Glaz LabVIEW
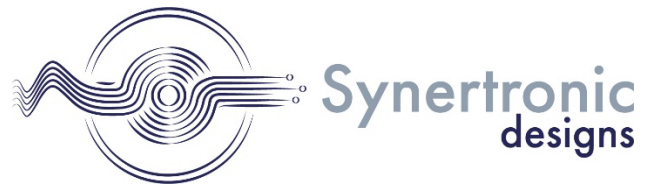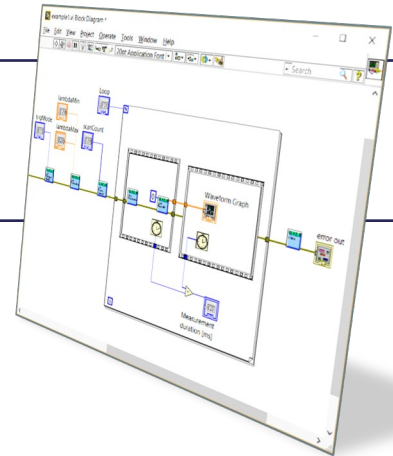# Driver 9.23

- LabVIEW VIs
- 32-bit and 64-bit libraries

Installation

Follow these steps to install the LabView driver:

1. Download *GlazLabView.zip*
2. Unzip *GlazLabView.zip* in the *user.lib* directory of your LabView installation
3. Run LabView
4. Select *Tools → Find VIs on Disk...*
5. Search the *user.lib* directory to find the Glaz LabView VIs

The VIs for interfacing with Glaz devices should now be visible in LabView's *Functions* palette.
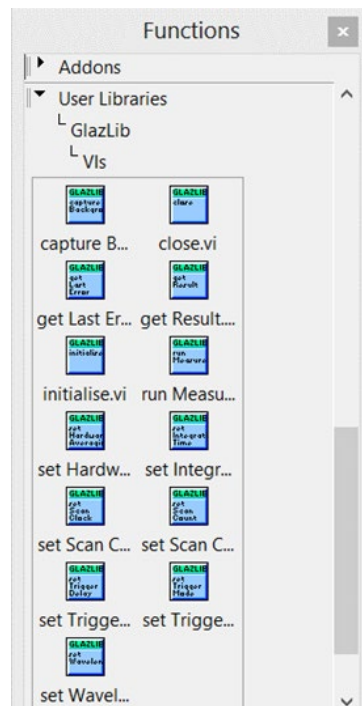
An example showing how to use the *Glaz* VIs can be found in the */user.lib/GlazLib/example* directory.

Usage notes and Error handling

The *Glaz* LabView integration consists of a set of VIs. Every VI uses the standard LabView error handling technique and each VI has an *Error in* and *Error out* parameter. The error parameters are standard LabView error structures, which provide error feedback and flow control.

The following workflow is recommended:

1. Initialise a session using either **initialise** VI or **initialise Single Device** VI.
2. Apply settings with the setter VIs.
3. Capture the background (optional, only used when background subtraction is required).
4. Run a measurement.
5. Retrieve and process results.
6. Repeat from either:
   - Step 1 and initialise with a new script file.
   - Step 2 with new settings.
   - Step 4 with the same settings.
7. Close the session by calling the **close** VI.

VIs

initialise

Initialise the session with the given **scriptFileName**. If the session was initialised before, the previous session is closed and disconnects from all previously connected devices.

Parameters:

scriptFileName	String (in)	File path of the *Glaz* script file.

Return error codes:

ERROR_NONE	No error and initialisation was successful.
ERROR_SCRIPT	The specified script was not found or contains an error.
ERROR_CONNECTING_TO_CAMERAS	There was an error while connecting to the devices specified in the script file. This can be caused by an USB communication error or the specified device was not found or is busy.
ERROR_INVALID_SETTINGS	The script contains an invalid combination of settings and devices.
ERROR_DOWNLOADING_CALIBRATIONS	There was an error while downloading the camera calibration from one of the target devices.

initialise Single Device

Initialise the session in single-device mode. If the session was initialised before, the previous session is closed and disconnects from all previously connected devices.

During single-device initialisation the *Glaz* back-end is initialised with the following script:

```
<!DOCTYPE GlazScript>"
<config>"
      <camera serial=<SN> number="1" master="1"/>
      <calculation name="Camera 1" keepscans=<KS>>
          <measurement camera="1"/>
      </calculation>
</config>
```

The serial number **<SN>** and keep-scans **<KS>** attribute are determined from the **singelDeviceType** and **keepScans** parameters.

Parameters:

singelDeviceType	Integer (in)	Specifies the type of *Glaz* LineScan camera. Must be one of the following values (as defined at the top of the header file):
		GLAZ_LINESCAN_I_PULSESYNC_S10453_SINGLE_DEVICE_TYPE 1 GLAZ_LINESCAN_I_PULSESYNC_S11639_SINGLE_DEVICE_TYPE 2 GLAZ_LINESCAN_I_TIMEFILL_S11639_SINGLE_DEVICE_TYPE 3 GLAZ_LINESCAN_I_SPECTROCAM_S11639_SINGLE_DEVICE_TYPE 4 GLAZ_LINESCAN_II_SINGLE_DEVICE_TYPE 5 GLAZ_LINESCAN_II_V2_SINGLE_DEVICE_TYPE 6 GLAZ_LINESCAN_LS_SINGLE_DEVICE_TYPE 7 GLAZ_LINESCAN_EC_SINGLE_DEVICE_TYPE 8

		Note: LineScan-I PulseSync S10453 was previously called the *Glaz-I*. LineScan-I TimeFill S11639 was previously called the *Glaz-S*.
keepScans	Boolean (in)	When set to **true** all individual scans (lines) will be stored in memory and can be accessed via the **get Scan** VI after the **run Measurement** VI was called.

Return error codes:

ERROR_NONE	No error and initialisation was successful.
ERROR_INVALID_SINGLE_DEVICE_TYPE	An invalid value was passed for **singelDeviceType**.
ERROR_INITIALISING_SINGEL_DEVICE	Unknown error while initialising the internal session.
ERROR_CONNECTING_TO_CAMERAS	There was an error while connecting to the devices specified in the script file. This can be caused by an USB communication error or the specified device was not found or is busy.
ERROR_INVALID_SETTINGS	Unknown error while initialising the internal session.
ERROR_DOWNLOADING_CALIBRATIONS	There was an error while downloading the camera calibration from one of the target devices.

close

Closes the script file and disconnects from all cameras.

⚠️ Always execute this VI at the end of the LabView program.

reset All Devices

Resets all devices. This causes the devices to re-initialise. If a session was open, it will be automatically closed.

⚠️ Wait at least 5 seconds after calling this VI before initialising a new session.

reset All Ports

Resets all ports. This forces a power cycle on all ports and causes the devices to re-initialise. This function is recommended if a normal reset does not work. If a session was open, it will be automatically closed.

⚠️ Wait at least 5 seconds after calling this VI before initialising a new session.

set Timeout

Sets the timeout between received scans (lines). Timeout is used to prevent an infinite wait when calling the **run Measurement** VI. If the PC waits longer than the specified timeout before receiving the next scan (line), the measurement is terminated with an error.

> ⚠️ If the trigger mode is set to burst trigger, the timeout is disabled and the **run Measurement** VI will not timeout.

Parameters:

ms	Integer (in)	Timeout in [ms]. Default = 4000 ms. Set the timeout to -1 to disable the timeout.

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised.
	First call the **initialise** VI or **initialise Single Device** VI.

---

set Scan Clock Speed

Sets pixel clock scan speed. See the device manual *LineScan-I* for more information.

Restrictions:

Only supported by *LineScan-I* devices.

Parameters:

speed	Integer (in)	The clock speed. Default = SCAN_CLOCK_FULL_SPEED. Must be one of the following values: SCAN_CLOCK_FULL_SPEED 0 SCAN_CLOCK_HALF_SPEED 1

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised.
	First call the **initialise** VI or **initialise Single Device** VI.
ERROR_CLOCK_SPEED_UNSUPPORTED	Variable clock speed is not supported. It is only supported by *LineScan-I* devices.
ERROR_INVALID_SCAN_CLOCK_SPEED	An invalid clock speed was specified.

---

set Wavelengths

Sets the minimum and maximum wavelengths. It is important to set these values when using the IFFT pre-processor. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.

Parameters:

lambdaMin	Double (in)	The minimum wavelength. Default = 1.0. Validation: lambdaMin > 0.0
lambdaMax	Double (in)	The maximum wavelength. Default = 2.0. Validation: lambdaMax > 0.0 lambdaMax > lambdaMin

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised.
	First call the **initialise** VI or **initialise Single Device** VI.

`ERROR_INVALID_WAVELENGTHS`		The `lambdaMin` and/or `lambdaMax` parameters failed validation.

set Hardware Averaging

Sets the number of scans to use during hardware averaging. This value is also equal to the number of hardware averaged scans $N_{a,HW}$. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Device	Maximum hardware averaging
LineScan-I	1024 (1024-pixel sensors) 256 (2048-pixel sensors)
LineScan-II	256
LineScan-I-Gen2	256
LineScan-NMOS *LineScan-LS* *LineScan-EC*	1

Parameters:

averaging	Integer (in)	The hardware averaging level. Default = `AVERAGING_X1`. Must be one of the following values:
		`AVERAGING_X1      0`
		`AVERAGING_X2      1`
		`AVERAGING_X4      2`
		`AVERAGING_X8      3`
		`AVERAGING_X16     4`
		`AVERAGING_X32     5`
		`AVERAGING_X64     6`
		`AVERAGING_X128    7`
		`AVERAGING_X256    8`
		`AVERAGING_X512    9`
		`AVERAGING_X1024  10`

Return error codes:

`ERROR_NONE`		No error and settings were applied.
`ERROR_NOT_INITIALISED`		The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
`ERROR_INVALID_AVERAGING`		The specified hardware averaging level is invalid or not supported.

set Scan Count

Sets the number of hardware average scans to perform. This value is also equal to the number of software averaged scans $N_{a,SW}$. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.

Parameters:

scanCount	Integer (in)	The number of scans (lines) to be measured during one measurement run. Default = 1. Validation: `scanCount >  0`

		scanCount <= 4000000 (LineScan-I-Gen2, 4.0 or higher) or scanCount <= 50000 (all other devices)

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_SCAN_COUNT	The scanCount parameter failed validation.

set Trigger Mode

Sets the trigger mode. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II*.

Restrictions:

Device	External	Internal	Burst
LineScan-I	Yes	Yes	No
LineScan-II	Yes	Yes	firmware version 6.0 or higher
LineScan-I-Gen2	Yes	Yes	Yes
LineScan-NMOS	No	Yes	Yes
LineScan-LS *LineScan-EC*	Yes	Yes	No

Parameters:

mode	Integer (in)	The hardware averaging level. Default = TRIGGER_EXTERNAL. Must be one of the following values:
		TRIGGER_EXTERNAL 0
		TRIGGER_INTERNAL 1
		TRIGGER_BURST 2

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_TRIGGER_MODE	The specified trigger mode is invalid or not supported.

set Trigger Delay

Sets the trigger delay in μs. This value is only used in external trigger mode. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Not supported by *LineScan-NMOS* devices.

Parameters:

us	Double (in)	The trigger delay. Default = 0 us. Validation: us >= 0 us <= 100000

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_TRIGGER_DELAY	The **us** parameter failed validation.

---

set Internal Trigger Frequency

This sets the internal trigger frequency. This trigger frequency is used when the trigger mode is set to *internal trigger.*

Restrictions:

Not supported by *LineScan-I* devices with *PulseSync* firmware.

Device	Internal trigger frequency range [Hz]
LineScan-I	2.4 .. 4000
LineScan-II *LineScan-I-Gen2*	2.4 .. 9000 (S10453) 2.4 .. 18000 (S12198-512Q) 2.4 .. 9000 (S12198-1024Q) 2.4 .. 2300 (S13496) 2.4 .. 18000 (G11620-256DA) 2.4 .. 18000 (G11620-512DA) 2.4 .. 4600 (S11639-01)
LineScan-NMOS	0.5 .. 483
LineScan-LS	0.5 .. 100
LineScan-EC	0.5 .. 50

Parameters:

Hz	Double (in)	The internal trigger frequency. Default = 1000 Hz.

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_TRIGGER_FREQUENCY	The specified trigger frequency falls outside the valid range.

---

set Integration Mode

Glaz LineScan-I devices are pre-programmed with a specific integration mode (PulseSync or TimeFill) and the integration mode cannot be changed at run-time. *Glaz LineScan-II* devices support dynamic integration modes and the integration mode can be changed at run-time. This function is only used for *LineScan-II* devices. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Device	Supported modes
LineScan-I	Depends on firmware flavour.
LineScan-II *LineScan-I-Gen2*	`INT_MODE_PULSESYNC` `INT_MODE_TIMEFILL` (CMOS sensors only)
LineScan-NMOS *LineScan-LS* *LineScan-EC*	`INT_MODE_TIMEFILL`

Parameters:

`mode`	`Integer (in)`	The integration mode. Must be one of the following: `INT_MODE_PULSESYNC    0` `INT_MODE_TIMEFILL     1`

Return error codes:

`ERROR_NONE` — No error and settings were applied.

`ERROR_NOT_INITIALISED` — The session was not initialised.
First call the **initialise** VI or **initialise Single Device** VI.

`ERROR_INVALID_INTEGRATION_MODE` — The specified integration mode is invalid or not supported.

---

set Integration Time

Sets the integration time in μs. See the device manual *LineScan-I*, *LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Not supported by *LineScan-NMOS* devices.

Device	Integration time range
LineScan-I	2 μs .. 400 ms
LineScan-II	2 μs .. 400 ms
LineScan-I-Gen2	2 μs .. 1.6 s
LineScan-LS	30 μs .. 2.0 s
LineScan-EC	40 μs .. 250 ms

Parameters:

`us`	`Double (in)`	The integration time. Default = 10 us.

Return error codes:

`ERROR_NONE` — No error and settings were applied.

`ERROR_NOT_INITIALISED` — The session was not initialised.
First call the **initialise** VI or **initialise Single Device** VI.

`ERROR_INTEGRATION_TIME_NOT_SUPPORTED`
Integration time is not supported. Integration time is controlled by the internal trigger frequency.

`ERROR_INVALID_INTEGRATION_TIME` — The specified integration time falls outside the valid range.

set Resolution

Sets the resolution in bits. See the device manual *LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

Parameters:

resolution	Integer (in)	The resolution. Default = `RESOLUTION_16BIT`. Must be one of the following values:
		`RESOLUTION_16BIT    3`
		`RESOLUTION_14BIT    2`
		`RESOLUTION_12BIT    1`
		`RESOLUTION_10BIT    0`

Return error codes:

`ERROR_NONE`   No error and settings were applied.

`ERROR_NOT_INITIALISED`   The session was not initialised.
First call the **initialise** VI or **initialise Single Device** VI.

`ERROR_RESOLUTION_OUT_OF_RANGE`   The specified resolution is invalid.

`ERROR_RESOLUTION_NOT_SUPPORTED` Setting the resolution is not supported by the connected device.

set Sync Out Mode

Sets the output mode of the *Sync* port. The supported modes are device-dependent. For devices in PulseSync mode, the *Sync* output mode is automatically forced to **OUT_BUSY**. See the device manual *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Device	Supported modes
LineScan-I	Depends on firmware flavour.

LineScan-II
LineScan-I-Gen2
	`OUT_INT_WINDOW`
	`OUT_TRIGGER`
	`OUT_BUSY`
	`OUT_TRIGGER_CYCLE_START`
	`OUT_TRIGGER_CYCLE_RUNNING`

LineScan-NMOS *LineScan-LS* *LineScan-EC*	`OUT_INT_WINDOW` `OUT_BUSY`

Parameters:

mode	Integer (in)	The output mode. Default = `OUT_INT_WINDOW`. Must be one of the following values:
		`OUT_INT_WINDOW           0`
		`OUT_TRIGGER              1`
		`OUT_BUSY                 2`
		`OUT_TRIGGER_CYCLE_START  3`
		`OUT_TRIGGER_CYCLE_RUNNING 4`
		`OUT_OFF                  5`

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_SYNC_OUT_MODE	The specified output mode is invalid or not supported.

set Sync Out Polarity

Sets the output polarity of the *Sync* port. This functionality is only supported by *LineScan-II* devices. For devices in PulseSync mode, the *Sync* output polarity is automatically forced to OUT_POLARITY_ACTIVE_LO. See the device manual *LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

Parameters:

polarity	Integer (in)	The output polarity. Default = OUT_POLARITY_ACTIVE_LO. Must be one of the following values: OUT_POLARITY_ACTIVE_HI 1 OUT_POLARITY_ACTIVE_LO 0

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_OUT_POLARITY_NOT_SUPPORTED	Polarity settings are not supported by the connected device.
ERROR_INVALID_OUT_POLARITY	The specified polarity is not one of the values listed above.

set Aux Out Mode

Sets the output mode of the *Aux* port. This functionality is only supported by *LineScan-II* devices. See the device manual *LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

Parameters:

mode	Integer (in)	The output mode. Default = OUT_INT_WINDOW. Must be one of the following values: OUT_INT_WINDOW 0 OUT_TRIGGER 1 OUT_BUSY 2 OUT_TRIGGER_CYCLE_START 3 OUT_TRIGGER_CYCLE_RUNNING 4 OUT_OFF *(input)* 5

Return error codes:

ERROR_NONE		No error and settings were applied.
ERROR_NOT_INITIALISED		The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_AUX_OUT_MODE		The specified output mode is invalid or not supported.

set Aux Out Polarity

Sets the output polarity of the *Aux* port. This functionality is only supported by *LineScan-II* devices. See the device manual *LineScan-I-Gen2* or *LineScan-II* for more information.

Restrictions:

Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

Parameters:

polarity	Integer (in)	The output polarity. Default = OUT_POLARITY_ACTIVE_LO. Must be one of the following values:
		    OUT_POLARITY_ACTIVE_HI 1
		    OUT_POLARITY_ACTIVE_LO 0

Return error codes:

ERROR_NONE	No error and settings were applied.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_OUT_POLARITY_NOT_SUPPORTED	Polarity settings are not supported by the connected device.
ERROR_INVALID_OUT_POLARITY	The specified polarity is not one of the values listed above.

capture Background

Captures the camera background. Subsequent measurements using the background subtract pre-processor, will use the last captured background. If no background was captured, the background defaults to zero.

Parameters:

scanCount	Integer (in)	The number of scans (lines) that will be measured and averaged to capture the background.

Return error codes:

ERROR_NONE	Backgrounds were captured successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_CAPTURING_BACKGROUNDS	There was a communication error while receiving data from the connected *LineScan* devices. Check the USB cable connections.

run Measurement

Starts a measurement run. The connected devices will perform a measurement with the previously specified settings. If settings were not specified, the default values are used. This function will only return, when the measurement run is completed, a time-out was encountered or an error was detected. A measurement run is completed after all `scanCount` number of scans (lines) were captured by the *Glaz LineScan* devices, the data was received via USB and processed by the back-end.

Return error codes:

ERROR_NONE	No error and measurement was run successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_SETTINGS	An invalid combination of settings were specified.
ERROR_RUNNING_MEASUREMENT	There was a communication error while receiving data from the connected *LineScan* devices. Check the USB cable connections.

start Measurement

Starts a measurement run. The connected devices will perform a measurement with the previously specified settings. If settings were not specified, the default values are used. This function returns immediately. Use the **is Measurement Done** VI to poll until the measurement is done.

Return error codes:

ERROR_NONE	No error and measurement was run successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_SETTINGS	An invalid combination of settings was specified.
ERROR_RUNNING_MEASUREMENT	There was a communication error while receiving data from the connected *LineScan* devices. Check the USB cable connections.

is Measurement Done

Call this VI after a measurement was started with the **start Measurement** VI. The VI will return *true* if the measurement is done or an error was encountered.

Return error codes:

ERROR_NONE	No error and measurement was run successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_RUNNING_MEASUREMENT	There was a communication error while receiving data from the connected *LineScan* devices. Check the USB cable connections.

get Result

Returns a vector with N_{pixel} values containing the software-averaged calculation with the given index. The calculation indices start at zero.

Parameters:

calculationIndex	Integer (in)	The index of the calculation specified in the *Glaz* script file. The index is zero-based and depends on the order of calculations defined in the script file. The first calculation in the script file will have **calculationIndex = 0**, the second calculation will have **calculationIndex = 1** and so on. When the session was initialised with the **initialise Single Device VI**, the only valid value is **calculationIndex = 0.**
values	Double array (out)	Array (vector) with N_{pixel} values containing the averaged data for the calculation with the given index *calculationIndex*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CALCULATION_INDEX	The **calculationIndex** is out of range. Check the script file and determine the correct index.
ERROR_INVALID_RESULT_DATA_SIZE	The array size of the result array **values** does not match the number of pixels. This is most likely caused if a gated calculation was defined, but the calculation was never triggered. Check the gating attributes in the script file and the trigger level of the *Glaz-PD* device.
ERROR_NO_MEASUREMENT_RUN	The **run Measurement** VI was not called and there are no results available.

get Complex Result

Returns two vectors with each N_{pixel} values containing the real and imaginary components of the software-averaged calculation with the given index. This VI is recommended for applications using inverse Fourier transforms. The calculation indices start at zero.

Parameters:

calculationIndex	Integer (in)	The index of the calculation specified in the *Glaz* script file. The index is zero-based and depends on the order of calculations defined in the script file. The first calculation in the script file will have **calculationIndex = 0**, the second calculation will have **calculationIndex = 1** and so on. When the session was initialised with the **initialise Single Device VI**, the only valid value is **calculationIndex = 0.**
real	Double array (out)	Array (vector) with N_{pixel} values containing the real component of the averaged data for the calculation with the given index *calculationIndex*.
imag	Double array (out)	Array (vector) with N_{pixel} values containing the imaginary component of the averaged data for the calculation with the given index *calculationIndex*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CALCULATION_INDEX	The `calculationIndex` is out of range. Check the script file and determine the correct index.
ERROR_INVALID_RESULT_DATA_SIZE	The array size of the result array `values` does not match the number of pixels. This is most likely caused if a gated calculation was defined, but the calculation was never triggered. Check the gating attributes in the script file and the trigger level of the *Glaz-PD* device.
ERROR_NO_MEASUREMENT_RUN	The **run Measurement** VI was not called and there are no results available.

get Scan

Returns a vector with N_{pixel} values containing the calculation with the given index for the given scan index. The calculation and scan indices start at zero. In order to obtain individual scans the `keepscans` parameter in the calculation start definition must be enabled.

Parameters:

calculationIndex	Integer (in)	The index of the calculation specified in the *Glaz* script file. The index is zero-based and depends on the order of calculations defined in the script file. The first calculation in the script file will have `calculationIndex = 0`, the second calculation will have `calculationIndex = 1` and so on. When the session was initialised with the **initialise Single Device VI**, the only valid value is `calculationIndex = 0.`

scanIndex	Integer (in)	The index of the scan. Validation:
		    `scanIndex >= 0`
		    `scanIndex < scanCount`
		Where `scanCount` is the parameter that was passed to the **set Scan Count** VI.

values	Double array (out)	Array (vector) with N_{pixel} values containing the individual calculation with the given index *calculationIndex* for the given index *scanIndex*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CALCULATION_INDEX	Either the `calculationIndex` or `scanIndex` is out of range. Check the script file and determine the correct calculation index. Also check the `scanCount` parameter that was passed to the **set Scan Count** VI.
ERROR_INVALID_RESULT_DATA_SIZE	The array size of the result array `values` does not match the number of pixels. This is most likely caused if a gated calculation was defined, but the calculation was never triggered. Check the gating attributes in the script file and the trigger level of the *Glaz-PD* device.
ERROR_NO_MEASUREMENT_RUN	The **run Measurement** VI was not called and there are no results available.

get Complex Scan

Returns two vectors with each N_{pixel} values containing the real and imaginary components of the calculation with the given index for the given scan index. This VI is recommended for applications using inverse Fourier transforms. The calculation and scan indices start at zero. In order to obtain individual scans the `keepscans` parameter in the calculation start definition must be enabled.

Parameters:

calculationIndex	Integer (in)	The index of the calculation specified in the *Glaz* script file. The index is zero-based and depends on the order of calculations defined in the script file. The first calculation in the script file will have `calculationIndex = 0`, the second calculation will have `calculationIndex = 1` and so on. When the session was initialised with the **initialise Single Device VI**, the only valid value is `calculationIndex = 0.`
scanIndex	Integer (in)	The index of the scan. Validation: `scanIndex >= 0` `scanIndex < scanCount` Where `scanCount` is the parameter that was passed to the **set Scan Count** VI.
real	Double array (out)	Array (vector) with N_{pixel} values containing the real component of the individual calculation with the given index *calculationIndex* for the given index *scanIndex*.
imag	Double array (out)	Array (vector) with N_{pixel} values containing the imaginary component of the individual calculation with the given index *calculationIndex* for the given index *scanIndex*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CALCULATION_INDEX	Either the `calculationIndex` or `scanIndex` is out of range. Check the script file and determine the correct calculation index. Also check the `scanCount` parameter that was passed to the **set Scan Count** VI.
ERROR_INVALID_RESULT_DATA_SIZE	The array size of the result array **values** does not match the number of pixels. This is most likely caused if a gated calculation was defined, but the calculation was never triggered. Check the gating attributes in the script file and the trigger level of the *Glaz-PD* device.
ERROR_NO_MEASUREMENT_RUN	The **run Measurement** VI was not called and there are no results available.

get All Scans

Returns a matrix containing all scans for the calculation with the given index. The calculation index starts at zero. The `keepscans` parameter in the *Glaz* script file must be enabled.

Parameters:

calculationIndex	Integer (in)	The index of the calculation specified in the *Glaz* script file. The index is zero-based and depends on the order of calculations defined in the script file. The first calculation in the script file will have `calculationIndex = 0`, the second calculation will have `calculationIndex = 1` and so on.

		When the session was initialised with the **initialise Single Device VI**, the only valid value is `calculationIndex = 0.`
`values`	Unsigned short 2D matrix (out)	2D Matrix containing all scans for the calculation with the given *calculationIndex*.

Return error codes:

ERROR_NONE   No error and values were returned successfully.

ERROR_NOT_INITIALISED   The session was not initialised.
                                      First call the **initialise** VI or **initialise Single Device** VI.

ERROR_INVALID_CALCULATION_INDEX   The `calculationIndex` is out of range. Check the script file and determine the correct index.

ERROR_INVALID_RESULT_DATA_SIZE   The sub-array size of the result array **values** does not match the number of pixels. This is most likely caused if a gated calculation was defined, but the calculation was never triggered. Check the gating attributes in the script file and the trigger level of the *Glaz-PD* device.

ERROR_NO_MEASUREMENT_RUN   The **run Measurement** VI was not called and there are no results available.

write All Scans To File

Writes all scans for the calculation with the given index to a binary file. The binary file is written in big-endian format and has the following structure if **writeTimestamps** is *false*:

```
uint16          number of scans, Ns
uint16          number of pixels, Np
Np x uint16     1. scan
Np x uint16     2. scan
…
Np x uint16     Ns. scan
```

The binary file has the following structure if **writeTimestamps** is *true*:

```
4 x uint8       preamble consisting of 4 bytes: 0x00, 0x00, 0xA5, 0xC3
uint8           version: 0x01
uint16          number of scans, Ns
uint16          number of pixels, Np
uint32          timestamp for 1. scan
Np x uint16     1. scan
uint32          timestamp for 2. scan
Np x uint16     2. scan
…
uint32          timestamp for Ns. scan
Np x uint16     Ns. scan
```

The timestamp value can be converted to [μs] by multiplying it with the following factor:

Model	Conversion factor to [μs]
LineScan-I	0.1 (half speed) 0.05 (full speed)
LineScan-I-Gen2	0.2
LineScan-II	0.2
LineScan-NMOS	0.25
LineScan-LS	0.25
LineScan-EC	0.25

⚠ This VI must be called before running a measurement. The scans are written to the target file while the measurement is performed.

Parameters:

calculationIndex	Integer (in)	The index of the calculation specified in the *Glaz* script file. The index is zero-based and depends on the order of calculations defined in the script file. The first calculation in the script file will have **calculationIndex = 0**, the second calculation will have **calculationIndex = 1** and so on. When the session was initialised with the **initialise Single Device VI**, the only valid value is **calculationIndex = 0.**
filename	String (in)	File path of the target data file.
writeTimestamps	Boolean (in)	When set to *true* a 32-bit long timestamp is written before each scan.

Return error codes:

ERROR_NONE — No error and settings were applied.

ERROR_NOT_INITIALISED — The session was not initialised.
First call the **initialise** VI or **initialise Single Device** VI.

ERROR_INVALID_CALCULATION_INDEX — The **calculationIndex** is out of range. Check the script file and determine the correct index.

get PD Reference

Returns the reference value used to normalise calculations. See the device manuals *LineScan-I*, *LineScan-I-Gen2* or *LineScan-II* for more information. After the **run Measurement** VI is called, this will be the first valid received value from a Glaz-PD device.

Parameters:

pdNumber	Integer (in)	The number specified for a Glaz-PD in the script file. See the device manuals *LineScan-I* or *LineScan-II* for more information.
pdChannel	Integer (in)	One of the following values: 1 Channel 1 2 Channel 2
value	Double (out)	The value used as reference for the specified channel and the Glaz-PD defined by *pdNumber*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_PD_NUMBER	The **pdNumber** is out of range. Check the script file and determine the correct device number.
ERROR_INVALID_PD_CHANNEL	The **pdChannel** is out of range or the specified channel is not enabled in the script file.

get PD Values

Returns all the measured values for a given Glaz-PD device and channel. Values which are invalid (i.e. the Glaz-PD device was not triggered) are shown as zero.

Parameters:

pdNumber	Integer (in)	The number specified for a Glaz-PD in the script file. See the device manuals *LineScan-I, LineScan-I-Gen2* or *LineScan-II* for more information.
pdChannel	Integer (in)	One of the following values: 1 Channel 1 2 Channel 2
value	Double (out)	Array (vector) with the measured values. The array is of length *scanCount*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_PD_NUMBER	The **pdNumber** is out of range. Check the script file and determine the correct device number.
ERROR_INVALID_PD_CHANNEL	The **pdChannel** is out of range or the specified channel is not enabled in the script file.

get AUX States

Returns all the measured *Aux* port states for a given LineScan device.

Restrictions:

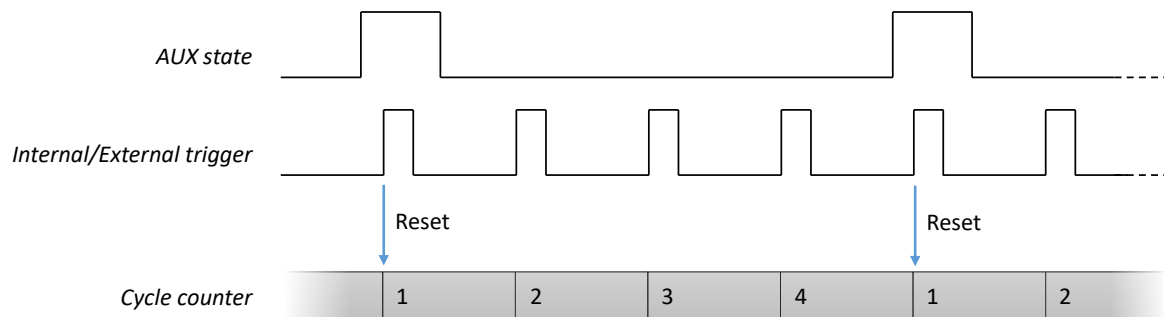Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

Parameters:

cameraNumber	Integer (in)	The number specified for a LineScan device in the script file. See the device manuals *LineScan-I-Gen2* or *LineScan-II* for more information.
value	Integer (out)	Array (vector) with the measured *Aux* port states. A "1" corresponds to a high state. A "0" corresponds to a low state. The array is of length *scanCount*.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CAMERA_NUMBER	The **cameraNumber** is out of range. Check the script file and determine the correct device number.
ERROR_AUX_STATES_NOT_SUPPORTED	AUX states are not supported. Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

get AUX Cycle Counts

Uses the *Aux* port states for a given LineScan device to perform a cycle count. With each trigger the cycle count is incremented. If the *AUX* state is high when triggered, the cycle count is reset to 1.

⚠️ The *AUX* port must be configured as an input when measuring external *AUX* port signals. See "set Aux Out Mode".

Restrictions:

Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.

Parameters:

cameraNumber	Integer (in)	The number specified for a LineScan device in the script file. See the device manuals *LineScan-I-Gen2* or *LineScan-II* for more information.
maxCount	Integer (in)	The maximum expected cycle count.
values	Integer (out)	Array (vector) with the measured *Aux* port cycle counts.

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CAMERA_NUMBER	The **cameraNumber** is out of range. Check the script file and determine the correct device number.

ERROR_AUX_STATES_NOT_SUPPORTED	AUX states are not supported. Only supported by *LineScan-II* and *LineScan-I-Gen2* devices.
ERROR_AUX_CYCLE_COUNT_INVALID	The cycle count exceeds the specified `maxCount`.

---

get Time Stamp

Returns the timestamp for a given camera number and scan index. The timestamp is given in [μs].

Parameters:

`cameraNumber`	`Integer (in)`	The number specified for a LineScan device in the *Glaz* script file.

`scanIndex`	`Integer (in)`	The index of the scan. Validation:
		`scanIndex  >=  0`
		`scanIndex  <   scanCount`
		Where `scanCount` is the parameter that was passed to the **set Scan Count** VI.

`value`	`Double (out)`	The time-stamp value in [μs].

Return error codes:

ERROR_NONE	No error and values were returned successfully.
ERROR_NOT_INITIALISED	The session was not initialised. First call the **initialise** VI or **initialise Single Device** VI.
ERROR_INVALID_CAMERA_NUMBER	The `cameraNumber` is out of range. Check the script file and determine the correct device number.

Error codes

Code	Description	Reason
0	ERROR_NONE	No error
1	ERROR_NOT_INITIALISED	The *initialise Session VI* was not executed. First execute the *initialise Session VI* with the correct script file, before executing any other *Glaz* VI.
2	ERROR_SCRIPT	There is an error in the script. The error output of the VI contains more information about the error.
3	ERROR_CONNECTING_TO_CAMERAS	There is a problem connecting with one or more cameras. Make sure that all cameras are connected and not used by another application. Check that the camera definition section in the script file contains the correct cameras.
4	ERROR_DOWNLOADING_CALIBRATIONS	Communication error while trying to download the camera calibration. Check the USB cable connection.
5	ERROR_INVALID_WAVELENGTHS	Incorrect wavelength parameters. Check that *lambdaMin* < *lambdaMax* and that both *lambdaMin* and *lambdaMax* are larger than zero.
6	ERROR_INVALID_AVERAGING	`averaging` parameter is out of range.
7	ERROR_INVALID_SCAN_COUNT	`scanCount` parameter is out of range.
8	ERROR_INVALID_TRIGGER_MODE	Trigger `mode` parameter is out of range.
9	ERROR_INVALID_TRIGGER_DELAY	Trigger delay `us` parameter is out of range.
10	ERROR_INVALID_INTEGRATION_TIME	Integration time `us` parameter is out of range.
11	ERROR_INVALID_SCAN_CLOCK_SPEED	Incorrect or unsupported scan clock `speed` parameter.
12	ERROR_INVALID_SETTINGS	There are some invalid settings. The error output of the VI contains more information about the error.
13	ERROR_CAPTURING_BACKGROUNDS	Communication error while trying to capture the camera background. Check the USB cable connection.
14	ERROR_RUNNING_MEASUREMENT	Communication error while trying to take a measurement. Check the USB cable connection.
15	ERROR_INVALID_CALCULATION_INDEX	Incorrect calculation index was specified. The calculation index starts at zero and must be smaller than the number of calculations specified in the script file.
16	ERROR_INVALID_RESULT_DATA_SIZE	The size of the result data is not N_{pixel}. Check that the calculation was specified correctly in the script file.
17	ERROR_INVALID_PD_NUMBER	*Glaz-PD* device number does not match any *Glaz-PD* device number specified in the *Glaz* script file.
18	ERROR_INVALID_PD_CHANNEL	*Glaz-PD* channel number may only be 1 or 2.
19	ERROR_INVALID_CAMERA_NUMBER	Camera number does match any camera number specified in the *Glaz* script file.
20	ERROR_INVALID_TRIGGER_FREQUENCY	Trigger frequency `Hz` parameter is out of range or trigger frequency function is not supported.
21	ERROR_NO_MEASUREMENT_RUN	Trying to retrieve results before a measurement was run.
22	ERROR_INITIALISING_SINGEL_DEVICE	Unable to initialise single-device mode.
23	ERROR_INVALID_SINGLE_DEVICE_TYPE	An invalid or unsupported single device type was specified.
24	ERROR_INVALID_SYNC_OUT_MODE	Invalid or unsupported *Sync* out mode.
25	ERROR_INVALID_INTEGRATION_MODE	Invalid or unsupported integration mode.
26	ERROR_CLOCK_SPEED_UNSUPPORTED	Variable pixel clock speed is not supported.
27	ERROR_INVALID_AUX_OUT_MODE	Invalid or unsupported *Aux* out mode.

Code	Description	Reason
28	`ERROR_CYCLE_COUNT_UNSUPPORTED`	Cycle counting is not supported.
29	`ERROR_INVALID_CYCLE_COUNT`	Cycle count parameter is out of range.
30	`ERROR_INVALID_TEST_MODE`	Invalid or unsupported test mode
31	`ERROR_OUT_POLARITY_NOT_SUPPORTED`	Configurable port polarity is not supported.
32	`ERROR_INVALID_OUT_POLARITY`	Invalid port polarity.
33	`ERROR_RESOLUTION_OUT_OF_RANGE`	The specified is not valid.
34	`ERROR_RESOLUTION_NOT_SUPPORTED`	Setting the resolution is not supported.
35	`ERROR_RUNNING_USB_COMMS_TEST`	Error encountered during USB communication test.
36	`ERROR_MEASUREMENT_STREAM`	A stream error was detected. Run the measurement again.
37	`ERROR_AUX_STATES_NOT_SUPPORTED`	The connected LineScan device does not have an *Aux* port. Recording of *Aux* states is not supported.
38	`ERROR_INTEGRATION_TIME_NOT_SUPPORTED`	Integration time not supported by the connected device.
39	`ERROR_INVALID_ADC_GAIN`	Invalid/Unsupported ADC gain.
40	`ERROR_AUX_CYCLE_COUNT_INVALID`	The maximum cycle count exceeds the value specified by maxCount.

Example guide

Inside the *Glaz* LabVIEW driver folder is a folder called *examples*. This folder contains an example VIs to demonstrate the basic driver usage.
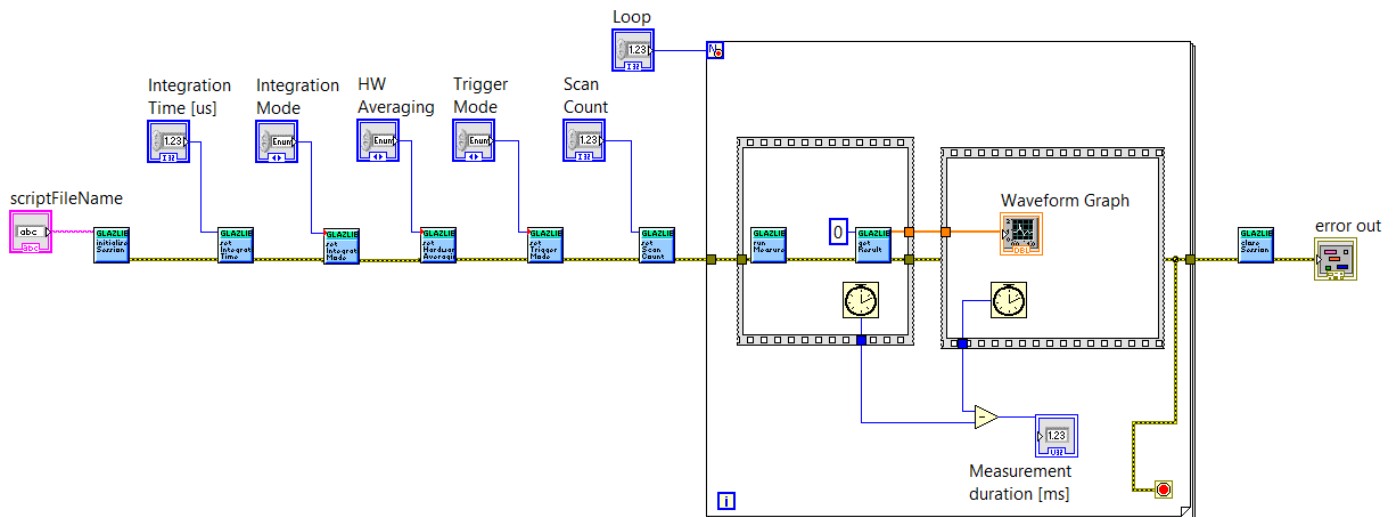
Example 1

Connects to a single camera. The example script file, *example1.gsc*, is provided also contained in the *example* folder.

Setup:

1. Connect a *LineScan* camera to your PC.
2. Edit the example script file:
    a. Open *example1.gsc* in any text editor.
    b. Replace the XXXXX on line 3 with the serial number of the *LineScan* device.
    c. Save the file and exit the text editor.

Program flow:

1. **initialise**
   Initialise the session with a script file. This VI must always be called first before any other *Glaz* VI.

   > ⚠ This VI must always be called first before any other *Glaz* VI.

2. **set Integration Time**
   Set the integration time in [μs].

3. **set Integration Mode**
   Set the integration mode to *PulseSync* or *TimeFill*. For *LineScan-I* devices this VI is not required. *LineScan-I* are pre-programmed with firmware supporting either *PulseSync* or *TimeFill* integration modes.

> ⚠️ For *LineScan-II* devices the integration mode must be specified. Make sure which integration mode to use. Incorrect integration modes will result in unexpected behaviour. Especially, in multi-camera setup.

4. **`set HW Averaging`**

   This VI is not required and by default no HW averaging is used.

5. **`set Trigger Mode`**

   Set the correct trigger mode. Using the external trigger mode without an external trigger will cause a time-out when calling the **`run Measurement`** VI.

6. **`set Scan Count`**

   Set the number of scans (lines) to be captured during the measurement run.

7. **`Loop`**

   This control demonstrates how to handle loops. Set the number of times the measurement must be repeated.

8. **`run Measurement`**

   Runs the measurement. The LabVIEW user interface will be unresponsive while this VI executes.

9. **`get Result`**

   Retrieve the result and plot it.

10. **`close`**

    Close the session. This VI also disconnects from all *Glaz* devices.

> ⚠️ This VI must always be called at the end of your program. Failing to do so, will keep LabVIEW connected to the *Glaz* devices. These devices will then be unavailable to any other application.

IMPORTANT NOTICE

Synertronic Designs reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to Synertronic Designs' terms and conditions of sale supplied at the time of order acknowledgment.

Synertronic Designs assumes no liability for applications assistance or customer product design. Customers are responsible for their applications using Synertronic Designs products. To minimize the risks associated with customer applications, customers should provide adequate operating safeguards.

Reproduction of information in Synertronic Designs data sheets, summary notes and brochures is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. Synertronic Designs is not responsible or liable for such altered documentation.

Synertronic Designs on the web: www.synertronic.co.za

E-mail: info@synertronic.co.za

Postal address: Kaneel Cr 34
Stellenbosch
7600
South Africa