

C# Serial Communications, AN008

This application note shows how to create a C# console application that can be used to send commands to Novatech Instruments' Signal Generators and display the response returned by the signal generator. The application consists of two code files, program.cs and SerialClass.cs that were created using Visual Studio 2019. Program.cs and SerialClass.cs are shown on pages 2 and 3 below.

The SerialClass.cs code creates a class named SerialClass() which makes use of the SerialPort() class from the C# library. It does this by creating an instance of SerialPort() named **_serialPort**. The SerialClass constructor then populates the **_serialPort** parameters Parity, StopBits, PortName, ReadTimeout and BaudRate with appropriate values. It then creates five functions named BytesToRead(), OpenPort(), ClosePort(), SendCommand() and ReadDataToList().

The Program.cs code provides an entry point for the application at the Main() function and then creates an instance of SerialClass() named **port** and opens the serial port by calling the SerialClass() function port.OpenPort(). Next the code creates a list variable named **results**. It then sends a "que" command to the serial port using the SerialClass() function port.SendCommand, waits 300 milliseconds and stores the value returned from the serial port in the **results** variable by calling the port.ReadDataToList() function.

Sometimes it takes more than 300 milliseconds for all the values to be returned by the serial port. For this reason the Program.cs code uses the ports.BytesToRead() function in a while loop to read any remaining values and store them in the **doubleCheckBuffer** list variable. Once the BytesToRead() function returns zero we know the serial port buffer is empty and the program.cs code then uses a foreach loop to display the returned values and then closes the port. The Console.ReadLine() command forces the command window to stay open so you can see the returned responses.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO.Ports;
using System.Threading;

namespace NT_Appnote
{
    class Program
    {
        static void Main(string[] args)
        {
            SerialClass port = new SerialClass();
            try
            {
                port.OpenPort();
                List<String> results = new List<string>();
                try
                {
                    port.SendCommand("que");
                    Thread.Sleep(300);
                    results = port.ReadDataToList();
                    while (port.BytesToRead() != 0)
                    {
                        List<String> doubleCheckBuffer = new List<String>();
                        doubleCheckBuffer = port.ReadDataToList();
                        results.AddRange(doubleCheckBuffer);
                    }
                    if(results.Count() == 0)
                    {
                        Console.WriteLine("Error! Please verify that the COM port is configured correctly and that the signal generator is connected and has power.");
                        Console.ReadLine();
                    }
                }
                catch (TimeoutException)
                {
                    Console.WriteLine("Error! Please verify that the COM port is configured correctly and that the signal generator is connected and has power.");
                    Console.ReadLine();
                }
                foreach (String s in results)
                {
                    Console.WriteLine(s);
                }
                Console.ReadLine();
                port.ClosePort();
            }
            catch (System.IO.IOException)
            {
                Console.WriteLine("Error! Please verify that the COM port is configured correctly and that the signal generator is connected and has power.");
                Console.ReadLine();
            }
        }
    }
}

```

SerialClass.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO.Ports;
using System.Collections;

namespace NT_Appnote
{
    class SerialClass
    {
        public SerialPort _serialPort = new SerialPort();
        public SerialClass()
        {
            _serialPort.Parity = Parity.None;
            _serialPort.StopBits = StopBits.One;
            _serialPort.PortName = "COM4";
            _serialPort.ReadTimeout = 5;
            _serialPort.BaudRate = 19200;
        }

        public int BytesToRead()
        {
            return _serialPort.BytesToRead;
        }

        public void OpenPort()
        {
            _serialPort.Open();
        }

        public void ClosePort()
        {
            _serialPort.Close();
        }

        public void SendCommand(String command)
        {
            _serialPort.Write(command + "\r");
        }

        public List<String> ReadDataToList()
        {
            List<String> data = new List<string>();
            String temp;
            while (_serialPort.BytesToRead !=0)
            {
                temp = _serialPort.ReadLine();
                data.Add(temp);
            }
            return data;
        }
    }
}
```

