

实验一：串口调试助手开发

一、实验器材

每人一台电脑，安装好同样版本的 Visual studio。

二、实验步骤

1、创建工作环境

1.1 打开 Visual studio 2017

串口调试助手可以有多种方法开发，当前选择速度最快的方式进行。首先打开 Visual studio，如图 4-1 所示。

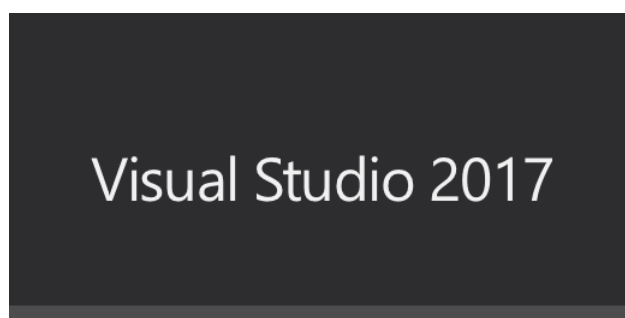


图 4-1 启动 Visual studio 2017

1.2 新建窗体应用

在新建的项目中，选择 Visual C# 中的 Windows 窗体应用(.NET Framework)，选择好保存路径和文件名(此处默认的命名为 SerialTool)，如图 4-2 所示。

稍作等待以后，应用程序就创建完了。相关界面介绍如图 4-3 所示。常用到的界面为解决方案资源管理器、属性、工具栏等。

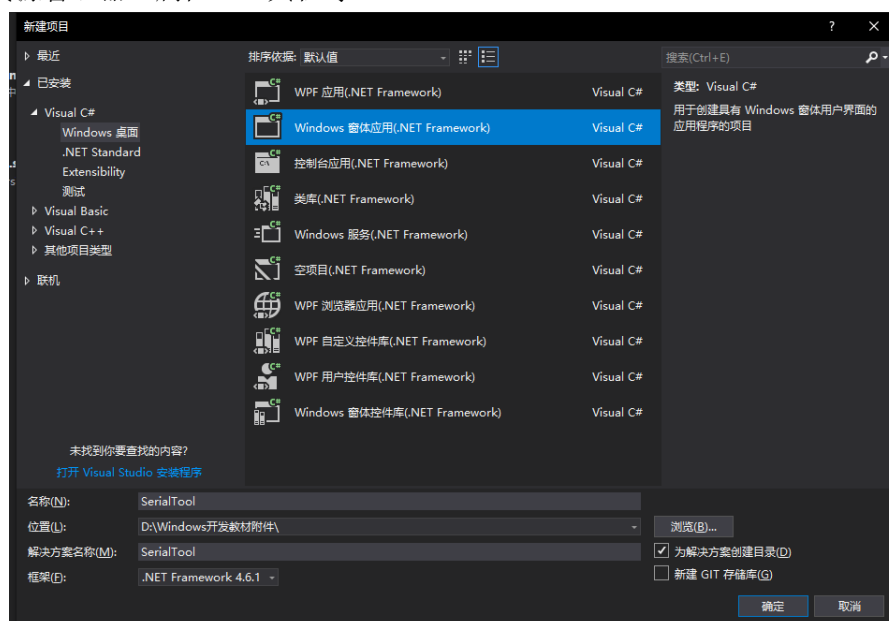


图 4-2 创建新工程

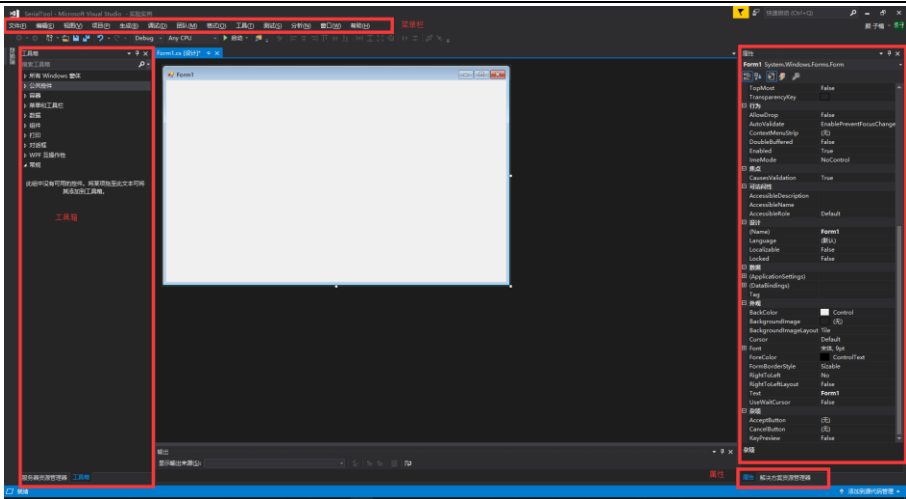


图 4-3 界面介绍

1.3 创建串口助手图形界面

工具栏中提供一般应用程序常用的控件，用于设计应用程序的 GUI 界面。在此项目工程中，我们创建一个比较经典的串口助手界面。它包含一个输入窗口，一个输出窗口，一块串口设置区域等多个功能区组成。

在属性页面中，包含我们选中的控件的属性信息，常用信息在属性页面中即可直接操作，比如背景色、字体大小、名字等，常用信息如表 4-4 所示；当涉及到特殊属性时，会单独列出。

表 4-4 常用信息表

属性信息	备注
Name	用来获取或设置窗体的名称，在应用程序中可通过 Name 属性来引用窗体
Text	该属性是一个字符串属性，用来设置或返回在窗口标题栏中显示的文字
BackColor	用来获取或设置窗体的背景色
Enabled	用来获取或设置一个值，该值指示控件是否可以对用户交互作出响应。如果控件可以对用户交互作出响应，则为 true；否则为 false。默认值为 true
Items	用于存放列表框中的列表项，是一个集合。通过该属性，可以添加列表项、移除列表项和获得列表项的数目

相关步骤如下：

- ◆ 在工具栏中的公共控件中选择 TextBox 控件拖拽到 Form1 窗口中，选择好合适位置，点击右上角的小三角，选择 MultLine 复选框，再选择合适大小作为输入框；
- ◆ 使用同样的方法，再拖拽一个 TextBox 控件作为输出框；
- ◆ 拖拽容器中的 GroupBox 放置在左边窗口左边，作为配置串口的一个 Group，里面存放相关串口配置功能；
- ◆ 在输入框旁边放置一个 Button 控件用于作为发送按键，将(Name)属性更改为“bt_Send”如图 4-5 所示；

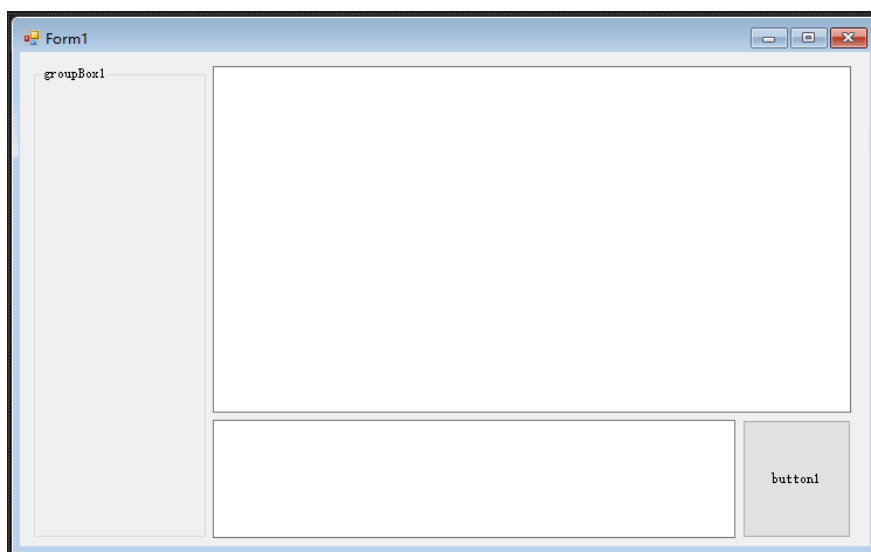


图 4-5 串口初级界面

- ◆ 拖拽一个 Combobox 控件到 groupBox1 中，作为串口入口选择，将(Name)属性更改为“cB_Portname”；
- ◆ 拖拽一个 Combobox 控件到 groupBox1 中，作为波特率选择，将(Name)属性更改为“cB_Baud”；
- ◆ 拖拽一个 Button 到 groupBox1 中，作为连接按钮，在属性框中，将 Text 属性更改为“连接”，将(Name)属性更改为“bt_Connect”；
- ◆ 拖拽一个 Button 到 groupBox1 中，作为清除按钮，在属性框中，将 Text 属性更改为“清除”将(Name)属性更改为“bt_Clear”；
- ◆ 拖拽多个 Label 控件作为 辅助性说明文字，如图 4-6 所示；

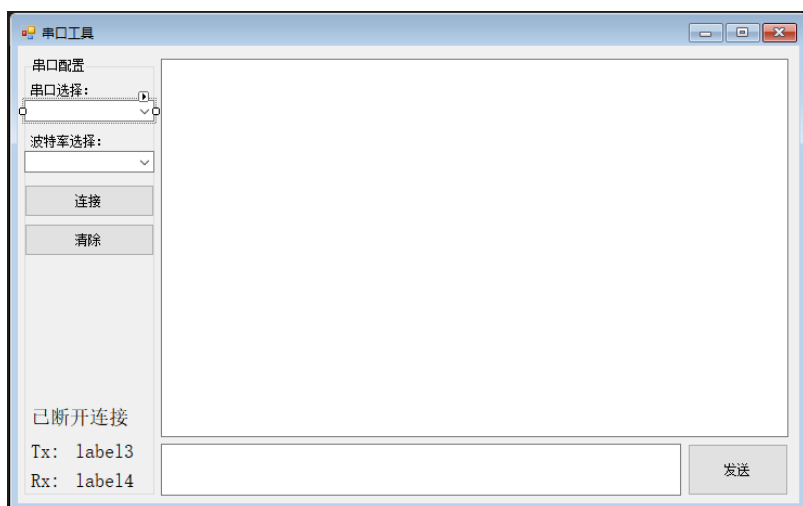


图 4-6 添加 Label 后页面

- ◆ 选择 cB_Baud 控件，点击 Items 属性后的集合，将常用波特率添加进去，如图 4-7 所示。添加集合后，即可在 ComboBox 控件中选择输入的 Items。
- ◆ 拖拽一个 SerialPort 模块到界面中，将(Name)属性更改为“Serial”；
- ◆ 拖拽一个 Timer 控件，将(Name)属性更改为“tim_Update”用以更新 GUI 状态；至此 GUI 的简单界面即绘制完毕了，接下来要编写事件代码。

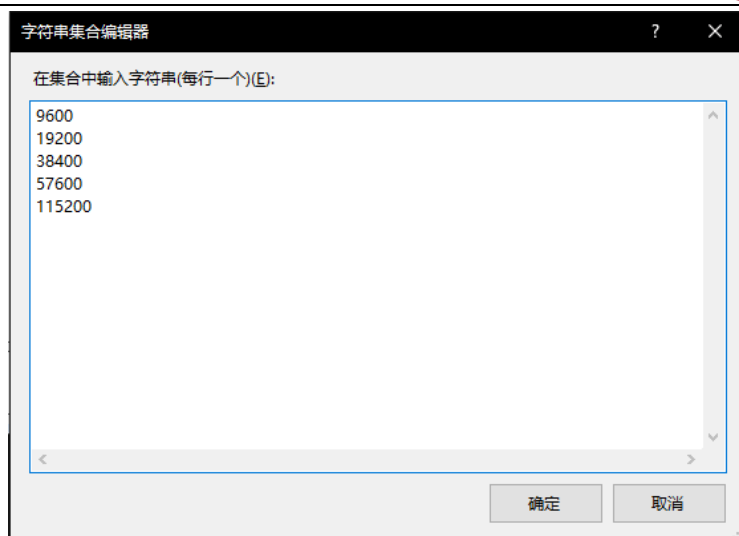


图 4-7 添加常用波特率

1.4 编写事件

控件通过事件连接到相关函数执行，此处需要为图形界面的三个按钮创建鼠标单击事件，在设计页面双击按钮控件即可。

双击按钮创建事件后，会进入函数编辑页面，即单击按钮后会执行此函数。例如连接按钮的单击事件函数如图 4-8 所示。

```
private void bt_Connect_Click(object sender, EventArgs e)
{
    ...
}
```

图 4-8 连接事件

还需要对 SerialPort 创建接收事件，点击 SerialPort，在属性页面的事件选项卡中的 DataReceived 上双击即可创建默认事件，如图 4-9 所示。这是对串口接收中断进行处理的函数，此函数稍有特别，此函数运行于单独的线程中，有些操作受限，此情况在遇到时再作说明。

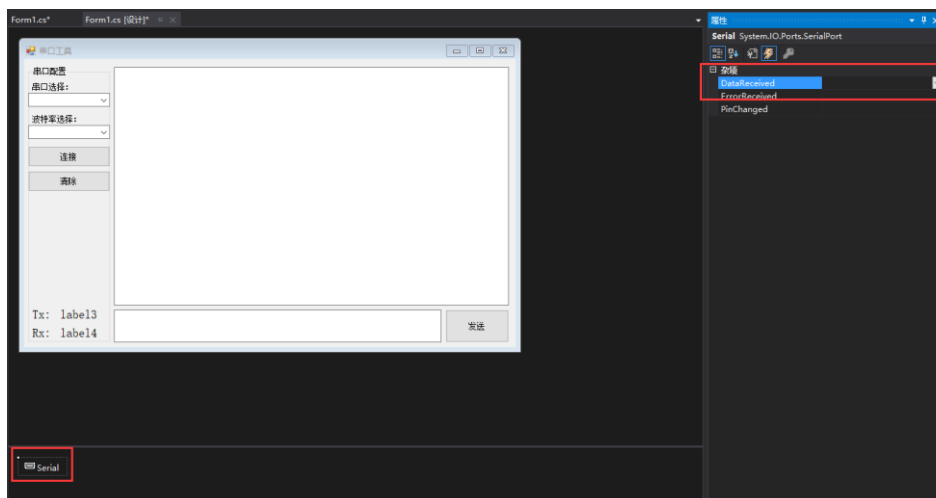


图 4-9 为 SerialPort 创建接收事件

2、打开和关闭串口

双击连接按钮，进入连接按钮单击事件函数。在编写代码之前，首先要知道串口通信的一些基本信息，根据这些信息，再编写控制逻辑。

首先，串口需要被打开才能进行收发工作，当串口打开失败时，收发工作将无法进行，由于串口连接可能并不牢靠，需要在程序上编写一些提升健壮性的防护措施，用以保证程序即便在出错情况下依然能正常运行的方法。

双击连接按钮，进入连接按钮事件，编入以下代码，尝试打开串口。

```
private void bt_Connect_Click(object sender, EventArgs e)
{
    if (!Serial.IsOpen)//判断当前串口是否打开
    {
        Serial.BaudRate = Convert.ToInt16(cB_Baud.Text);
        Serial.PortName = cB_Portname.Text;
        Serial.Open();
    }
    else
    {
        Serial.Close();
    }
}
```

打开串口需要将 Serial.PortName 属性赋值为 COMx，此处 COM 口需要查看设备管理器获得。将波特率和其他相关信息设置以后，运行 Open() 函数，若打开正常，则串口打开，若失败则抛出异常。

3、发送和接收

本节会给出串口的发送和接收代码，点击发送按钮，即可发送数据，自动显示接收到的数据：

3.1 发送

在类 Form1 中，创建两个 int 类型字段，用以记录串口收发计数；

双击发送按钮，在发送事件函数中，键入以下代码用以发送：

```
private void bt_Send_Click(object sender, EventArgs e)
{
    if(Serial.IsOpen)
    {
        Serial.Write(tB_Send.Text);
        serialtx_count += tB_Send.TextLength;
    }
}
```

3.2 接收

在默认创建的接收事件函数中键入以下代码：

```
private void Serial_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    byte[] Rx_Buff = new byte[Rx_Buff_Size];

    int ReadByte_Size = Serial.Read(Rx_Buff, 0, Serial.BytesToRead);

    if (ReadByte_Size == 0)
        return;

    serialrx_count += ReadByte_Size;
    tB_Received.Text += Encoding.UTF8.GetString(Rx_Buff);
}
```

此函数会在接收到数据时自动调用，其中 Serial.Read 函数会读取数据到 Rx_Buff 数组中，并返回读取到的数据长度，最后在接收 TextBox 中显示接收到的数据长度。

4、GUI 状态更新

在显示界面中，需要一个一直刷新显示界面数据的函数来查询和更新状态信息，比如接收数据长度和发送数据长度等。Timer 定时器可以作为一个固定时间触发的函数来刷新 GUI 上的数据。在 tim_Update 定时器的事件函数下键入以下代码：

```
private void tim_Update_Tick(object sender, EventArgs e)
{
    if (Serial.IsOpen) //更新串口连接信息
    {
        lb_Connect.Text = "已连接";
    }
    else
    {
        lb_Connect.Text = "已断开连接";
    }

    //更新接收和发送的数据计数
    lb_RxCount.Text = serialrx_count.ToString();
    lb_TxCount.Text = serialtx_count.ToString();
}
```

6、清除信息

在串口通信中，时常需要清除一些信息来更新界面，此时要清除的信息包括发送计数、接收计数、接收信息界面、发送信息界面 4 项。

双击清除按钮，在清除事件函数中键入以下代码：

```
private void bt_Clear_Click(object sender, EventArgs e)
{
    serialrx_count = 0;
    serialtx_count = 0;

    tB_Received.Text = "";
    tB_Send.Text = "";
}
```

其余详情请翻看源码。