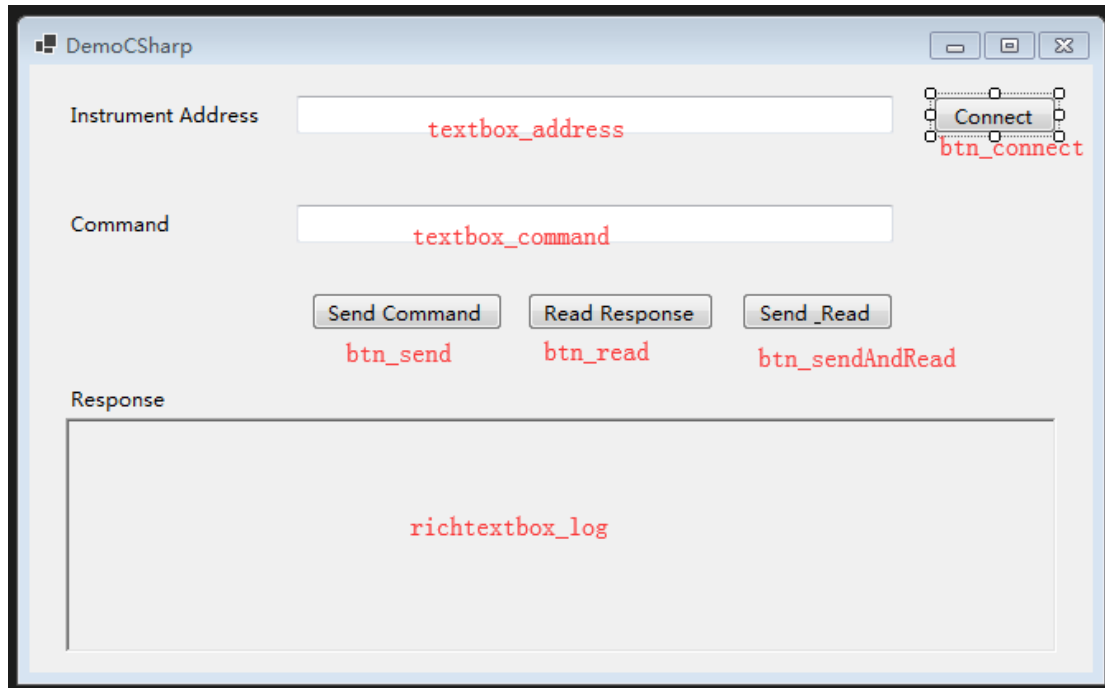


C# programming example

The development environment used in this example is VS2019, net5.0, c#

This example implements functions: find instruments, connect instruments, send commands, and read instrument return values.

1. Run VS2019 to create a new winform project.
2. Modify the Form1 form automatically generated by the system and add the following controls



3. Right-click on the project and select [Add]->[Existing Item], and select visa32.cs and VisaInstrument.cs. These two files encapsulate the C# call of the visa dynamic library in the C# language, and you can encapsulate the function by referencing the file later. to communicate with the device.

4. Add a visa communication instance to the form1 class

```
public partial class Form1 : Form
{
    VisaInstrument visaInstrument = null;
    public Form1()
    {
        InitializeComponent();
        visaInstrument = new VisaInstrument();
    }
}
```

Add function function of buttonConnect button

```

private void btn_connect_Click(object sender, EventArgs e)
{
    var userDefineAddress = textBox_address.Text;
    if (userDefineAddress == string.Empty)
    {
        var addressList = visaInstrument.FindDevices();
        if(addressList.Any()==false)
        {
            AddLog("no any device find");
        }
        foreach(var address in addressList)
        {
            if (visaInstrument.OpenSession(address) == true)
            {
                visaInstrument.WriteBytes("*IDN?\n");
                AddLog("*IDN?\n");
                var readValue = visaInstrument.ReadBytes();
                AddLog(readValue);
                if (readValue.Contains("HDM"))
                {
                    textBox_address.Text = address;
                    AddLog($"connected device {address}");
                    return;
                }
            }
        }
        AddLog("no any device connected");
    }
    else
    {
        var connectState = visaInstrument.OpenSession(userDefineAddress);
        if (connectState)
        {
            AddLog($"device {userDefineAddress} connect success");
        }
        else
        {
            AddLog($"device {userDefineAddress} connect faild");
        }
    }
}

```

Add function function of buttonSend button

```

private void btn_send_Click(object sender, EventArgs e)
{
    var sendStr = $"{textBox_command.Text}\n";
    visaInstrument.WriteBytes(sendStr);
    AddLog(sendStr);
}

```

Add function function of buttonRead button

```
private void btn_read_Click(object sender, EventArgs e)
{
    var readStr = visaInstrument.ReadBytes();
    AddLog(readStr);
}
```

Add function function of buttonSendAndRead button

```
private void btn_sendAndRead_Click(object sender, EventArgs e)
{
    var sendStr = $"{textBox_command.Text}\n";
    visaInstrument.WriteBytes(sendStr);
    AddLog(sendStr);
    var readStr = visaInstrument.ReadBytes();
    AddLog(readStr);
}
```

Add print log function

```
private void AddLog(string logStr)
{
    logStr = logStr.Replace("\n", "");
    var log = $"{DateTime.Now.ToString("HH:mm:ss fff")}:{logStr}\n";
    richTextBox_log.AppendText(log);
}
```

5. Compile and run, if the address bar is filled with the specified address, it will directly connect to the address device. If the address bar is empty, the program will automatically find all online devices and connect to the HDM3000 device.