NI-SCOPE Instrument Driver Quick Reference Guide

Easy Programming for National Instruments Oscilloscopes

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Initiate and Close Functions				
	**niScope_init** Creates a new session to the instrument.	ViRsrc ViBoolean ViBoolean ViSession *	resourceName IDQuery resetDevice vi	DAQ::#, where # is the device number NISCOPE_VAL_TRUE, NISCOPE_VAL_FALSE NISCOPE_VAL_TRUE, NISCOPE_VAL_FALSE Reference to the new session
	**niScope_close** Closes the current session to the instrument.	ViSession	vi	Session handle
Application Functions				
	**APP_EasyAcquire** Shows the basics of acquiring data. Configures the scope to acquire one record of data specified in time per record. Initiates the acquisition, waits for it to complete, and returns the acquired data.	For parameters and other function-specific information, see `niScope APP Easy Acquire.vi` for LabVIEW or `App_EasyAcquire.c` for CVI and Microsoft Visual C++ (MSVC)		
	**APP_TimeBaseAcquire** Configures the scope to acquire one record of data specified in time per record. Initiates acquisition, waits for it to complete, and returns the acquired data.	For parameters and other function-specific information, see `niScope APP Time Base Acquire.vi` for LabVIEW or `APP_TimeBaseAcquire.c` for CVI and MSVC		
	**APP_SampleRateAcquire** Configures the scope to acquire one record of data specified in sample rate. Initiates the acquisition, waits for it to complete, and returns the acquired data.	For parameters and other function-specific information, see `niScope APP Sample Rate Acquire.vi` for LabVIEW or `APP_SampleRateAcquire.c` for CVI and MSVC		
	**APP_BinaryAcquire** Configures the scope to acquire one record of 8-bit binary data specified in sample rate. Initiates acquisition, waits for it to complete, and returns the acquired data.	For parameters and other function-specific information, see `niScope APP Binary Acquire.vi` for LabVIEW or `APP_BinaryAcquire.c` for CVI and MSVC		
	**APP_MultiRecordAcquire** Configures the scope to acquire multiple records of data specified in sample rate. Initiates the acquisition, waits for it to complete, and returns the acquired data.	For parameter information, see `niScope APP Multi Record Acquire.vi` for LabVIEW or `APP_MultiRecordAcquire.c` for CVI and MSVC		
	**APP_MultiChannelAcquire** Configures the scope to acquire one record of data specified in sample rate for two channels. Initiates the acquisition, waits for it to complete, and returns the acquired data. Trigger occurs only on channel 0, but both channels 0 and 1 acquire simultaneous data.	For parameters and other function-specific information, see `niScope APP Multi Channel Acquire.vi` for LabVIEW or `APP_MultiChannelAcquire.c` for CVI and MSVC		

[1] Function name for C, C++, LabWindows/CVI, and Visual Basic.

[2] In C, C++, and LabWindows/CVI, constant names such as NISCOPE_VAL_TRUE and NISCOPE_VAL_EDGE refer to the use of # defines in your program. In LABVIEW, these constants refer to boolean or ring controls with corresponding entries. For example, NISCOPE_VAL_EDGE corresponds to the LabVIEW ring control entry "Edge." Refer to LabVIEW online help (Show Help) for more details.

322575B-01 *April 2000*

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Configuration Functions				
	**niScope_ConfigureAcquisition** Configures the oscilloscope acquisition mode.	ViSession	vi	Session handle
		ViInt32	acquisitionType	NISCOPE_VAL_NORMAL, NISCOPE_VAL_FLEXRES, NISCOPE_VAL_PEAK_DETECT
	**niScope_AutoSetup** Automatically configures the instrument. When you call this function, the oscilloscope senses the input signal and automatically configures many of the instrument settings.	ViSession	vi	Session handle
	**niScope_ConfigureVertical** Configures the common properties of the oscilloscope's vertical subsystem for the specified channel.	ViSession	vi	Session handle
		ViConstString	channel	Channel to configure
		ViReal64	range	Volts peak-to-peak; varies depending on product
		ViReal64	offset	Location of the center of the range parameter
		ViInt32	coupling	NISCOPE_VAL_AC, NISCOPE_VAL_DC
		ViReal64	probeAttenuation	Any positive real number such as 1, 10, and 100
		Vi Boolean	enabled	NISCOPE_VAL_TRUE, NISCOPE_VAL_FALSE
	**niScope_ConfigureChanCharacteristics** Configures the common properties of the oscilloscope's specified channel.	ViSession	vi	Session handle
		ViConstString	channel	Channel to configure
		ViReal64	inputImpedance	NISCOPE_VAL_50_OHM, NISCOPE_VAL_75_OHM, NISCOPE_VAL_1_MEG_OHM
		ViReal64	bandwidth	0 —Use the hardware's default value; check your hardware user manual to find a device's other supported bandwidths in hertz
	**niScope_ConfigureHorizontal** Configures the common properties of the horizontal subsystem for a single record acquisition specified in terms of time per record.	ViSession	vi	Session handle
		ViReal64	timePerRecord	Time duration of the record in seconds
		ViInt32	minNumPts	Minimum number of points you need in the record for each channel; call **niScope_ActualRecordLength** for the actual acquired record length
		ViReal64	refPosition	Percent of waveform record that is pretriggered
	**niScope_ConfigureHorizontalRate** Configures the common properties of the horizontal subsystem for a single record acquisition specified in terms of minimum sample rate.	ViSession	vi	Session handle
		ViReal64	minSampleRate	Minimum sampling rate for the acquisition in samples per second
		ViInt32	minNumPts	Minimum number of points you require in the record for each channel; call **niScope_ActualRecordLength** for the actual record length acquired
		ViReal64	refPosition	Percent of waveform record that is pretriggered
	**niScope_ConfigureMultiHorizontal** Configures the common properties of the horizontal subsystem for a multi-record acquisition specified in terms of time per record.	ViSession	vi	Session handle
		ViReal64	timePerRecord	Time duration of the record in seconds
		ViInt32	minNumPts	Minimum number of points you need in the record for each channel; call **niScope_ActualRecordLength** for the actual record length acquired
		ViReal64	refPosition	Percent of waveform record that is pretriggered
		ViInt32	numRecords	Number of waveform records to acquire
	**niScope_ConfigureMultiHorizontalRate** Configures the common properties of the horizontal sub-system for a multi-record acquisition specified in terms of minimum sample rate.	ViSession	vi	Session handle
		ViReal64	minSampleRate	Minimum sampling rate for the acquisition in samples per second
		ViInt32	minNumPts	Minimum number of points you need in the record for each channel; call **niScope_ActualRecordLength** for the actual record length acquired
		ViReal64	refPosition	Percent of waveform record that is pretriggered
		ViInt32	numRecords	Number of waveform records to acquire

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Configuration Functions (Continued)				
	**niScope_ConfigureClock** Configures the properties for synchronizing the oscilloscope to an external clock or for sending the oscilloscope's clock out as a synchronizing clock for other oscilloscopes.	ViSession	vi	Session handle
		ViConstString	inputClockSource	NISCOPE_VAL_NO_SOURCE, NISCOPE_VAL_RTSI_CLOCK, NISCOPE_VAL_PFI_<1..2>, NISCOPE_VAL_PXI_CLOCK
		ViConstString	outputClock-Source	NISCOPE_VAL_NO_SOURCE, NISCOPE_VAL_RTSI_CLOCK, NISCOPE_VAL_PFI_<1..2>
		ViConstString	clockSyncPulse-Source	NISCOPE_VAL_NO_SOURCE, NISCOPE_VAL_RTSI_<0..6>, NISCOPE_VAL_PFI_<1..2>
		ViBoolean	masterEnabled	NISCOPE_VAL_TRUE, NISCOPE_VAL_FALSE
	**niScope_ActualRecordLength** Returns the actual number of points the oscilloscope acquires for each channel.	ViSession	vi	Session handle
		ViInt32 *	actualRecord-Length	Value is equal to or greater than the minimum number of points you specify with a horizontal configuration function; length of record is available for each channel
	**niScope_SampleRate** Returns the effective sample rate of the acquired waveform using the current configuration in samples per second.	ViSession	vi	Session handle
		ViReal64	actualSampleRate	Returns the effective sample rate of the waveform acquired for each channel
	**niScope_ConfigureTriggerSource** Configures the common properties of the trigger subsystem.	ViSession	vi	Session handle
		ViConstString	triggerSource[3]	NISCOPE_VAL_IMMEDIATE, "0", "1", NISCOPE_VAL_EXTERNAL, NISCOPE_VAL_SW_TRIG_FUNC (see **niScope_SendSWTrigger**), NISCOPE_VAL_RTSI_<0..6>, NISCOPE_VAL_PFI_<1..2>, NISCOPE_VAL_TTL<0..6>, NISCOPE_VAL_PXI_STAR
		ViInt32	triggerType	NISCOPE_VAL_EDGE, NISCOPE_VAL_HYSTERESIS, NISCOPE_VAL_DIGITAL
		ViReal64	triggerDelay	Time to wait after the trigger before marking the reference position in seconds
		ViReal64	holdoff	Time to wait between one acquisition and arming for a trigger for another acquisition of a multi-record acquisition in seconds
	**niScope_ConfigureEdgeTrigger** Configures the edge trigger. An edge trigger occurs when the trigger signal passes through the voltage threshold that you specify with the level parameter. Its slope is specified with the slope parameter.	ViSession	vi	Session handle
		ViReal64	level	Voltage threshold for edge triggering
		ViInt32	triggerCoupling	NISCOPE_VAL_AC, NISCOPE_VAL_DC
		ViInt32	slope	NISCOPE_VAL_POSITIVE, NISCOPE_VAL_NEGATIVE
	**niScope_ConfigureHysteresisTrigger** Configures the hysteresis trigger. If the slope parameter is set to positive, a trigger occurs if the trigger signal starts at a voltage below the level parameter minus the hysteresis parameter and then crosses above the voltage of the level parameter. If slope is set to negative, a trigger occurs if the trigger signal starts at a voltage above the level plus the hysteresis parameters and then crosses below the voltage of the level parameter.	ViSession	vi	Session handle
		ViReal64	level	Voltage threshold for edge triggering
		ViReal64	hysteresis	Size of the hysteresis window in volts
		ViInt32	triggerCoupling	NISCOPE_VAL_AC, NISCOPE_VAL_DC
		ViInt32	slope	NISCOPE_VAL_POSITIVE, NISCOPE_VAL_NEGATIVE
	**niScope_ConfigureDigitalTrigger** Configures the digital trigger. A digital trigger occurs when the trigger signal has the slope that you specify with the slope parameter.	ViSession	vi	Session handle
		ViInt32	slope	NISCOPE_VAL_POSITIVE, NISCOPE_VAL_NEGATIVE
	**niScope_ConfigureTriggerOutput** Configures the oscilloscope to generate a signal pulse that other scopes can detect when configured for digital triggering. The trigger event parameter specifies what condition causes the oscilloscope to generate the signal pulse. The trigger output source parameter specifies the hardware source on which the signal pulse will be generated.	ViSession	vi	Session handle
		ViInt32	triggerEvent	NISCOPE_VAL_NO_EVENT, NISCOPE_VAL_STOP_TRIGGER_EVENT
		ViConstString	triggerOutput	NISCOPE_VAL_NO_SOURCE, NISCOPE_VAL_RTSI_<0..6>, NISCOPE_VAL_PFI_<1..2>, NISCOPE_VAL_TTL<0..6>, NISCOPE_VAL_PXI_STAR

[3] In LabVIEW, triggerSource is implemented as a string control. LabVIEW 5.1 and any subsequent versions implement a ring control for this parameter. However, if you use LabVIEW 5.0, you must enter the specific string constant. Refer to the LabVIEW online help (Show Help) for these constant names.

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Acquisition Functions				
	**niScope_InitiateAcquisition** Initiates a waveform acquisition. After you call this function, the oscilloscope leaves the idle state and waits for a trigger. The oscilloscope acquires a waveform for each channel you have enabled with **niScope_ConfigureVertical**.	ViSession	vi	Session handle
	**niScope_WaitForAcquisitionToFinish** Polls your oscilloscope by calling **niScope_AcquisitionStatus** until the acquisition is complete. If the acquisition does not finish within the maximum time you specify, the function returns an error.	ViSession ViInt32	vi maximumTime	Session handle Specifies the maximum time length for an acquisition to complete in milliseconds
	**niScope_Abort** Aborts an acquisition and returns the oscilloscope to the idle state. Acquisition initiated with the **niScope_ReadWaveform** or **niScope_InitiateAcquisition** functions.	ViSession	vi	Session handle
	**niScope_AcquisitionStatus** Shows if an acquisition is in progress or complete.	ViSession ViInt32 *	vi status	Session handle NISCOPE_VAL_ACQ_IN_PROGRESS (0) NISCOPE_VAL_ACQ_COMPLETE (1)
	**niScope_SendSWTrigger** Sends a command to trigger the oscilloscope. Call if you pass NISCOPE_VAL_SW_TRIG_FUNC for the trigger source parameter of **niScope_ConfigureTriggerSource**.	ViSession	vi	Session handle
	**niScope_ReadWaveform** Initiates an acquisition on all the channels that you enable with **niScope_ConfigureVertical**, waits for the acquisition to complete, and returns the waveform for the channel you specify. **niScope_FetchWaveform** obtains the waveforms for each of the remaining channels.	ViSession ViConstString ViInt32 ViInt32 ViReal64 ViInt32 * ViReal64 * ViReal64 *	vi channel waveformSize maxTime waveformArray[ ] actualPoints initialX xIncrement	Session handle Channel to acquire from Number of elements to put in the waveform array Maximum length of time to allow the read waveform operation to complete in milliseconds Waveform that the oscilloscope acquired Actual number of points placed in the waveform array Time of the first point in the waveform array in seconds; time is relative to the reference position Time increment between points in the waveform array in seconds
	**niScope_ReadMinMaxWaveform** Initiates a peak detect acquisition on all the channels that you enable with **niScope_ConfigureVertical**, waits for the acquisition to complete, and returns the minimum and maximum waveforms for the channel you specify. The two waveforms are simultaneously sampled. **niScope_FetchMinMaxWaveform** obtains the waveforms for each of the remaining channels.	ViSession ViConstString ViInt32 ViInt32 ViReal64 ViReal64 ViInt32 * ViReal64 * ViReal64 *	vi channel waveformSize maxTime minWaveform[ ] maxWaveform[ ] actualPoints initialX xIncrement	Session handle Channel to acquire from Number of points to insert into each of the minWaveform and maxWaveform arrays Maximum length of time to allow the read waveform operation to complete in milliseconds Minimum waveform that the oscilloscope acquired Maximum waveform that the oscilloscope acquired Actual number of points placed into each of the minWaveform and maxWaveform arrays Time of the first point in the waveform array in seconds; time is relative to the reference position Time increment between points in the waveform array in seconds

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Acquisition Functions (Continued)				
	**niScope_FetchWaveform** Returns the waveform the oscilloscope acquired for the channel you specify. The waveform is from a previously initiated acquisition.	ViSession	vi	Session handle
		ViConstString	channel	Channel to fetch waveform from
		ViInt32	waveformSize	Number of elems. to place in the waveform array
		ViReal64	waveformArray[ ]	Waveform that the oscilloscope acquired
		ViInt32 *	actualPoints	Actual number of points placed in the waveform array
		ViReal64 *	initialX	Time of the first point in the waveform array in seconds; time is relative to the reference position
		ViReal64 *	xIncrement	Time increment between points in the waveform array in seconds
	**niScope_FetchWaveformFromOffset** Returns the part of the waveform from the offset you supply for the channel you specify. The waveform is from a previously initiated acquisition.	ViSession	vi	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViConstString	channelName	
		ViInt32	retrievalOffset	Offset (in samples) within the record you would like to retrieve from; driver retrieves the waveform starting at this offset
		ViInt32	waveformSize	
		ViReal64	waveformArray[ ]	
		ViInt32 *	actualPoints	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViReal64 *	initialX	
		ViReal64 *	xIncrement	
	**niScope_FetchMinMaxWaveform** Returns the minimum and maximum waveforms from a peak detect acquisition. The acquisition must have been previously initiated in peak detect mode. The two waveforms are simultaneously sampled.	ViSession	vi	See the **niScope_FetchWaveform** function earlier in this document for parameter descriptions
		ViConstString	channelName	
		ViInt32	waveformSize	
		ViReal64	minWaveform[ ]	Minimum acquired waveform
		ViReal64	maxWaveform[ ]	Maximum acquired waveform
		ViInt32 *	actualPoints	See the **niScope_FetchWaveform** function earlier in this document for parameter descriptions
		ViReal64 *	initialX	
		ViReal64 *	xIncrement	
  	**niScope_FetchBinary8Waveform** Returns the waveform the oscilloscope acquires for the channel you specify in 8-bit binary form. The waveform is from a previously initiated acquisition. For 16-bit binary form, use **niScope_FetchBinary16Waveform**. For 32-bit binary form, use **niScope_FetchBinary32Waveform**.	ViSession	vi	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViConstString	channelName	
		ViInt32	retrievalOffset	Offset (in samples) within the record you would like to retrieve from; driver retrieves the waveform starting at this offset
		ViInt32	waveformSize	
		ViInt8/16/32	waveformArray[ ]	
		ViInt32 *	actualPoints	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViReal64 *	initialX	
		ViReal64 *	xIncrement	
		ViReal64 *	gainFactor	Voltage value that is represented by the full-scale level of the binary data; for use in reconstructing voltage data after the acquisition: *Voltage = waveform array element * gain factor + vertical offset*
		ViReal64 *	verticalOffset	Vertical offset (in volts) of the acquisition; for use in reconstructing the voltage data after the acquisition: *Voltage = waveform array element * gain factor + vertical offset*

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Acquisition Functions (Continued)				
	**niScope_FetchMultiWaveform** Returns the waveform the oscilloscope acquires for the record and channel you specify. The waveform is from a previously initiated acquisition.	ViSession	vi	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViConstString	channelName	
		ViInt32	recordNumber	Record number you want to retrieve from the channel indicated in channelName parameter
		ViInt32	retrievalOffset	Offset (in samples) in the record you want to retrieve from; driver retrieves the waveform starting at this offset
		ViInt32	waveformSize	
		ViReal64	waveformArray[ ]	
		ViInt32 *	actualPoints	See the **niScope_FetchWaveform** function earlier in this document for comments
		viReal64 *	initialX	
		ViReal64 *	xIncrement	
	**niScope_FetchMultiMinMaxWaveform** Returns the minimum and maximum waveforms from a peak-detect acquisition for the channel you specify.	ViSession	vi	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViConstString	channelName	
		ViInt32	recordNumber	Record number you want to retrieve from the channel indicated in channelName parameter
		ViInt32	retrievalOffset	Offset (in samples) in the record you want to retrieve from; driver retrieves the waveform starting at this offset
		ViInt32	waveformSize	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViReal64	maxWaveform Array [ ]	Returns the maximum waveform that the oscilloscope acquired
		ViReal64	minWaveform Array [ ]	Returns the minimum waveform that the oscilloscope acquired
		ViInt32 *	actualPoints	See the **niScope_FetchWaveform** function earlier in this document for comments
		viReal64 *	initialX	
		ViReal64 *	xIncrement	
  	**niScope_FetchMultiBinary8Waveform** Returns the waveform the oscilloscope acquires for the record and channel you specify in 8-bit binary form. The waveform is from a previously initiated acquisition. For 16-bit binary form, use **niScope_FetchMultiBinary16Waveform**. For 32-bit binary form, use **niScope_FetchMultiBinary32Waveform**.	ViSession	vi	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViConstString	channelName	
		ViInt32	recordNumber	Record number you want to retrieve from the channel indicated in channelName
		ViInt32	retrievalOffset	Offset (in samples) in the record you want to retrieve from; the driver retrieves the waveform starting at this offset
		ViInt32	waveformSize	
		ViInt8/16/32	waveformArray[ ]	
		ViInt32 *	actualPoints	See the **niScope_FetchWaveform** function earlier in this document for comments
		ViReal64 *	initialX	
		ViReal64 *	xIncrement	
		ViReal64 *	gainFactor	For comments on the gainFactor and verticalOffset, see **niScope_FetchBinary8Waveform**
		ViReal64 *	verticalOffset	
Error Function				
	**niScope_errorHandler** Translates an error code and its source into a detailed error description.	ViSession	vi	Session handle
		ViInt32	errorCode	Error code to translate
		ViChar	errorSource [MAX_FUNCTION_NAME_ SIZE]	Function returning the error code, can be VI_NULL
		ViChar	errorDescription [MAX_ERROR_ DESCRIPTION]	Translated description
Utility Functions				
	**niScope_reset** Resets the instrument to a known state.	ViSession	vi	Session handle

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Utility Functions (Continued)				
	**niScope_self_test** Runs the instrument's self-test routine and returns the test result(s).	ViSession ViInt16 * ViChar	vi selfTestResult selfTestMessage [IVI_MAX_ MESSAGE_BUF_ SIZE]	Session handle 0–Self test passed, 1–Self test failed Self-test response string from the instrument; see the device user manual for an explanation of the string's contents
	**niScope_revision_query** Returns the revision numbers of the instrument driver and instrument firmware.	ViSession ViChar ViChar	vi instrumentDriver-Revision[IVI_MAX_ MESSAGE_BUF_ SIZE] firmwareRevision [IVI_MAX_ MESSAGE_BUF_ SIZE]	Session handle Instrument driver software revision numbers in the form of a string Instrument firmware revision numbers in the form of a string
	**niScope_ProbeCompensationSignalStart** Starts the square wave output on PFI 1 for probe compensation.	ViSession	vi	Session handle
	**niScope_ProbeCompensationSignalStop** Stops the square wave output on PFI 1 for probe compensation.	ViSession	vi	Session handle
Waveform Measurement Functions				
	**niScope_AddWaveformProcessing** Adds one array measurement to the list of processing steps. It is completed before any other measurements.	ViSession ViConstString ViInt32	vi channelName measFunction	Session handle Channel to add processing to Array measurement to add; see *NI-SCOPE Function Reference Help* for constants
	**niScope_ClearWaveformProcessing** Clears the waveform processing on the channel you specify.	ViSession ViConstString	vi channelName	Session handle Channel to clear processing
	**niScope_ClearWaveformMeasurementStats** Clears the waveform measurement statistics for the channel and measurement you specify.	ViSession ViConstString ViInt32	vi channelName measFunction	Session handle Channel to clear statistics from Any measurement; Use NISCOPE_VAL_ALL_MEASUREMENTS to clear all measurements
	**niScope_ConfigureRefLevels** Sets the corresponding NISCOPE_ATTR_MEAS_CHAN_*XXX*_REF_LEVEL attributes for each channel. Attributes can be set differently for each channel by calling **niScope_SetAttributeViInt32** or by using a LabVIEW property node.	ViSession ViReal64 ViReal64 ViReal64	vi low medium high	Session handle Low reference level; see *NI-SCOPE Function Reference Help* for level usage Mid reference level High reference level
	**niScope_ReadWaveformMeasurement** Calls **niScope_ReadWaveform**, and performs the scalar measurement you specify.	ViSession ViConstString ViInt32 ViInt32 ViReal64*	vi channelName measFunction maxTime measurement	Session handle Channel to acquire and read from Scalar measurement to perform; see *NI-SCOPE Function Reference Help* for constants Maximum time to wait for acquisition to finish in milliseconds Resulting measurement
	**niScope_FetchWaveformMeasurementStats** Fetches a waveform from a previously initiated acquisition and performs the specified scalar measurement. The statistics from multiple acquisitions are also returned. The statistics are updated once per acquisition if the measurement is fetched.	ViSession ViConstString ViInt32 ViInt32 ViReal64* ViReal64* ViReal64* ViReal64* ViReal64* ViInt32*	vi channelName recordNumber measFunction measurement mean stdev min max numInStats	Session handle Channel to fetch waveform from Record number to fetch; use "0" for any single record acquisition Scalar measurement to perform; see *NI-SCOPE Function Reference Help* for constants Resulting measurement Mean of last numInStats measurements Standard deviation of last numInStats measurements Minimum of last numInStats measurements Maximum of last numInStats measurements Number of measurement used for statistics

ICON	FUNCTION NAME AND DESCRIPTION[1]	TYPE	PARAMETER	VALUE TO SET, COMMENTS[2]
Waveform Measurement Functions (Continued)				
NI SCOPE ARRAY MEAS	**niScope_FetchWaveformMeasurementArray** Fetches a waveform from a previously initiated acquisition and performs the specified array measurement.	ViSession	vi	Session handle
		ViConstString	channelName	Channel to fetch waveform from
		ViInt32	recordNumber	Record number to fetch; use "0" for any single record acquisition
		ViInt32	measFunction	Array measurement to perform; see *NI-SCOPE Function Reference Help* for constants
		ViInt32	measArraySize	Size of measArray parameter
		ViReal64	measArray[ ]	Array for the resulting measurement
		ViInt32*	actualPoints	Number of valid points available for the result (even if you do not fetch all the points)
		ViReal64*	initialX	X value of first point; see *NI-SCOPE Function Reference Help* for specific measurement units
		ViReal64*	xIncrement	Change in x values between points
NI SCOPE FETCH MEAS	**niScope_FetchMultiWaveformMeasurement** Fetches the waveform from a previously initiated acquisition and performs the scalar measurement you specify.	ViSession	vi	Session handle
		ViConstString	channelName	Channel to fetch waveform from
		ViInt32	recordNumber	Record number to fetch; use "0" for single record acquisitions
		ViInt32	measFunction	Scalar measurement to perform; see *NI-SCOPE Function Reference Help* for constants
		ViReal64*	measurement	Resulting measurement

Programming Flow