

Application Note 1008

Using Multiple SXI-D Instruments With a Single Application Program



Sid Levingston
Engineering Manager
10/22/2007

Introduction

Spectrum Detector offers two application software programs for use with our SXI-D series of digital instruments. One program is for single instrument usage, and the other offers dual instrument usage for dual channel applications. While these two software packages are available free of charge to our customers, the capability exists to use more than two instruments at a time. The programmer can use the provided Lab View drivers to construct a custom software package that uses multiple instruments, with the limit being the number of USB ports available along with the instrument data rate and the speed of the host computer. Alternatively, the user can write custom software using the development environment of their choice, as long as it supports the use of com ports. This application note discusses the outline for writing a multiple instrument program in Lab View, but the ideas are applicable to any high level language.

The Find Devices on Ports VI

When the SXI-D is plugged into a USB port, the drivers will cause Windows to associate a Virtual Com Ports (VCP) with the instrument. The key to using multiple instruments is locating the instruments and finding their assigned VCP. Fortunately, this is relatively easy to do. Once the instruments have been plugged into the USB ports, the general program flow is:

1. Obtain a list of the available com ports currently open. Lab View contains a VI specifically for this purpose. Other languages may need to access the Windows registry to find the currently used ports.
2. Initialize each port found in step 1 at its maximum baud rate (usually 921600), 8 data bits, no parity, and 1 stop bit.
3. Send a 250ms com break to each of the VCP's found in step 1. A com break is an empty bit frame followed by a stop bit. This will cause the SXI-D's to reset and enter an internal automatic find baud rate routine. This is done so that PC's with slower baud rates can communicate their baud rate to the SXI-D. On reset, the SXI-D will enter a loop and wait for a carriage return character to be received. Once the character is received, the SXI-D compares it to the expected results for baud rates from 921600 to 115200. It then sets its internal baud rate to the baud rate that matches that of the host PC, and returns that baud rate to the host PC.
4. Send a carriage return to the first port in the list, and read the reply. If an SXI-D is on the port, it will return the discovered baud rate. If this occurs, send the IDN query to the port. If an SXI-D is on the port, it will respond with "SXI-D USB". You can then save the port and assign a unique name to the instrument by using the USN<Unique name goes here> command.
5. Repeat step 4 for all the VCP's in the list.

Note that any time the SXI-D is powered up or reset; it will enter the internal auto baud routine and wait for a carriage return to be sent before returning to its measuring state.

Since a com break will always reset the instrument it provides a handy way to programmatically make sure the instrument is in a known state.

There is a VI in the Lab View driver library available called “Find Dual Devices on Ports”. While the VI is named as if it finds two devices, it will find all SXI-D devices and return the number of devices found, as well as two clusters. One cluster contains arrays of the discovered VCP’s and the associated SXI-D ID strings. The second cluster contains arrays of the discovered VCP’s and the associated SXI-D unique user names if any have been previously assigned. Figure 1.0 shows the VI front panel after having found 4 unique instruments.

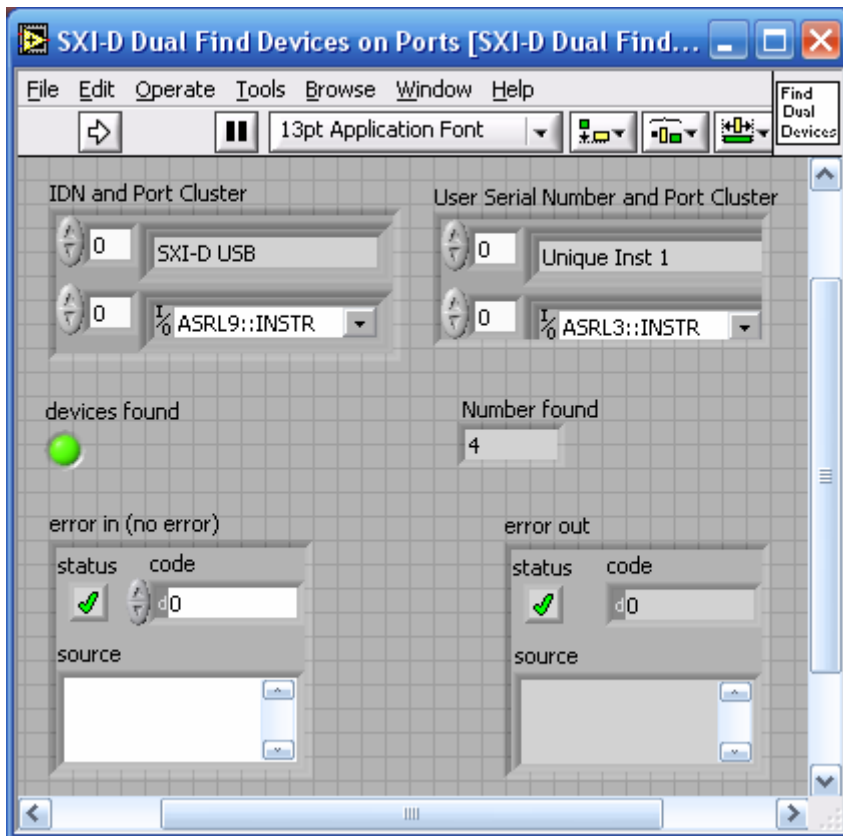


Figure 1.0

Note that the User Serial Number and Port Cluster now contain the unique names assigned by the user and the associated VCP. Now each of the SXI-D’s can be commanded or queried as needed to implement the desired data acquisition program.

SXI-D Command Set

The command set is used to set the SXI-D state as needed for data acquisition. It is recommended that the user establish communications with the device and try the commands to ensure each is understood before writing the data acquisition program. Hyper terminal

is very useful for these experiments. All host commands and queries must consist of 3 alphabetic characters followed by an optional argument list. If there are multiple arguments, they must be comma separated. All commands must be terminated by a carriage return followed by a line feed for the SXI-D to correctly interpret them. The format is:

```
abc<arg1,arg2,...,argN>CR LF
```

The 3 characters are case independent. If the command has an argument and the argument is valid, "OK\r\n" is returned unless noted below. If the command or the argument is invalid, "ERR\r\n" is returned. If the command has no arguments, it is interpreted as a query if applicable and the value of the current state of the parameter referenced by the command is returned. All returned parameters are terminated with a carriage return followed by a line feed.

The following commands are not a complete listing. The Lab View Driver library contains VI's to implement all the supported commands.

The programmer must use caution when using the VI library. There are commands used for factory diagnostics and calibration that should not be used by the user. The following is a list of commands recommended for use in developing a custom Lab View application.

Command and Query List

VER. Query only. Returns the factory firmware version.

VER\r\n Returns "n.nn\r\n" where n.nn is the firmware version.

STR. Command only. Enables the trigger engine for triggered devices, and starts/stops the data flow for non-triggered devices. The argument is a Boolean flag.

STR1\r\n enables the trigger engine or data flow. Nothing is returned.
STR0\r\n disables the trigger engine or data flow. "OK\r\n" is returned.

TRG. Command and Query. Sets the trigger level for triggered devices. This command is not used on non triggered devices. The argument is an integer from 2% to 20%.

TRG<arg>\r\n Sets the trigger level in % of full scale. Nothing is returned
TRG\r\n returns the trigger level in % of full scale, "dd\r\n".

RNG Command and Query. Sets the measurement range. The argument is the range index value. Since ranges are not fixed for some devices, if the range is not allowed the command will return an error.

RNG<arg>\r\n Sets the range index. Returns "OK\r\n" if successful, "ERR\r\n"

otherwise.

RNG\r\n returns the current range index, "nnE±N\r\n".

MAX Command only. Reads the Sensor Maximum Range index.

MAX\r\n Reads the Sensor Maximum Range.

MIN Command only. Reads the Sensor Minimum Range index.

MIN\r\n Reads the Sensor Minimum Range.

Note: The range indices are:

0 = 2pJ/pW, 1 = 20pJ/pW, 2 = 200pJ/pW
 3 = 2nJ/nW, 4 = 20nJ/nW, 5 = 200nJ/nW
 6 = 2μJ/μW, 7 = 20μJ/μW, 8 = 200μJ/μW
 9 = 2mJ/mW, 10 = 20mJ/mW, 11 = 200mJ/mW
 12 = 2J/W, 13 = 20J/W, 14 = 200J/W
 15 = 2KJ/KW.

ZRO Command only. Sets the baseline of the internal signal at the ADC when using an Optical Power Meter or a Joule meter. Uses no argument. The baseline voltage is 400mV, or 655 ADC counts. Command may be sent multiple times to achieve the correct baseline.

ZRO\r\n Sets the offset DAC so that the ADC baseline is correct. Returns the value the offset DAC was set to and the ADC counts, "dddd,dddd\r\n".

USN. Command and Query. Sets>Returns the user serial number. The argument is a string.

USN<arg>\r\n sets the serial number to <arg>

USN\r\n returns the factor serial number, "abcd\r\n".

UCD. Command and Query. Sets>Returns the user calibration date. The argument is a data in mm/dd/yyyy format.

UCD<arg>\r\n sets the calibration date to <arg>

UCD\r\n returns the calibration date, "mm/dd/yyyy\r\n".

SQL Command only. Sets the internal attenuator to full attenuation or to the current attenuation value. Used to simulated blocking the external beam so dual channel applications can synchronize on the same pulse. Argument is a Boolean flag.

SQL1\r\n returns nothing, full attenuation, signal is blocked from trigger engine.

SQL0\r\n returns nothing, current value attenuation, signal is applied to trigger engine.

ADC Query only. Takes 256 sample of the signal and returns the minimum, mean, and maximum values. This command can be used in conjunction with the DAC command to implement a custom zero algorithm.

DAC Command/Query. Sets the 12 bit voltage offset DAC. The argument is 0 to 4095. 0 counts equals -2.5V offset, 4095 counts equals +2.5V offset. This command can be used in conjunction with the ADC command to implement a custom baseline setting algorithm. Note that the ADC is ground based with a 0 to +2.5V input range. The DAC range includes negative voltages to be able to offset a voltage that is positive.

HLP Query only. Returns the list of commands/queries. Note that all commands/queries will be returned, including the factory commands for calibration and diagnostics. Use of any commands other than the ones listed above can invalidate the calibration.

Taking Data from Multiple Devices

Measurement Data is sent from the SXI-D as an Hexadecimal string.

For a Radiometer or Joule meter, pulse amplitude and period information are sent.

The format is: DDDD,DDDDDDDDCRLF

Where DDDD is a 12 bit value representing the pulse amplitude and DDDDDDDD is a 32 bit value representing the pulse period counter. The values are comma separated for easy parsing, and terminated by a carriage return followed by a line feed.

A full scale reading in any range is 3276 counts. To find the value of the pulse, divide it by 3276 and multiply the result by the range.

One million period counts represent 1 second. To obtain the pulse frequency, divide one million by the pulse period counts.

For an Optical Power meter, the counts from the internal ADC are sent every 100ms.

The format is: DDDDCRLF

Where DDDD is a 12 bit value representing the ADC counts and terminated by a carriage return followed by a line feed. The MSB is the sign bit. If it is set, the value is negative. Internally, the SXI-D ADC treats 400mV as “ground” so it can measure below ground if needed.

A full scale reading in any range is 3276 counts. To find the value of the measured power, divide the pulse count by 3276 and multiply the result by the range.

To stream data from multiple instruments, send the STR1 command to each instrument to turn on its data stream. The read each instrument’s VCP in sequence. The data will be

sent by each device in real time and stored in the VCP buffer. This requires that the data be read in a timely manner or the port buffers will eventually overrun. This will not be an issue if a set of fixed data is to be gathered and real time viewing is not important. Size the VCP buffers to hold the expected data and then read it out as needed. Should data acquisition in real time be required, the time required for the SXI-D to send its data after a measurement event is $180\mu\text{s}$ for a baud rate of 921600. (11 bits per character, 15 character per measurement in Radiometer or Joule meter mode. $11 \times 15 / 921600 = 179\mu\text{s}$) As an example, suppose 4 SXI-D's are being used. Each instrument is sending data at 10 Hz rate. If the measured events are synchronous, then the host PC must read and process the data in less than 9.98ms in order to be ready to read the VCP's for the next data. Most PCs will easily be able to do this, but as data rates increase, the programmer must be aware of the timing constraints associate with real time data acquisition. When no data is desired from a given SXI-D, send the STR0 command to prevent VCP buffer overflow.

The SXI-D dual channel software sits in a measurement loop when any of the data streams are active. It reads each instrument in sequence, and then processes the data as needed depending on the state of the front panel controls. It then returns to the wait for next data state. The largest amount of processing time is spent plotting the data graphs as Lab View requires that all data to be graphed be saved. This can create large arrays of double precision numbers that must be reprocessed each pass of the loop. The programmer can time these events and determine where processing time can be saved if need be.

Conclusion

The SXI-D can be used as a custom solution in multiple data source measurement systems. Its versatility allows the user to write application software using familiar com port communication protocols, and choose the high level programming language that is best suited to the project. While the Spectrum Detector applications software is written in Lab View, the program flow is similar for any language that supports com port communications. As there are many possible application programs, it is not possible to discuss in detail how to implement them. The basic program will always:

1. Establish Communications with the SXI-D's on the VCP's.
2. Set the instruments to the desire range, trigger level, etc.
3. Turn on the data streams and gather data.
4. Process and store or display that data as required.
5. Turn off the data streams when data gathering is not active.

Each of these basic steps can be accomplished using the information in this application note.