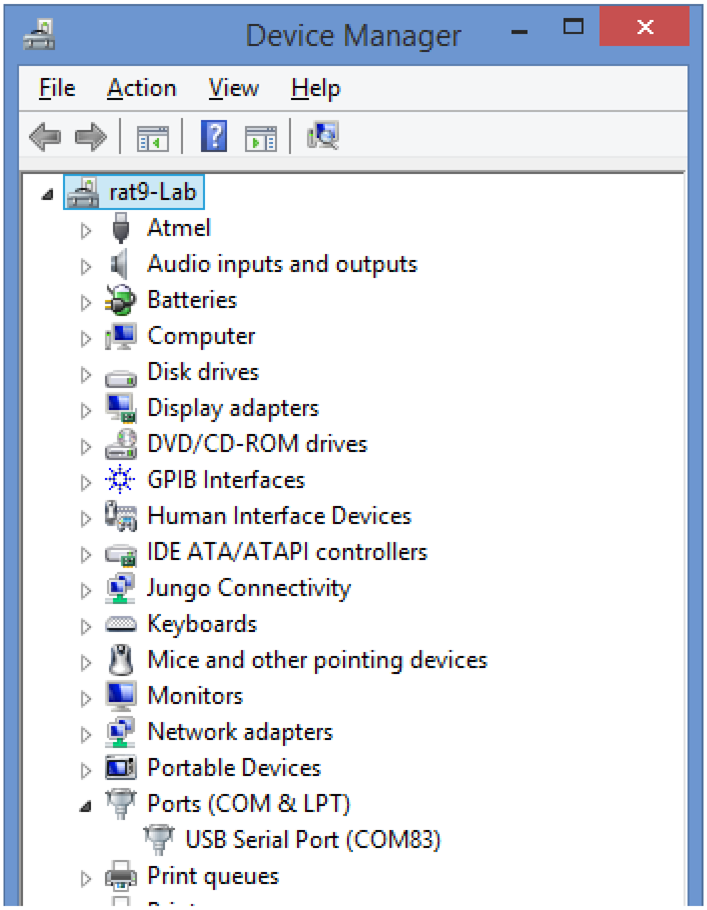
DS INSTRUMENTS
WWW.DSINSTRUMENTS.COM

Serial Remote Operation Programming Guide

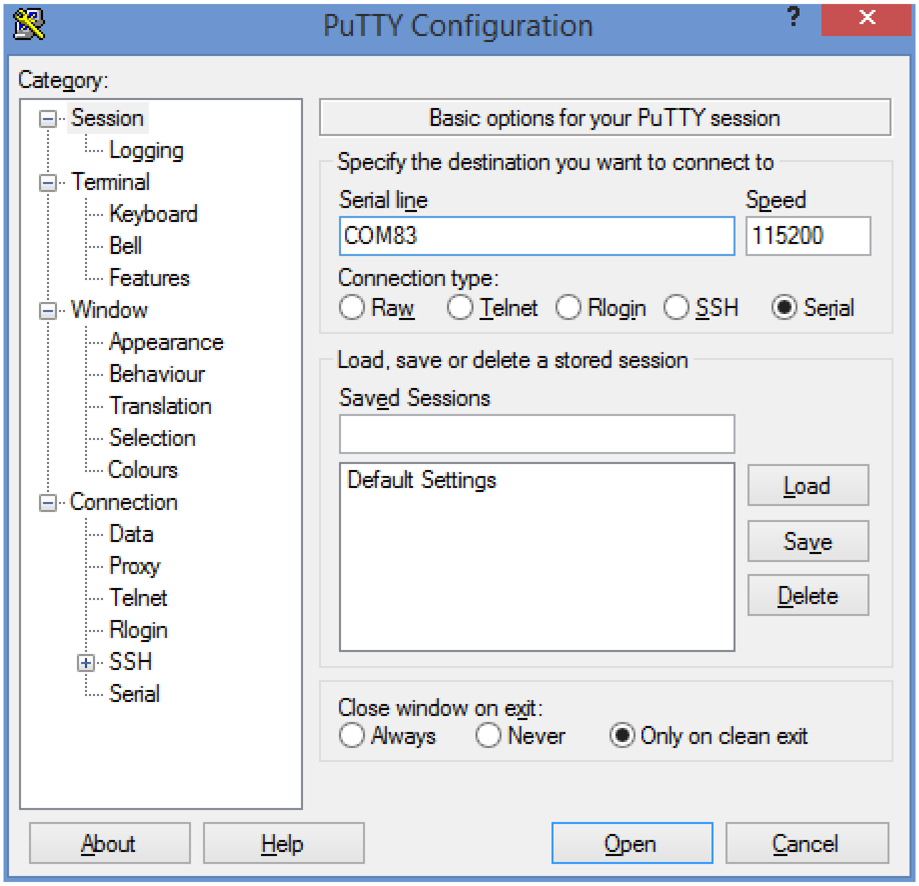
Automated / Script Control:

All of our products can be controlled from any serial-capable programming language or environment. MATLAB, .NET, Linux, python are all popular. We use Visual Studio 2015 and C# for our standard GUI. First determine the port number that your device has installed itself as:



- rat9-Lab
  - Atmel
  - Audio inputs and outputs
  - Batteries
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - GPIB Interfaces
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Jungo Connectivity
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Portable Devices
  - Ports (COM & LPT)
    - USB Serial Port (COM83)
  - Print queues

We recommend using the free serial terminal program "puTTY" to connect and test out commands, the serial port settings are: **115200bps, no parity, 8 data bits, 1 stop bits.** Command terminator is LINEFEED ("\n").



Specify the destination you want to connect to

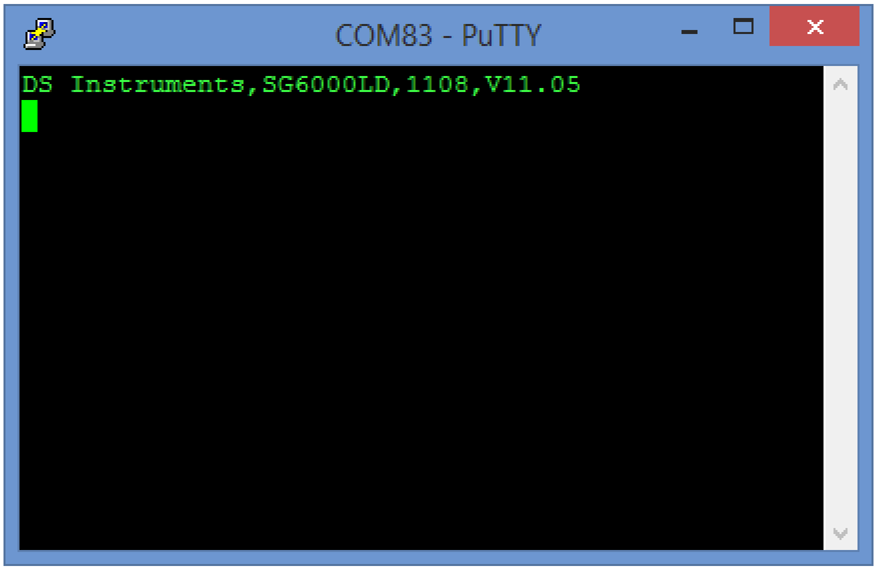
Serial line: COM83

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

Once connected you can send commends. Note that commands you send to the unit are not displayed in puTTY, only the response. Shown here is the response to the command "***IDN?**":



```
DS Instruments,SG6000LD,1108,V11.05
```

DSI 2017 V1.1 USA

DS Instruments Combined Command List:

Command	Example 1	Example 2	Description
FREQ:CW	FREQ:CW 2GHZ	FREQ:CW 123.5MHZ	Set signal generator output frequency
FREQ:CW?			Return output frequency
OUTP:STAT	OUTP:STAT ON	OUTP:STAT OFF	Turn RF output on or off
OUTP:STAT?			Return RF output state setting
ATT	ATT 30	ATT 13.5	Set value for DAT units
ATT?			Request current attenuation setting
PHASE	PHASE 90	PHASE -30	Set phase shift in degrees
PHASE?			Return current phase setting
*IDN?			Return the SCPI identification string
*PING?			returns "PONG!" if device is responding
SYST:ERR?			Returns any pending error codes
*RST			Reset unit now
*DISPLAY	*DISPLAY OFF	*DISPLAY ON	Power on or off the display
*BUZZER	*BUZZER ON	*BUZZER OFF	Mute the buzzer
*SAVESTATE			Save frequency & attenuation as boot defaults
*UNITNAME	*UNITNAME Bob	*UNITNAME DEV-34	Set a unique name in flash memory
*UNITNAME?			Return this device's name

Example Code (C# .NET Framework):

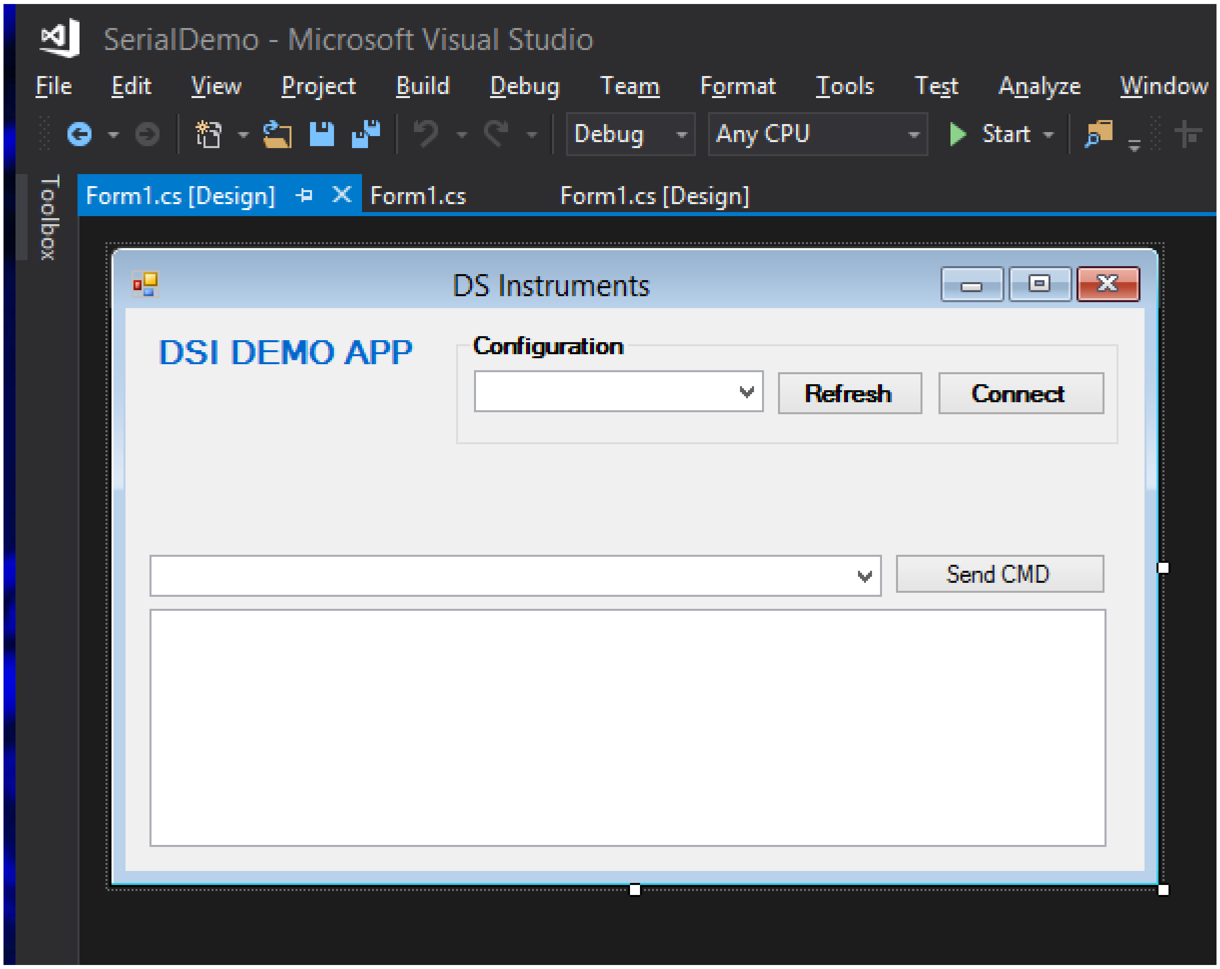
```
using System;
using System.IO.Ports;      // include serial port library

SerialPort myPort = new SerialPort(“COM5”, 115200, System.IO.Ports.Parity.None, 8, System.IO.Ports.StopBits.One);

myPort.Open();   // open the port we just made

myPort.WriteLine("*IDN?");          // send any command here

myPort.ReadTimeout = 250;

string myResponse = myPort.ReadLine();      // read back the response

System.Threading.Thread.Sleep(30);        // delay before sending the next command
```

To execute commands faster the OLED display and the buzzer can be disabled. We can also request a list of the installed COM ports from the system:

```
string[] ports = SerialPort.GetPortNames();      // string array of installed COM ports
```

A full working GUI example made in Visual Studio 2017 is ready for download here:

https://www.dsinstruments.com/support/visual-studio-csharp-programming-example-code/



- More information on serial ports with .NET can be found here: https://msdn.microsoft.com/en-us/library/system.io.ports.serialport(v=vs.110).aspx
- Serial programming for Linux information can be found here: https://en.wikibooks.org/wiki/Serial_Programming/Serial_Linux

Example Code (PYTHON):

After python and the pyserial module has been installed on your system, this example code will send connect, send, and receive commands from our products:

```
# for PYTHON 3+ with pySerial module installed
# DS INSTRUMENTS 2017 PYTHON SCPI REMOTE CONTROL EXAMPLE
import serial  # use the serial module (https://pypi.python.org/pypi/pyserial)
import time    # delay functions
badCommandResponse = b'[BADCOMMAND]\r\n'   # response if a command failed (b makes it into bytes)

ser = serial.Serial("COM79", 115200, timeout=1) #Change the COM PORT NUMBER to match your device
if ser.isOpen():   # make sure port is open
   print(ser.name + ' open...')   # tell the user we are starting

   ser.write(b'*IDN?\n')  # send the standard SCPI identify command
   myResponse = ser.readline()   # read the response
   print(b'Device Info: ' + myResponse) # print the unit information

   time.sleep(0.1)   # delay 100ms
   ser.write(b'PHASE?\n')      # try asking for phase
   myResponse = ser.readline() # gather the response
   if myResponse != badCommandResponse:   #is this is not a phase shifter why print the error
      print(b'Phase=' +myResponse)

   time.sleep(0.1)  # delay 100ms
   ser.write(b'FREQ:CW?\n')  # try asking for signal generator setting
   myResponse = ser.readline()  # gather the response
   if myResponse != badCommandResponse:  # is this is not a signal generator why print the error
      print(b'Frequency=' + myResponse)

   time.sleep(0.1)  # delay 100ms
   ser.write(b'ATT?\n')  # try asking for step attenuator setting
   myResponse = ser.readline()  # gather the response
   if myResponse != badCommandResponse:  # is this is not an attenuator why print the error
      print(b'Attenuation=' + myResponse)

   time.sleep(0.1)  # delay 100ms
   ser.write(b'FREQ:CW 3GHZ\n')  #lets change a setting now!
```

Notes

- Complete SCPI command lists are located on our website (https://www.dsinstruments.com/documents/)
- Full datasheets are available on the product pages (https://www.dsinstruments.com/store/products/)
- Tech support email can be reached at **support@dsinstruments.com**

Visit us for control software and tech support: https://www.dsinstruments.com/documents/

Thanks for your business!