9 嵌入式主板采用 WinCE 操作系统，很多客户都选择了界面友好、开发便利的 C#语言(使用 Visual Studio 2005.NET 集成开发环境)；在 VS2005 中，微软提供了一个标准的串口类 SerialPort 供客户使用，该串口类可以灵活方便地进行串口设置、收发等操作，英创也提供了基于该串口类的 C#例程供客户参考。但是，微软所提供的串口类虽然方便灵活，但是没有提供 RTS 控制的功能，而客户在进行工业现场通讯的时候，往往是需要对 RTS 进行设置的。为了解决这个问题，英创公司现在提供了操作串口的 COM 组件接口；用户只需要在目标平台上进行组件注册，然后在 C#程序开发的时候，对该 COM 组件添加引用，即可在程序里面自由地通过对象操作来使用函数。

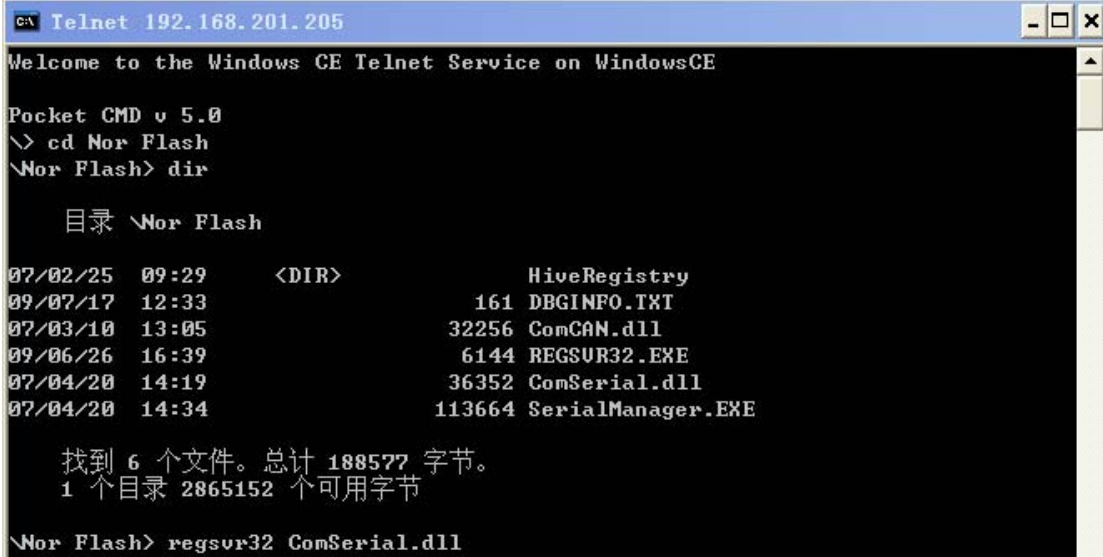
注：英创提供的可以供 C#开发使用的串口 COM 组件与 eVC 开发使用的串口 COM 组件在成员数据类型上有显著区别，不能互换使用。

下面我们就以 EM9000 嵌入式主板为实例，在 C#使用 COM 组件接口进行串口操作作一个说明：

### 一、在目标嵌入式主板上进行 COM 组件注册

串口 COM 组件的核心文件就是英创提供的 ComSerial.dll 动态链接库文件，首先需要将它安装在目标嵌入式工控主板上注册。用户可以使用 Visual Studio Remote Tools 提供的“远程文件查看器”连接目标嵌入式工控主板，并将英创提供的 ComSerial.dll 和 REGSVR32.exe 这两个文件下载到目标主板的 Flash 存储器上去(这里假设下载到 EM9000 的 Nor Flash)。然后在开发主机 PC 上使用 Telnet 连接目标工控主板(如图)，进入放置刚才下载这两个文件的 Flash 存储器目录，运行：

REGSVR32 ComSerial.dll



```
CA Telnet: 192.168.201.205
Welcome to the Windows CE Telnet Service on WindowsCE

Pocket CMD v 5.0
\> cd Nor Flash
\Nor Flash> dir

    目录 \Nor Flash

07/02/25  09:29      <DIR>          HiveRegistry
09/07/17  12:33             161  DBGINFO.TXT
07/03/10  13:05          32256  ComCAN.dll
09/06/26  16:39           6144  REGSVR32.EXE
07/04/20  14:19          36352  ComSerial.dll
07/04/20  14:34         113664  SerialManager.EXE

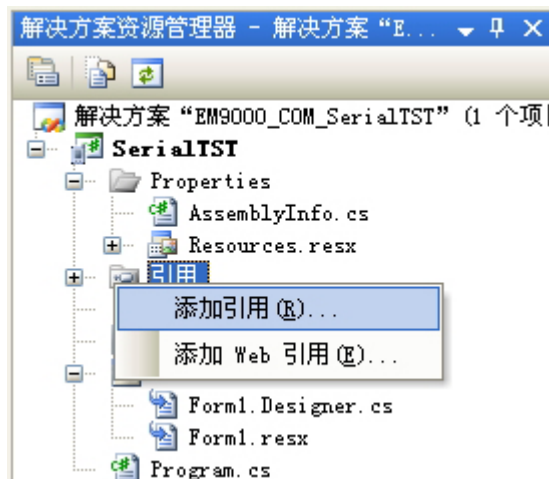
找到 6 个文件。总计 188577 字节。
1 个目录 2865152 个可用字节

\Nor Flash> regsvr32 ComSerial.dll
```

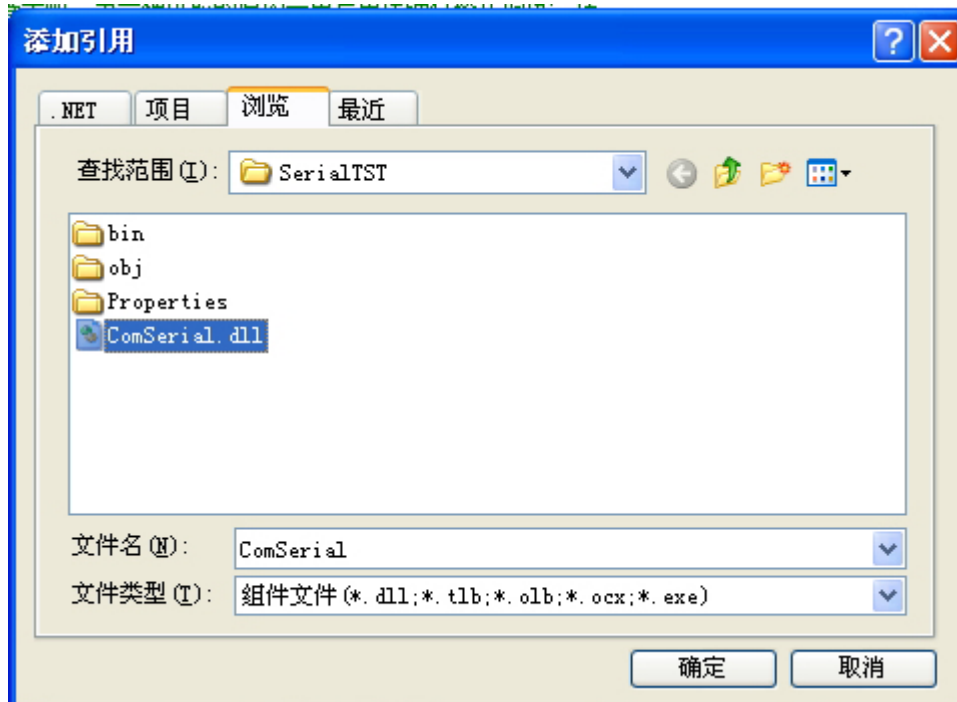
即可完成 COM 组件注册。

## 二、在 C# 工程中添加 COM 组件引用

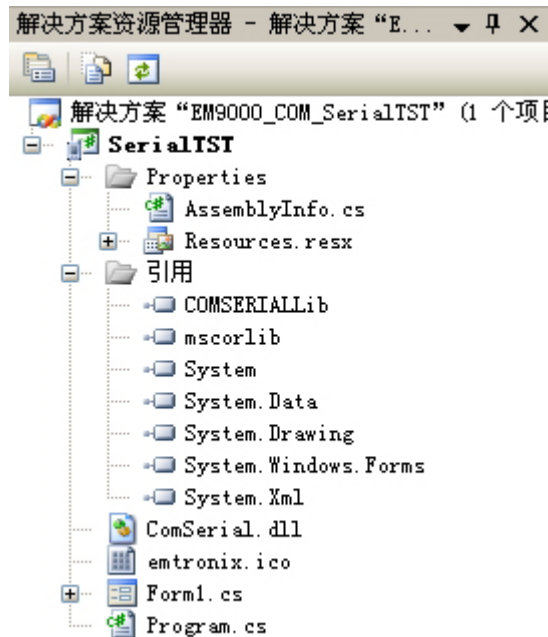
接下来可以进入 C# COM 组件开发。首先将 ComSerial.dll 文件复制到用户的 C# 工程目录下,然后在 Visual Studio 2005.NET 集成开发环境中打开用户工程。为了使用 ISA 总线 COM 组件,必须为该工程添加外部引用;用户可在集成开发环境的右侧的“解决方案资源管理器”中找到“引用”,对其单击鼠标右键,然后单击“添加引用”(如图)。



在弹出的“添加引用”对话框中,选择“浏览”选项卡,并在你的工程目录下找到你刚才复制过来的 ComSerial.dll 文件,选择它,并单击“确定”。



这时,你已经完成了添加工程中的 COM 组件的引用工作,这时,你可以在集成开发环境的右侧的“解决方案资源管理器”中看到了一个名称为 COMSERIALLib 的引用,同时, ComSerial.dll 也加入到了你的工程中(如图)。



### 三、在应用程序开发中使用串口 COM 组件

首先，为了使应用程序能直接调用相关的类和函数，必须在程序的起始 `using` 语句中添加：

```
using COMSERIALLib;
```

程序中对于串口的操作完全围绕 `CoSerialClass` 这个类来进行的。用户需要首先定义这个类的对象实体，然后通过对对象里面的方法（函数）来进行所需要的操作。下面大致做一些讲解：

#### （1）定义类的对象

```
CoSerialClass COM_myserialPort = new CoSerialClass();
//定义一个 CoSerialClass 串口类对象 COM_myserialPort
```

#### （2）使用对象的方法函数

该串口类的对象实体主要包括四个函数和两个事件，下面我们着重就 `OpenPort`、`ClosePort`、`WritePort`、`ReadPort` 四个函数和 `OnRecieve` 事件进行举例说明（举例中所使用的类对象已经在上文中定义）：

- `int OpenPort(uint portNo, uint baud, sbyte parity, uint dataBits, uint stopBits, uint rtsCtr);`

打开串口函数，使用示例如下：

```
uint portNO = 1;    //串口1
uint myrate = 9600;  //波特率
COM_myserialPort.OpenPort(portNO, myrate, Convert.ToSByte('n'), 8, 1, RTS_CONTROL_TOGGLE);
//参数依序： 串口号、波特率、校验位（无）、数据位（8）、停止位（1）、RTS
```

注意 `RTS` 常数应该在程序开始就定义，我们一般做如下定义：

```
const uint RTS_CONTROL_DISABLE = 0x00;
const uint RTS_CONTROL_ENABLE = 0x01;
const uint RTS_CONTROL_HANDSHAKE = 0x02;
const uint RTS_CONTROL_TOGGLE = 0x03;
```

- `OnRecieve` 事件

我们可以利用 COM 组件提供的 OnReceive 事件（当串口有数据收到的时候发生）来启动一个事件函数来读取串口数据。OnReceive 事件可以放在程序定义串口对象实例之后程序刚开始的某个地方：

```
COM_myserialPort.OnReceive += new _ICoserialEvents_OnReceiveEventHandler(COM_myserialPort_OnReceive);
//当串口有数据收到时，启动 COM_myserialPort_OnReceive 事件函数
```

- **byte ReadPort()**

读取串口数据函数，往往在 OnReceive 事件函数里使用，注意该函数是一次读取一个 byte，所以一次需要将数据读取完毕（结束标志为 0xff）。使用示例如下：

```
private void COM_myserialPort_OnReceive() //串口有数据收到
{
    byte recieved_byte = 0;
    byte[] RCV_bytes = new byte[1024];
    int i=0;
    for (i=0;i<1024;i++ )
    {
        recieved_byte = COM_myserialPort.ReadPort();
        if (recieved_byte == 0xff) break;
        RCV_bytes[i] = recieved_byte;
    }
    //此处可对收到的数据进行进一步处理
}
```

- **int WritePort(int elems, byte[] buf);**

发送串口数据函数，使用示例如下：

```
byte[] sendbytmessage = Encoding.Default.GetBytes(textBox1.Text); //将某文本框数据转为byte数组
COM_myserialPort.WritePort(sendbytmessage.Length, sendbytmessage); //发送
```

- **void ClosePort();**

关闭串口函数，直接使用即可关闭串口。

用户还可以参考英创提供的 C#串口 COM 组件例程，进行实际的应用开发。