Cscope Driver vi Description – Version 2.2

Summary

The Cscope Driver vi is used by Labview programs to communicate with the Cleverscope CS328A acquisition unit. The driver has storage allocated for up to 32 simultaneous connections to 32 connected Cleverscope Acquisition Units (CAU's).   This document is illustrated using the Labview environment, but the principles also apply to use of the DLL. Version 2.2 adds the Cscope Function.vi to explicitly call functions. Functions will be used for non-acquisition related capability in the future.

Control driver vi's.

There are two vi's that act as the interface to the Cleverscope Acquisition Unit (CAU).
These are:

- **Cscope control driver.vi**
  This vi is used by the user code to interact with the CAU control system.
- **Cscope Function.vi**
  This vi is used to exercise system level functions, and gain access to calibration and the link port.
- **Cscope control driver runtime.vi**
  This vi is launched by Cscope control driver.vi when a new acquisition unit slot is initialized. The vi is **dynamically loaded**. If you are building your own executable, you will need to include Cscope control driver runtime.vi in your dynamically loaded vi list. If you upgrade the driver to a new Labview platform, remember to load Cscope control driver runtime.vi, and then do a 'Save All' to ensure that all of the dynamically loaded portion is correctly updated to the new Labview version. One copy of the runtime is loaded for each CAU being used. If you use large sample size transfers, be aware that significant memory will be allocated and retained for each CAU that uses the large sample size transfer. The runtime continues to operate in the background even though the control driver maybe stopped. The runtime is only removed when the 'Finish' command is called.

Cscope Control Driver Parameters

Important parameters are:

Acquisition Unit In

Input: Signed 32 bit number.
This value sets the Cleverscope Acquisition Unit (CAU) being addressed (0..31). Typically up to 32 simultaneous CAU's can be controlled at the same time.   Each connected acquisition unit occupies one slot. If a slot is not occupied it uses no additional memory – the CAU runtime support is dynamically loaded when the CAU is opened.

Acquisition Unit Out

Output: Signed 32 bit number.
Flow through from Acquisition Unit Out, allows sequential flow.

Command

Input: Unsigned 16 bit value.
Values are:

- 0 - **Initialize**. Call this once to initialise the acquisition system. Further calls are ignored.
- 1 – **Acquire**. Call to acquire data as defined by the Acquire Definition and other parameters.. Caalling acquire automatically updates the acquisition unit with any changed acquire values.
- 2 – **Replay**. Call this to re-decimate the capture buffer, and return new samples, based on the SamplesIn Replay, ReplayStartTime and ReplayStopTime values.
- 3 – **Wait for samples**. Call this to check if a trigger has occurred, and the samples are available. The Value GotSamples is set true when all the samples have been received. The call will wait up to 40ms for a trigger. After 40ms, the call times-out, returning false. The wait blocks the thread, but relinquishes control to the operating system during the wait. This maximizes throughput.
- 4 – **Update.** This call updates acquisition unit values if the acquisition unit is not acquiring, or is waiting for a trigger. Can be used to update the signal generator values for example.
- 5 – **Close**. Call this to close down and remove the runtime for the currently selected acquisition unit. After calling close the CAU status will be returned as 'Runtime Closed'.
- 6 - **Status**. Calling the Control Driver with the status command just returns the current connection status. Status values are 'Runtime Closed', 'Closed', 'Open' and 'Fault' (0..3). See below.
- 7 - **Function**. This value is used to retrieve specific values about the connected Cleverscope Acquisition Unit.
- 8 - **Get Frames**. Gets a multi-frame sequence as one array. The value num_samples is the number of samples in one frame. The value num_frames are the number of frames included in the array. After sending the command, call 'Wait for Samples' until the samples are transferred.
- 9 - **Finish**. Call this to close down any remaining unclosed runtime components, and to remove the control driver from memory.

ReplayStartTime

Input: Double.

This value specifies, in seconds, the start time of the samples to be returned in the decimated replay from the sample buffer. If the start time is outside the actual available buffer start and stop times (relative to the trigger), the start time will be clipped to either the beginning or end of the buffer, as necessary.

ReplayStopTime

Input: Double.

This value specifies, in seconds, the stop time (inclusive) of the samples to be returned in the decimated replay from the sample buffer. If the start time is outside the actual available buffer start and stop times (relative to the trigger), the start time will be clipped to either the beginning or end of the buffer, as necessary.

SamplesInReplay

Input: Signed 32 bit number.

This value specifies the number of samples that will be returned in the decimated replay from the sample buffer. Values may vary from 0 to the size of a frame. If you request more samples than in a frame, the number will be set to the frame size. The maximum size is the acquisition storage size (4 or 8M) divided by 2.

Max Units

Input: Signed 32 bit number.

This value specifies the maximum number of simultaneous connections to Cleverscope Acquisition Units that can be maintained at one time. If unwired, the default is 32.

Frame Number

Input: Signed 32 bit number.

a. This value specifies which frame to return from a multi frame replay. The default is 0.
b. For a Function, this is the function number to be executed (only provided for previous version support. For new programs use the Cscope Funnction.vi instead)

GotSamples

Output: Boolean
Returns false if samples are not yet all received. True = received the values.

T0dT

Output –Cluster of [T0 – double, dT – double, n –I32]
This cluster returns the waveform details:
- T0 - Returns the start time of the waveform being replayed relative to the trigger, which is time 0, in seconds.
- dT - Returns the interval between successive samples, in seconds.
- n - Returns the number of samples in the sample array.

NumFrames

Output – I32.
a. Returns the number of frames that the sample array is segmented into – only used when returning all the frames in a sequential capture in one transfer. As an example, assuming 2000 samples per frame, and 100 frames sequentially captured, one data array of 200,000 samples will be returned, being composed of 100 segments of 2000 samples.
b. For a Function this is the result of the function call (only provided for previous version support. For new programs use the Cscope Funnction.vi instead)

ChanA

Output - array of single real
Returns the Channel A waveform as single precision reals.

ChanB

Output - array of single real
Returns the Channel B waveform as single precision reals.

Digital Inputs

Output - array of U16
Returns the digital inputs waveform as U16

CAU Status

Output - Unsigned 16 bit integer

This value is returned after every call to the control driver.
Status values are 'Runtime Closed', 'Closed', 'Open' and 'Fault' (0..3). See below.
1. At first power up, with a CAU Unit not initialized, The CAU status will be 'Runtime Closed'.
2. Following a call to Initialize the Control Driver, the CAU status will change to 'Closed.
3. When the connection is made (via USB or Ethernet), the CAU status changes to 'Open'.
4. While closed you may continue to call initialize, with a new interface specification, if needed.
5. If a Fault occurs during a transfer (for example loss of power to the CAU), the Fault status will be returned once, and then the runtime will be automatically be closed, and the status returns 'Runtime Closed'. You will need to Initialize the CAU unit again.

Error In, Error Out

Input – Error Cluster, Output – Error Cluster
Defines any input errors. If there is an error, the unit will not acquire (but will close and finish). Any errors are Output.

AcquireDefinition

	Item	Description	Data Type
Acquire Mode: stop	Acquire Mode	How to acquire: 0 = Single, 1= automatic, 2 = triggered, 3 = stop	U16
Acquisition mode: Sampled	Acquisition Mode	Method of acquisistion: 0 = sampled, 1= Peak captured, 2 = Filtered, 3= Repetitive, 4= Waveform avg, (for which make sure there are at least waveform avg +1 buffers).	U16
Acquirer: Cleverscope	Acquirer	Sets the acquirer to use. Always use 4 = cleverscope	U16
Transfer Chans: Chan A+B	Transfer Chans	Always set to 2 = transfer all channels.	U16
A max scale: 5.00	A max scale	Maximum A channel scale value.	Double
A min scale: -5.00	A Min scale	Minimum A channel scale value – make lower than max	Double
B max scale: 5.00	B max scale	Maximum B channel scale value.	Double
B min scale: -5.00	B min scale	Minimum B channel scale value – make lower than max	Double
A Probe: x1	A probe	A Probe Multiplier 0 = x1, 1 = x 10, 2 = x100, 3 = x1000	U16
B Probe: x1	B probe	A Probe Multiplier 0 = x1, 1 = x 10, 2 = x100, 3 = x1000	U16
A Coupling: DC	A Coupling	A Coupling, 0 = AC, 1= DC	U16
B Coupling: DC	B Coupling	B Coupling, 0 = AC, 1= DC	U16
A Bandwidth: 100 MHz	A Bandwidth	Bit 0 - Global Filter enable, 0 = no filter, 1 = use filter Bit 2:1 - Pre-filter frequency 0 = No filter, 1 = 20 MHz filter Bit 3: - If true, use the moving average (MA) filter Bit 4: - Reserved Bits 7:5 - Filter time constant, in taps: 000 = no filter, 001 = 4 tap, 010 = 8 tap, 011 = 16 tap 100 = 32 tap, 101 = 64 tap, 110 = 128 tap111 = reserved MA For the moving average only the channel A moving average value is used, and it also used for Channel B	U16
B Bandwidth: 100 MHz	B Bandwidth	As for A, but uses Chan A bits 7:5 for filter setting	U16
Trigger Source: Chan A	Trigger Source	Sets trigger source. 0 = A chan, 1 = B chan, 2 = Ext Trigger, 3 = Dig Input, 4 = Rear Input	U16
Trigger Amplitude: 0.0	Trigger Amplitude	Level at which to trigger	Double
A Trigger Amplitude: 0.0	A Trigger Amplitude	Not used in driver.	Double
B Trigger Amplitude: 0.0	B Trigger Amplitude	Not used in driver.	Double
Trigger Filter: None	Trigger Filter	Sets filter on trigger. 0 = None, 1 = Low Pass, 2 = Hi Pass, 3 = noise (2 divisions of hysteresis)	U16
Trig Slope	Trig Slope	Sets the trigger slope. 0 = rising, 1 = falling	U8
Trigger Holdoff: 0.0	Trigger Holdoff	Not used in driver.	Double
Dig Pattern Rqd	Dig Pattern Rqd	Sets if the digital pattern qualifies the analog trigger. 0 = not required. 1= required.	U8
Dig Pattern: 0	Dig Pattern	Sets the digital pattern for digital input triggering. Byte 0 = Select mask, 1= input is used. Byte 1 = Pattern required before trigger Byte 2 = Pattern required to trigger Byte 3 not used. Bit 0 is input 1 .. Bit 7 is input 8	U32
Ext Trig Threshold: 0.00	Ext Trig Threshold	Sets the amplitude of the external trigger input, -6..+18V	Double
Dig Input Threshold: 0.00	Dig Inp Threshold	Sets the amplitude of the digital input threshold, 0 .. 10V	Double
Start Time: -3.00m	Start Time	Sets the start time relative to the trigger, at which acquisition will begin. If positive delayed triggering is used.	Double
Stop Time: 3.00m	Stop Time	Sets the stop time relative to the trigger. Range is –22 .. + 22 seconds. Resolution is 10 ns.	Double
Pre Trig Time: 3.000m	Pre Trig Time	Not used in driver.	Double
Port: Port 1	Port	Not used in driver.	U16
Num divisions: 6	Num divisions	Not used in driver.	I16
Num seq frames: 1	Num seq frames	Sets the number of frames captured sequentially. If not waveform avg method of capture set to 1. If waveform avg capture, set to the number of averages used, 4,16,64,128. If capturing sequential frames, set to number of frames to capture.	I16
Num Buffers: 2	Num Buffers	Sets the number of buffers allocated for frame capture. Must be at least num waveform averages + 1.	I32

Control	Value
Sig Gen Freq	1000.00
Sig Gen Amp	4.00
Sig Gen Offset	0.00
Sig Gen Waveform	sine
Sig Gen Sweep	Log
Sig Gen Func	Standard
Sig Gen Freq 2	1000.00
Sig Gen Phase	180.00
Trig 2 Function	None
Min Trigger Period	10n
Max trigger Period	100u
Trigger count	1
Trig 2 Slope	⎍
Trig 2 Source Chan	Chan A
Trig 2 Level	0
Dig Pattern 2 Rqd	●
Dig Pattern 2	0
Trigger 2 Source	Trigger 1 inverted
Waveform Averages	4
Value changed	1280051
Freq Span	0
Freq Res	0
Duration	0
Resolution	0
Link Port	Uart
Ext Sample Clock	●
F Spare 2	●
F Spare 3	●
F Spare 4	●
Sampler Resolution	10 bit
Intf Source	USB
Update Rate	20 Frame/sec
Transfer Size	Normal
Sig Gen Freq Step	0.00
TCP Adr	xC0A80168
TCP port	53270

Name	Description	Type
Sig Gen Freq	Set the signal generator frequency in Hz. Range is 0.003..10e6 Hz.	Double
Sig Gen Amp	Amplitude of signal generator output. Range is 0..8V	Double
Sig Gen Offset	Offset of signal generator output. Range is –5..+5V	Double
Sig Gen Waveform	Sets the signal generator waveform. 0 = sine, 1= triangle, 2 = square, 3 = DC, 4 = 0V.	U16
Sig Gen Sweep	Not used in driver	U16
Sig Gen Func	0 means normal sig gen use, 1 means step the sig gen upwards by Sig Gen Freq Step automatically following a trigger.	U16
Sig Gen Freq 2	Not used in driver.	Double
Sig Gen Phase	Not used in driver.	Double
Trig 2 Function	Sets the use of Trigger 2. 0 = Not used, 1 = T1~2 < min, 2 = min<= T1~2 <= max, 3 = Count T1, 4 = Wait for T1, then count T2. T1~2 = time duration from trigger 1 to trigger 2.	U16
Min Trigger Period	Sets the min period. 0..22 secs, resolution is 10 ns.	Double
Max Trigger Period	Sets the max period. 0..22 secs, resolution is 10 ns.	Double
Trigger Count	Sets the number of counts for counting. 0..4,294,967,295	U32
Trig 2 slope	Sets the slope for trigger 2. 0 = rising, 1 = falling	U8
Trig 2 Source han	Sets the trigger 2 source channel. 0 = A chan, 1 = B chan, 2 = Ext Trigger, 3 = Dig Input, 4 = Rear Input	U16
Trig 2 Level	Sets the trigger 2 threshold level.	Double
Dig Pattern 2 Rqd	Sets if Trigger 2 is qualified by the pattern.	U8
Dig Pattern 2	Defines the trigger 2 digital pattern.	U32
Trigger 2 Source	Defines the trigger 2 source – 0 = Trigger 1 inverted, 1= Use the Trigger 2 definition	U16
Waveform Averages	Sets how many waveforms to average in acquisition unit if acquisition mode = waveform avg. Values are 4, 16, 64 and 128	I32
Value Changed	Change this value to cause the driver to check for changes in all the values in this data structure. If not changed, data structure values will not update.	I32
Freq Span	Not used in driver	Double
Freq Res	Not used in driver	Double
Duration	Not used in driver	Double
Resolution	Not used in driver	Double
Link Port	Defines how the Link port will be used. 0 means debug Uart, 1 means the Link Port outputs trigger, 2 is the port is disabled, 3 means this is a trigger slave cleverscope link port, 4 means this is a trigger master cleverscope link port, 5 means Uart Port, 6 means SPI Port, 7 means I2C port, 8 means used as 3 bit digital port, 9 means use Sig Gen to output arbitrary.	U8
Ext Sample Clock	0 means use internal 100 MHz sample clock. 1 means use external sample clock. Clock must be a sine or square wave, with 45-55% duty cycle, amplitude 0.3V – 3V p-p, biased to 0V or CMOS logic levels. The external clock range currently supported is 10 – 49 MHz.	U8
Fspare 2	Reserved for future use	U8
Fspare 3	Reserved for future use	U8
Fspare 4	Reserved for future use	U8
Sampler Resolution	Sets the sampler resolution to be used, 0 = 10 bits, 1 = 12 bits, 2 = 14 bits. Will clip to maximum resolution available.	U16
IntfSource	Source for connections – 0 = USB, 1 = Ethernet – Open specified IP address, 2 = Find first connected cleverscope, 3 = find Cleverscope with given serial number.	U16
Update Rate	Not used in driver	U16
Transfer Size	Use 0 to transfer one frame. Use 6 to transfer all the frames in a sequential capture as one array. See num frames value in next section.	
Sig Gen Freq Step	Frequency increment used when acquisition unit automatically steps the signal generator frequency following a trigger, if Sig gen Func = 1.	Double
TCPAdr	TCP address of acquisition unit. Format is bb.bb.bb.bb	U32
TCPPort	TCP port used for acquisition unit.	U32

CAU Ser Num Hi 17223 CAU Ser Num Lo 8756401 Function Number 0 Function Parameter 0 Function Result 0 Link Start 1 Link Timebase 0 Link Timer 0 Link Setup 0 Spare 1 0 Spare 2 0 Spare 3 0 Spare 4 0	CAU Ser Num Hi / Lo	Cleverscope Acquisition Unit serial number split into two U32 values. The upper two ascii characters are stored in the lower 16 bits of CAU ser Num Hi. The lower 4 ascii characers are stored in CAU ser Num Lo.	U32 U32
	Function Number	Should not be used direct – used by Cscope Function.vi for Function number transfer	Double
	Function Parameter	Should not be used direct – used by Cscope Function.vi for Function parameter transfer	
	Function Result	Should not be used direct – used by Cscope Function.vi for Function result transfer	
	Link Start	Used to define how the message stored in the link buffer is transmitted: Bit 0 = Transmit when link_port data received Bit 1 = transmit at start of sampling Bit 2 = transmit on trigger Bit 3 = transmit on completion of sampling. Bit 4 = transmit on timer Bit 5 = transmit on digital pattern 1 Bit 6 = transmit on digital pattern 2 Bit 7 = transmit on external trigger rising Bit 8 = transmit on external trigger falling Bit 9 = transmit on link input rising Bit 10 = transmit on link input falling Bit 11 = transmit again as soon as previous transmit completes.	
	Link Timebase	Clock used to run uart, i2C, spi, or digital outputs. In 1/70Mhz units. = 14.29 ns. eg 9600 baud = 7292 (an error of 0.4 parts in 9600). 115,200 baud = 608, with an error of 70 parts in 115,200.	
	Link Timer	timer used for repeated messages, in 10us units. Bit 31 of the timer is enable. 0 = disable timer, 1 = enable, and start timer	

	Link Setup	Setup bits for the different ports
		 UArt dependant setup values.
		  Bit 0 = standard(0), inverted (1) output
		  Bits 2:1 - flow control
		    0 = no flow control,
		    1 = ASCII Xon/Xoff flow control
		    2 = Link Input low stops transmitting
		    3 = Digital Input 8 low stops transmitting
		
		 SPI dependant setup.
		   Bit 4 = CPHA = 0 means data is latched on the leading edge of CLK, and changes on the trailing edge = 1 means data is latched on the trailing edge of CLK, and changes on the leading edge
		   Bit 5 = CPOL = 0 means the clock is low while idle, = 1 means it is high while idle.
		
		 Digital Setup:
		  Bit 8 = Sampled/Timed
		    = 0, means digital output at sample rate
		    = 1. means digital output after count time_base values.
		
		 Sig Gen setup:
		  Bits 14:12 - Sig gen Destination
		    0 = DC Offset (uses DC offset as low frequency ARB)
		    1 = DC gain (amplitude modulation)
		    2 = Frequency Register (frequency modulation) 28 bit
		    3 = Phase Register (phase modulation).
		    4 = Select Freq 0 / Freq 1 (1 bit)
		    5 = Select Phase 0/ Phase 1 (1 bit)
		  Bit 15: = Sampled/Timed
		    = 0, means sig gen output at sample rate .
		    = 1 means sig gen output after count time_base values.	

Using *Cscope control driver.vi*

To use the *Cscope control driver.vi* carry out the following steps:

1. Set the Acquisition Unit number (0..31). Call the DLL with the **Inititialize (**0) command. You will need to **Inititialize** each Cleverscope slot used.
2. Use the **CAU status** (6) command until the status is 'Open'.
3. Setup the Acquire Definition, and call using the **Acquire** (1) command. The Acquire call automatically updates the acquisition unit to the contents of the acquire structure.
4. Use a timed loop that achieves the desired throughput. Maximum throughput is typically 20 updates per second (50msec intervals). If the sequence
   Wait for samples – Acquire Samples – wait 50ms
   is used, maximum throughput will be reached. Call the **Wait for samples** (3) command until GotSamples = 1. The data will now be in the data array. The driver uses a runtime that operates in the background.
5. If you want to replay another portion of the acquired data, use the **Replay** (2) command followed by **Wait for samples** (3) to check for the samples being transported. Any returned signal subset will be clipped to the start and end times specified when the acquire was made.
6. If you want to update the acquisition unit, without making an acquisition, or while waiting for a trigger, use the **Update** (4) command. You can control the signal generator this way.
7. Close this Acquisitions Unit by using the **Close** (5) command. Following the close command the CAU Status will return 'Runtime Closed'.
8. After having closed all other Acquisition Unit slots used, finish by calling the **Finish** (9) command. The **Finish** command automatically closes any slots you have not explicitly closed.

Notes:
1. The driver will automatically take the next lowest available USB port if more than one CS328 or CS328A are connected, and IntfSource = 0, or USB. The port/CAU binding is repeatable through PC restarts.
2. The CAU Status value may be used to check if the Cleverscope is available and turned on. If after an Init, the CAU status does not change to Open, the CAU is not available.
3. **ErrorOut** may be used to check for errors.

Using *Cscope Function.vi*

The *Cscope Function.vi* is used to recover Cleverscope Acquisition Unit (CAU) information, control the Link Port, and will be used for Calibration. The older method of using Cscope control driver.vi command 7 is still available. The following parameters are used by *Cscope Function.vi*:

Important parameters are:

Acquisition Unit In

Input: Signed 32 bit number.

This value sets the Cleverscope Acquisition Unit (CAU) being addressed (0..31).

Acquisition Unit Out

Output: Signed 32 bit number.

Flow through from Acquisition Unit Out, allows sequential flow.

Function Command

Input: Unsigned 16 bit value. Defines the command executed:

0 – Get Serial Num	Returns the serial number of the attached CAU
1 – Get Firmware Ver	Returns the version number of the firmware in the CAU
2 - Get Driver Ver	Returns the version number of the Cscope Control Driver (currently 2.2)
3 - Get Resolution	Returns the native bit resolution of the attached CAU
4 - Get Frame Length	Returns the space allocated to each frame in the CAU, in samples.
5 - Get Temperature	Returns the temperature inside the CAU in °C units.
6 – Start Link Send	Starts transmission of a message already in the Link Buffer programmatically
7 – Send Link Data	Sends the attached data to the Link Buffer

8 - Read Link data   Receives any data arrived since the last call into the Link Data Received.

Link Data Send

Input: An array of U8, which contains the data to be sent using Send Link Data

Link Data Received

Output: An array of U8, which contains the data received since the last call to Link Data Received.

Parameter

Input: A double real parameter to send to the Function.

Function Result

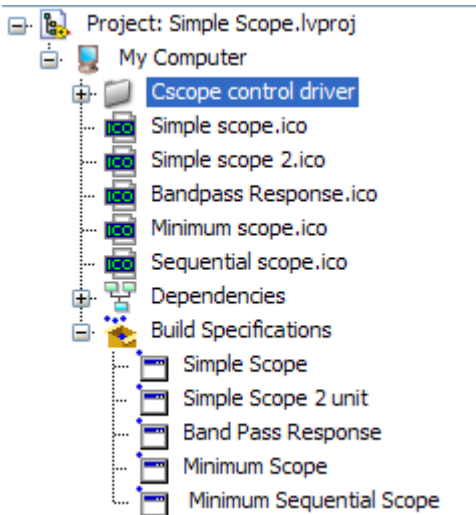Output - The double real result of calling the function.

Error In, Error Out

Input – Error Cluster, Output – Error Cluster
Defines any input errors. If there is an error, the unit will not acquire (but will close and finish). Any errors are Output.

Project arrangement

A project, Simple Scope.lvpoj is saved in the control driver for easy access, and to manage dependencies.
The project folder includes builds for the 5 examples.

- Project: Simple Scope.lvproj
  - My Computer
    - Cscope control driver
    - Simple scope.ico
    - Simple scope 2.ico
    - Bandpass Response.ico
    - Minimum scope.ico
    - Sequential scope.ico
    - Dependencies
    - Build Specifications
      - Simple Scope
      - Simple Scope 2 unit
      - Band Pass Response
      - Minimum Scope
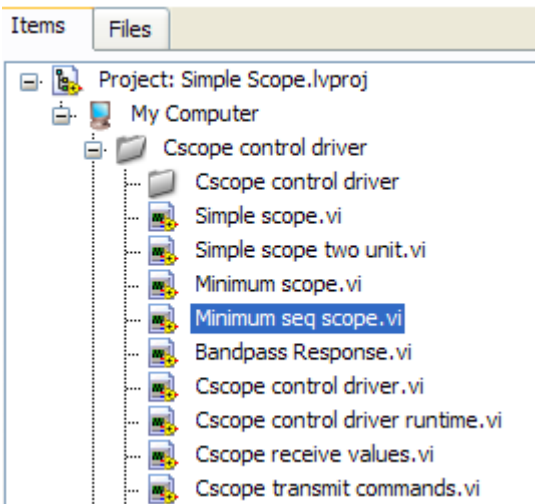   - Minimum Sequential Scope

The Cscope Control Driver entry includes all the dependant vi, which are stored in Cscope control driver.llb.
The first few entries are the major vi's in the project, arranged to be easily opened.

First are the 5 example applications, discussed in the next section.
Next we have the cscope control driver and runtime vi's.
Cscope receive values and Cscope Transmit commands represent the next level down from the Cscope control driver
runtime. Many of the vi's maintain state information, and are therefore re-entrant, with one copy per instance.

Items	Files

- Project: Simple Scope.lvproj
  - My Computer
    - Cscope control driver
      - Cscope control driver
      - Simple scope.vi
      - Simple scope two unit.vi
      - Minimum scope.vi
      - Minimum seq scope.vi
      - Bandpass Response.vi
      - Cscope control driver.vi
      - Cscope control driver runtime.vi
      - Cscope receive values.vi
      - Cscope transmit commands.vi

Example applications

A number of example applications are included with the driver. They can all be found in the 'Cscope control driver.llb'
An executable of each is included so that the environment can be tested before development to ensure that the Labview Runtime, and NI-Visa runtime, and the USB/Ethernet interfaces are working correctly. Build specifications are included for each of them in the 'Simple Scope' project included with the driver.
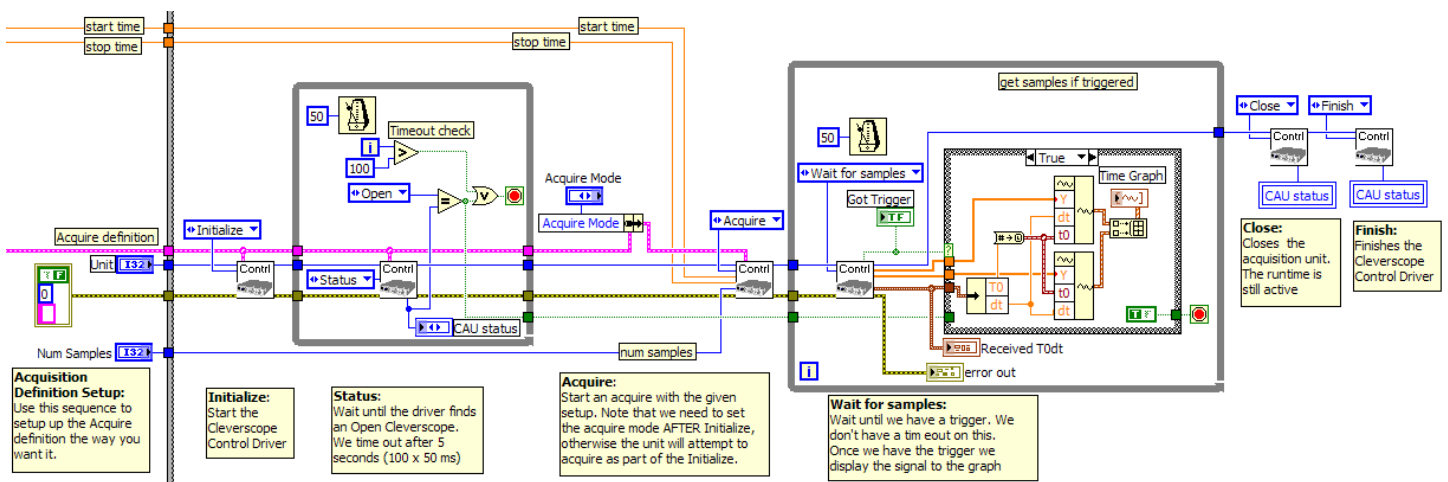
The applications are:
- SimpleScope
  This application implements a simple oscilloscope with knobs to control A, B Volts/Div, Time/Div, the Trigger level and source, and the Signal generator frequency and amplitude. It also includes a button to exercise the Function command. The app includes the ability to choose the interface connection, and to connect to multiple Cleverscope Acquisition Units (CAU).
- SimpleScope 2 Unit
  This application is the same as SimpleScope, but controls two CAU's at the same time.
- Band Pass Response
  This application illustrates use of multi-frame capture to capture up to 3000 sequential frames, with automatic stepping of the signal generator between frame captures to quickly construct the band pass response of a network.
- Minimum Scope
  This very simple application shows the minimum steps need to open, acquire, transfer and close the driver.
- Minimum Sequential Scope
  This very simple application illustrates starting and transferring a sequential frame capture.

Minimum Scope Example

The Minimum Scope application is included with the Control Driver. It gives you a very simple test bench in which to test out ideas. The application shows you the minimum needed to initialize, open, acquire and close the Cscope Control Driver.

Here is a block diagram of the active part of the minimum scope application (in the pdf, zoom to see the detail):
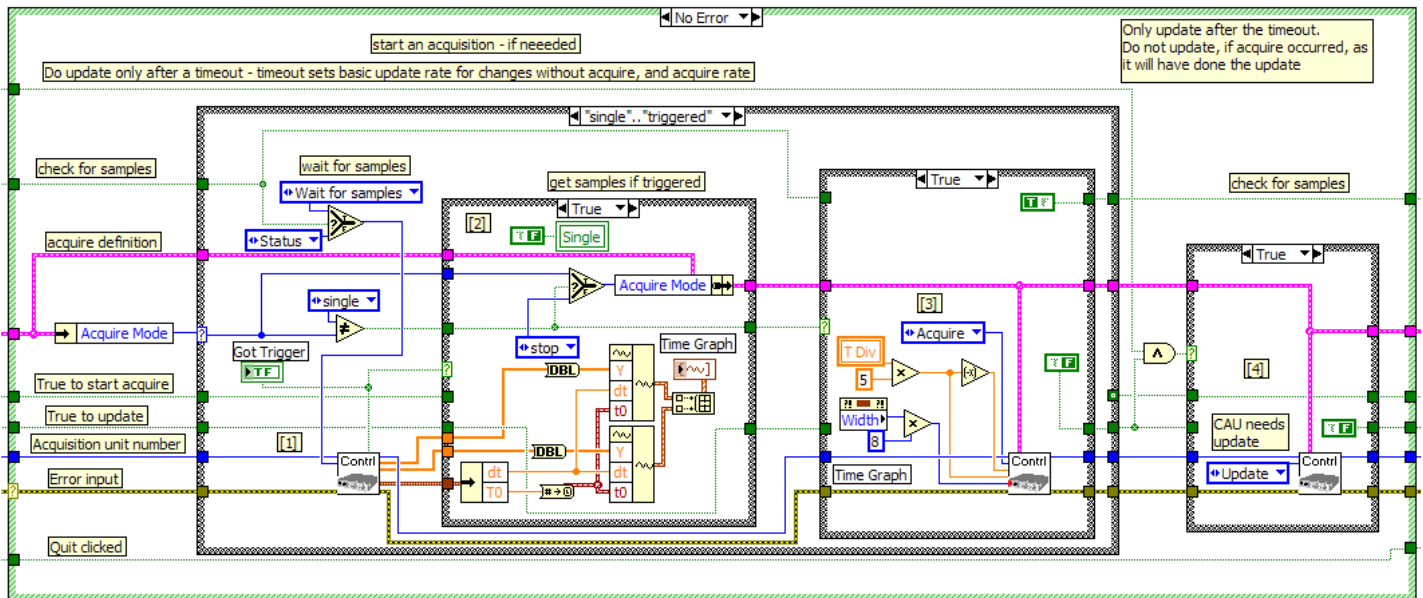
Values you need to wire as inputs:

Item	Step it applies to	Explanation
Acquisition Definition	Initialize, Acquire	Defines the acquisition definition that is used for this capture.
Command	All	Specifies what the Cscope Control Driver will do.
Unit	All (optional)	Specifies which acquisition unit is being controlled 0~7. If not used, addresses acquisition unit 0.
Error In	All	Maintains the error condition, and sets ordering
Start Time	Acquire	The capture start time, relative to the trigger point. Can be + or –. The value -10u means start capturing 10us before the trigger. The value +600m means delay 600 ms after the trigger, then capture.
Stop Time	Acquire	The capture stop time relative to the trigger. The total capture duration will be stop – start time.
Acquire Mode (in Acquire Definition)	Acquire	This specifies how we are going to capture the frames. Make sure it is Stop (3) when calling Initialize. Triggered capture is Single (0) and automatic capture is Auto (1).

These are steps to open the Cleverscope Acquisition Unit, capture a sample set, and then close it down.

Step	Command	Description
1. Acquisition Definition Setup	None	Here you need to ensure that all the values in the acquisition definition variable are correct. Off course you can change these values later. If you change any values make sure you change the value in 'Value Changed' as well, as this forces an update scan in the driver (the driver only updates values that have changed to the acquisition hardware unit). A simple way is to wire the msec timer to value changed.
2. Initialize	Initialize	Initialize the Cscope Control Driver, by wiring the Initialize command. The acquisition unit will be setup as specified by the Acquisition Definition, and the acquisition unit chosen. Make sure that the 'Acquire Mode' value is set to stop, otherwise the unit will do an acquisition following initialization. Initialization, which is opening the port, retrieving the calibration coefficients, and sending the entire acquisition definition to the acquisition unit takes about 1.4 seconds.
3. Status	Status	Wait for Status = Open. This step ensures that further commands are talking to an open system.
4. Acquire	Acquire	Acquire a samples set. You need to set the Acquire mode entry to either Single or Auto to capture the sample set. Single uses Triggered capture. Auto will capture with a trigger, if there is one, otherwise it just captures regardless.
5. Wait for samples	Wait for samples	Iterate until Got Samples is true, and then read the sample set. If you call at 50 msec intervals you will get maximum throughput. Calling any faster makes no gain. The Signal Output is an array that we turn into a waveform in this example.
6.Close	Close	If you have multiple acquisition units connected, you may independently initialize and close a specific acquisition unit while leaving other acquisition units un-affected.
7. Finish	Finish	Finishes operation –closes the runtime system.

Simple Scope example

This is the SimpeScope code that is called every timer tick, and implements the main control system:

Control flow is:

1. If the user has clicked either single or auto, the acquire mode will be set to single or auto triggered.. In addition, as we have not yet asked for samples, the check for samples Boolean will be false. As a result we use the **Status** command to get the CAU status [1]
2. Assuming we do not have an acquisition pending, but we need to start an Acquire, we do not run [2], but proceed to use the **Acquire** command [3], if the Acquire Mode is not Stop.   We set the 'check for samples' flag true.
3. After having started the acquire (which can take a few millisecs), the user may have changed something on the front panel, which will cause the update flag to be set true. If the update flag is true, we issue the **Update** command to update the CAU with the setup in Acquire. We set the update flag off.
4. Assuming we do now have an acquisition pending, we start the loop again. This time [2] issues a **Wait for Samples** command. Once we see Got Trigger as true, we transfer the samples [2] to the graph. We stop acquisition if the acquire mode is Single, and release the Single button. If Auto, we make no change to the Acquire Mode. Processing now carries on from step 2, and repeats

Applying Minimum Scope and the Control Driver

The Minimum Scope application can be used to test variations on the capture theme.

Acquisition Unit based waveform averaging

The acquisition unit can be used to automatically acquire 4, 16, 64 or 128 frames synchronous with the trigger, and then average across all of them, and return one averaged waveform as a result. The result can be rendered as 10, 12 or 14 bits. In this example we will average 16 frames, and return with 12 bit precision

With Minimum Scope set these values:

Acquire.element	Value	Explanation
Acquisition mode	Waveform averaged (4)	This value sets the type of acquisition to do. We want waveform averaged, so we set the Acquire Definition. Acquisition mode to Waveform avg (4)
Waveform averages	16	We want 16 frames to be averaged together.
Num Buffers	17	This is the number of buffers assigned for frame capture. This must always be at least one more than the number of frames captured sequentially (because we reserve one buffer for circular sampling to find the next trigger). In this example we set 'Num Buffers' to 17
Num Seq Frames	16	This is the number of frames we want to capture in sequence =16. Now when a 'Single' trigger is started, the acquisition unit will capture 16 frames in sequence and then carry out the waveform average.
Acquire Mode	Single (0)	We use 'Single' (0), which means capture with a trigger, and start a capture only once, but do a sequence, because Num seq frames>1.
Sampler Resolution	12 bit (1)	Set the returned values to have a resolution of 12 bits.

After setting these values in the Acquire Definition, we run Minimum Scope once, and it captures a waveform of 16 averages, with 12 bit resolution.

Filtering the Signal

The Cleverscope Acquisition unit includes a 20 MHz pre-filter and a 40ns ~ 1.28 us moving average filter for filtering the 100 MSPS captured data stream. When enabled the filters provide real-time results. The filters include time delay compensation so that all channels are time aligned.

For this example we will not use the 20 Mhz prefilter, but use a 640ns moving average filter, and return 12 bit filtered results. Channel B will be set to using the 20 MHz pre-filter only.

With Minimum Scope set these values:

Acquire.element	Value	Explanation

A Bandwidth	10101001b = 0xA9 (169 dec)	A Bandwidth (changed from firmware 4639).
		Bit 0 - Global Filter enable, 0 = no filter, 1 = use filter
		Bit 2:1 - Pre-filter frequency 0 = No filter, 1 = 20 MHz filter, 2 and 3 reserved
		Bit 3: - If true, use the moving average (MA) filter
		Bit 4: - If true use exponential (E) filter - mutually exclusive with moving average filter
		Bits 7:5 - Filter time constant:
		    000 = no filter
		    001 = 40ns MA , 20ns E
		    010 = 80ns MA, 40ns E
		    011 = 160ns MA, 80ns E
		    100 = 320ns MA, 160ns E
		    101 = 640ns MA, 320ns E
		    110 = 1280ns MA, 640ns E
		    111 = reserved MA, 1280ns E
		For the moving average only the channel A moving average value is used, and it also used for Channel B
		If Bit 0 is 0, then all the other bits are ignored.

B Bandwidth	00000011b = 3	Same description as above.
Sampler Resolution	12 bit (1)	Set the returned values to have a resolution of 12 bits.

Sequential Capture

Sequential capture is the process of capturing a set of N frames, each started by a separate trigger, and then transferring the samples captured to the PC. The sequential frame capture system requires about 200us between triggers minimum.

Examples of sequential frame capture are in 'Bandpass Response.vi' and 'Minimum seq scope' included with the Cscope Control Driver package. TheBandpass response example uses the acquisition unit sig gen auto step facility to step the signal generator after every trigger. The example steps the signal generator over a user defined frequency range (eg 1M to 3M Hz) in steps (eg 1 kHz), using sequential capture to capture up to 3000 separate triggered frames. It then transfers the sequence to the PC, and does an FFT on each frame, and plots the results to a peak captured graph. This shows the bandpass response of any network that might be between the signal generator output and the Channel A/B inputs. Because all of the capturing and stepping is done automatically in the acquisition unit, this process is fast.

The Minimum seq scope example shows the minimum needed to capture and display a sequence of frames. For this example we will assume we capture 200 triggered frames, each of 100us duration, and 1000 samples per frame. We will start capturing 50us before the trigger, and stop 50us after the trigger. We will transfer all the samples.

Setup

We assume the values on amplitude scale, coupling, bandwidth etc are all setup as required. This example only works through the values needed to meet what we want to do.

Acquire Definition

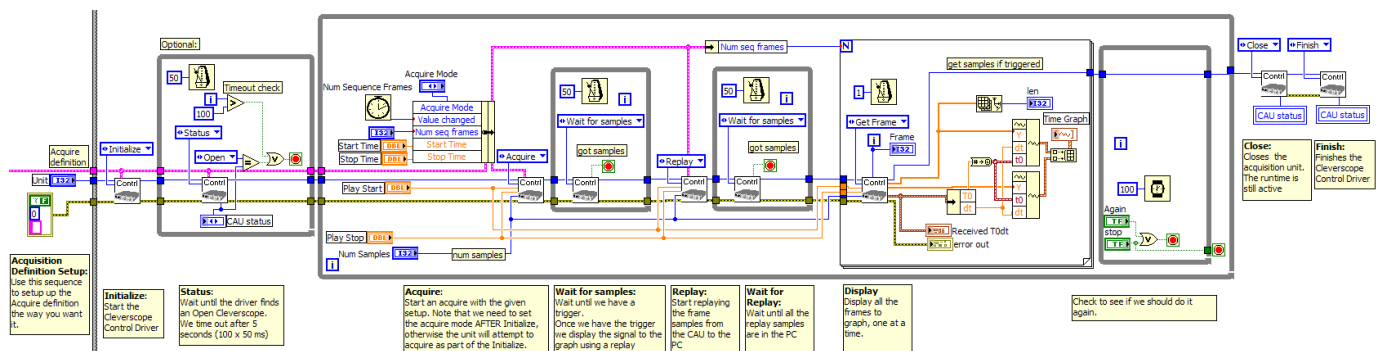| Item	Explanation
Start Time -50us	The capture start time, relative to the trigger point. Can be + or –. The value -10u means start capturing 10us before the trigger. The value 600m means start capturing 600 ms after the trigger (ie delay after the trigger). We use -10u.
Stop Time +50us	The capture stop time relative to the trigger. We will use 40us. The total capture duration will be 50us.
Num Buffers 1000	This is the number of buffers assigned for frame capture. This must always be at least one more than the number of frames captured sequentially (because we reserve one buffer for circular sampling to find the next trigger). In this example we set 'Num Buffers' to 1000 in the Acquire definition.
Num Sequence Frames 200	This is the number of frames we want to capture in sequence. Set this to 200. Now when a 'Single' trigger is started, the acquisition unit will capture 200 frames in sequence and then signal completion.
Acquire Mode 0 (Single)	This specifies how we are going to capture the frames. We use 'Single' (0), which means capture with a trigger, and start a capture only once.
Transfer Size Sequence (6)	This value specifies the size and type of transfer being used. 0 is Normal. The value we want is 'Sequence' (6) which means the acquisition will transfer a full sequence whenever it is told to do a transfer. We set up the Acquire definition to sequence.

Input values to specify

In addition to the Acquire Definition, we have values that specify the playback. For Cleverscope the capture process is setup independently of the playback process (think of a tracker graph displaying 10us of a signal that is 800ms wide). So we need to specify these values:

Item	Explanation
Replay Start Time -50u	The replay start time, relative to the trigger point. Can be + or –. The value -5u means start displaying 5us before the trigger. The value 300m means start displaying 300ms ms after the trigger. We use -50u, which is the same as the capture start time (it can be different).

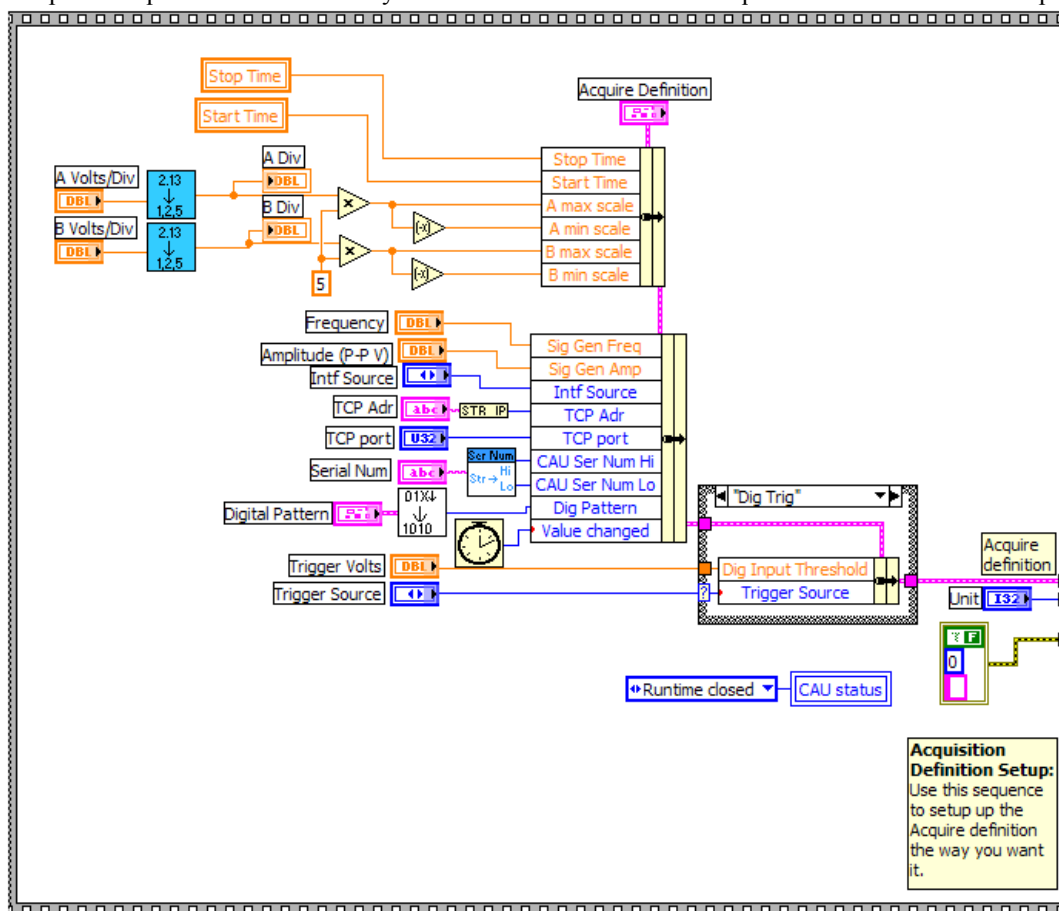| Replay Stop Time +50u	The replay stop time relative to the trigger. We will use 50us. The total replay duration will be 1050us, which is the same as the capture duration.
Num Samples 1000	This specifies how many samples we want in our replay. It may be truncated. The Cleverscope acquisition unit might capture 4M samples for a particular signal. We might only want to playback 20,000. The replay system automatically decimates the samples in the acquisition unit to meet the replay values. In this example we want 1000 samples.
Acquisition Unit 0	We can control up to 32 concurrent units. We only want one for now, so we leave the Acquisition Unit input unwired, to default to 0.

Use

The diagram below shows the sequence of operations. If you are looking at the PDF, zoom in on it.
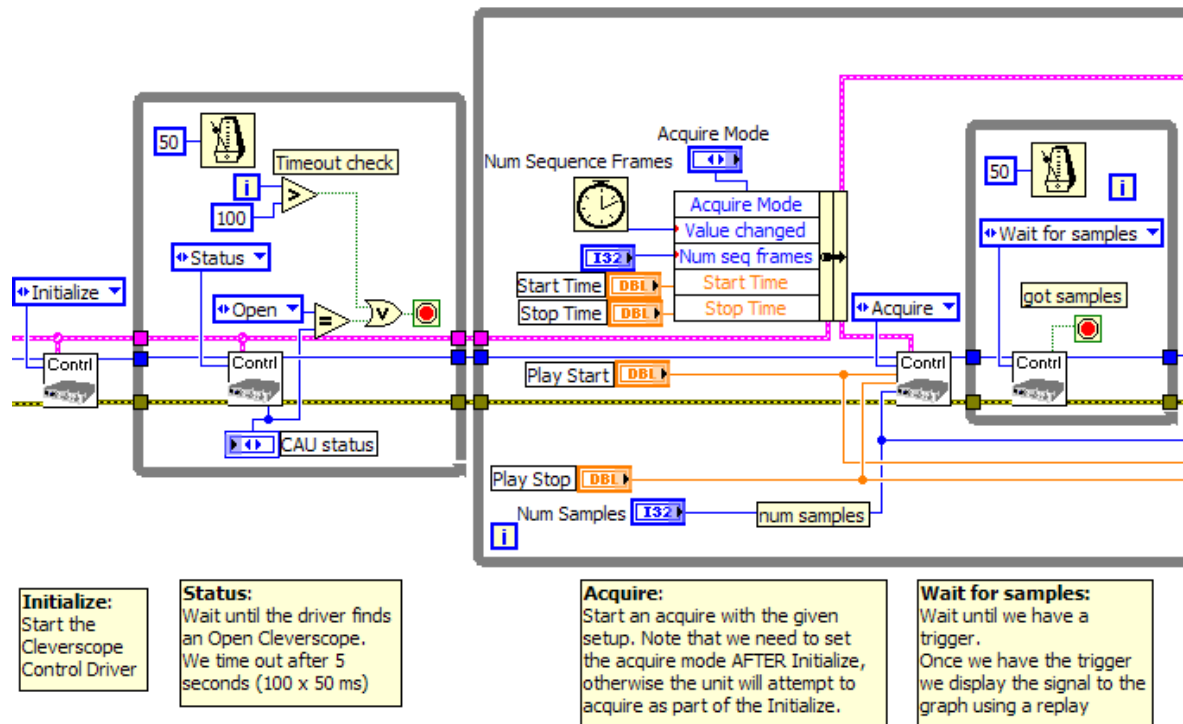

At a minimum we need to do these things:

1. Setup the Acquire definition the way we want it. We do the actual sequential stuff in the runtime part.
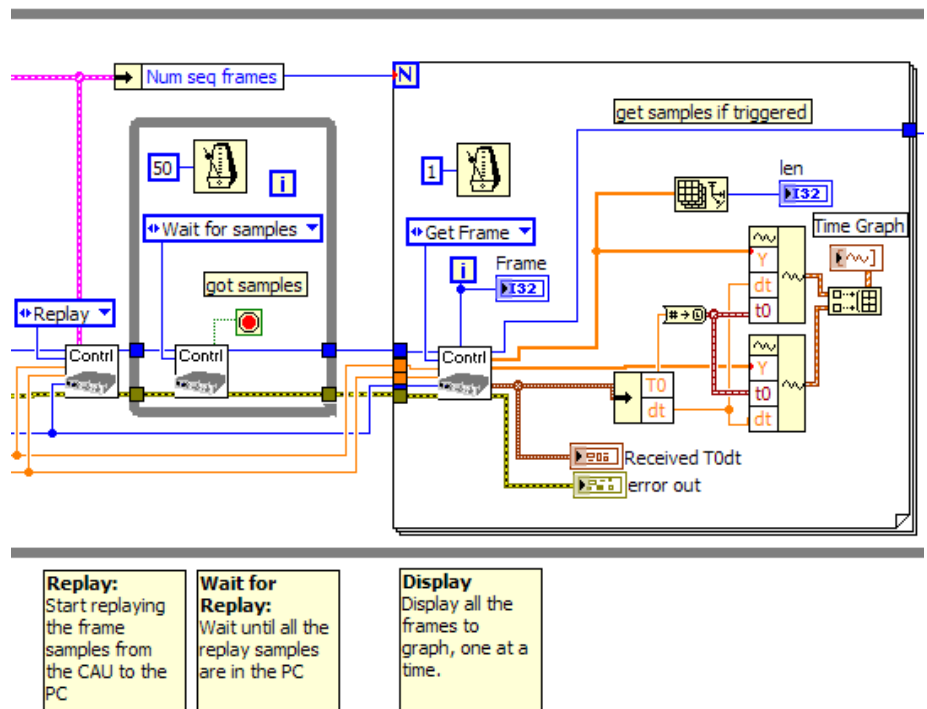

2. Call Cscope Control Driver.vi with the Command 'Initialize' (0). The acquire definition is wired.

3. Call the Control driver until the CAU status is 'Open'.
4. Call the control driver with the Command 'Acquire' (1). The Acquire definition is wired, as are the Replay Start Time, Replay Stop Time and Samples in Replay. We also wire the number of sequence frames (if num seq frames = 1, we just acquire one frame, anything larger than this will trigger a sequential capture), and the number of samples in the replay.
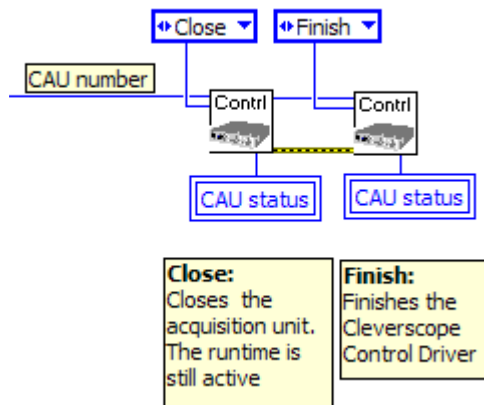

Initialize:
Start the Cleverscope Control Driver

Status:
Wait until the driver finds an Open Cleverscope. We time out after 5 seconds (100 x 50 ms)

Acquire:
Start an acquire with the given setup. Note that we need to set the acquire mode AFTER Initialize, otherwise the unit will attempt to acquire as part of the Initialize.

Wait for samples:
Wait until we have a trigger.
Once we have the trigger we display the signal to the graph using a replay

The Acquire Mode is set on the front panel to single. The acquire definition was manually set to include the

number of buffers allocated to 1000. . Once the buffers are allocated we can proceed to acquire up to num_buffers – 1 frames to the buffers.

5. Following Acquire, we call the control driver with 'Wait for Samples' until we see that the samples have been captured. Each frame will have been individually triggered until the number of frames specified have been captured.
6. Once the samples are in CAU memory, we need to transfer them as a frame transfer to the PC (Acquire does include the capability to automatically transfer the latest frame to the PC, but this is not sufficient to transfer all the frames.). This is done by starting a Replay. Because we captured a sequence, the replay will replay the full sequence.
7. After starting the replay we need to make sure all the samples have been transferred to PC memory. We use **Wait for Samples** for this.
Once we have the samples in PC memory, we can get the frames, one at a time from PC memory, and process them as we wish. Here we display them. We use the **Get Frame** command to do this.

Replay:
Start replaying the frame samples from the CAU to the PC

Wait for Replay:
Wait until all the replay samples are in the PC

Display
Display all the frames to graph, one at a time.

8. Following completion of the sequence capture, we give the user the opportunity to do the capture again.
9. If the user does not start again, we complete the acquisition by closing the CAU.

For greatest reliability and flexibility, first **Close** the connections you have made for each CAU, and then call **Finish**.

Auto step Signal Generator Control

It is possible to automatically step the signal generator following an acquisition. This is useful for carrying out voltage/frequency sweeps while capturing a sequence. Here we use the Bandpass response example to show how this was done.
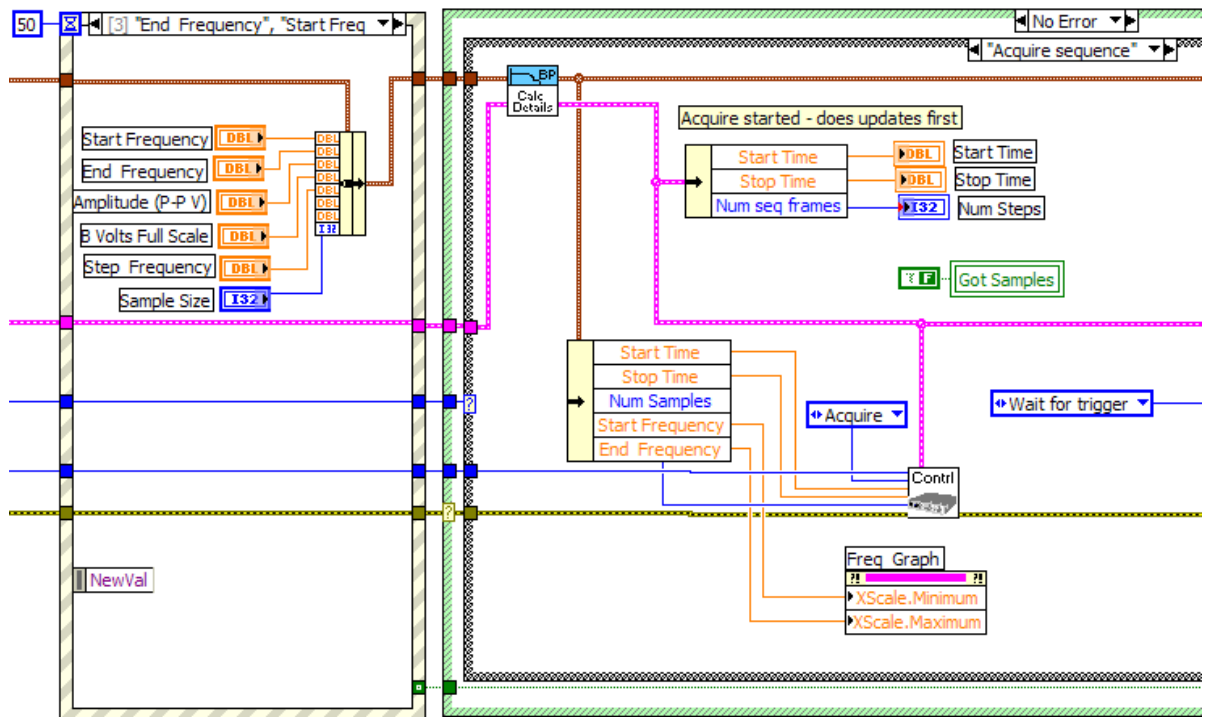
Acquire Definition

Item	Explanation
Sig Gen Func 1	Defines the signal generator function. Without Auto stepping we use 'Standard' (0). We want 'Auto advance' (1) to automatically step the signal generator after an acquisition is complete. This happens in the acquisition unit.
Sig Gen Freq Step 1000	Specifies the step size in Hz. In our example, it is user configurable, and defaults to 1kHz.
Sig Gen Freq	Specifies the start frequency for auto stepping. In our example we use 1MHz

We must update the Acquire Definition and input it to the Cscope Control Driver when calling the driver with the Acquire or Update commands.
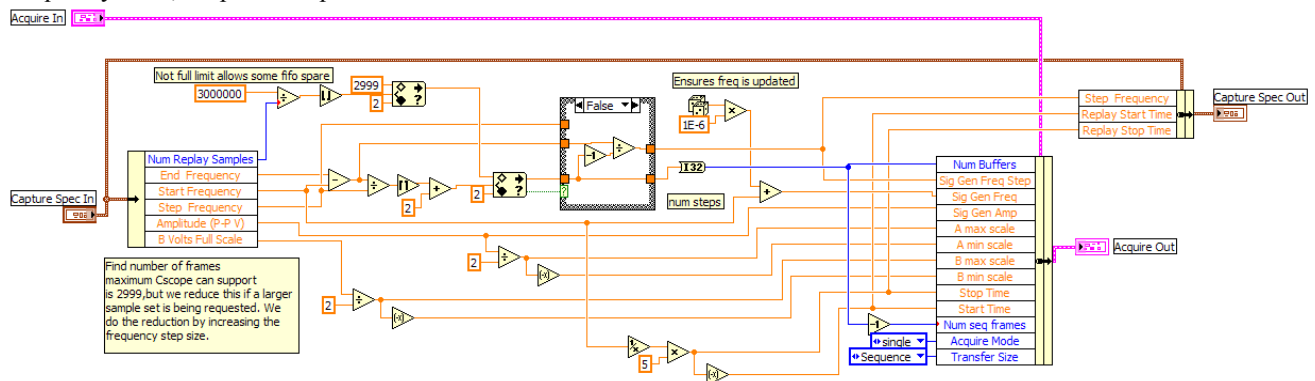
Use

At a minimum we need to do these things:
1. Setup the Acquire definition the way we want it.
2. Call the control driver with the command **Acquire** (1). In the example below we show how the bandpass filter application sets up the start, stop and step frequencies. We have also shown the 'Acquire sequence' pane which captures a signal based on the latest setup:
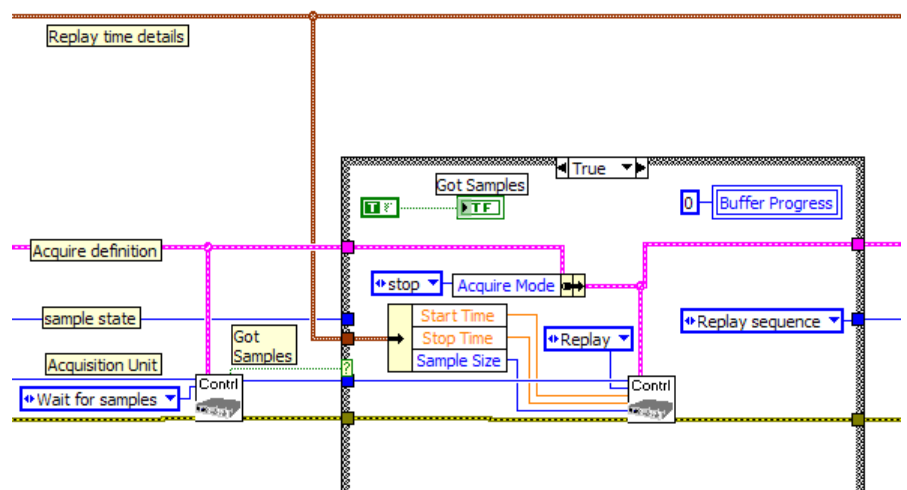
3. Here is the Calc Details vi, which shows the Acquire Definition being loaded with new values for the Frequency Start, Stop and Step values:
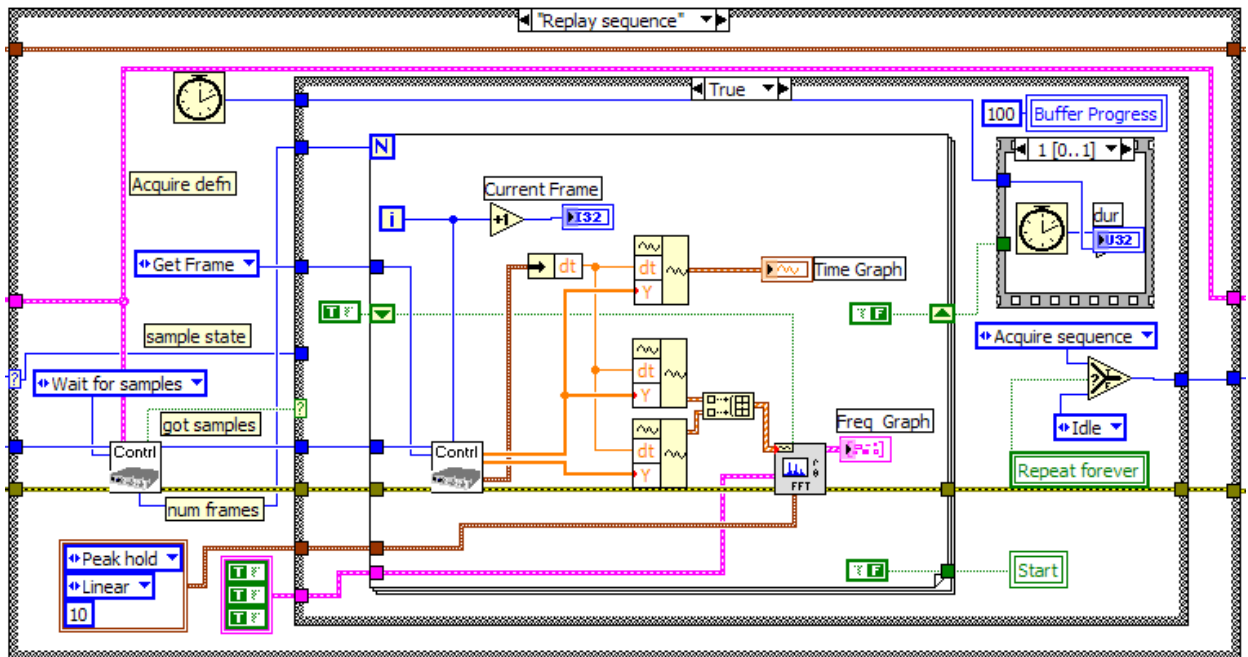

This vi covers a bit more than just frequency stepping, but you can see that we put values into the Sig Gen Freq Step and Sig Gen Freq elements. The Sig Gen Func is set up at initialization time.

4. Once we have started an acquisition (samples state = wait for trigger) we wait for samples as shown :

5. Once we have started the replay, we must wait until the frame set is transferred into the PC. This is the 'replay sequence' state:

Once the **Wait for samples** command returns Got Samples true, we send each frame to the graph, using the **Get Frame** command, and also calculate the spectra with peak averaging, and display that