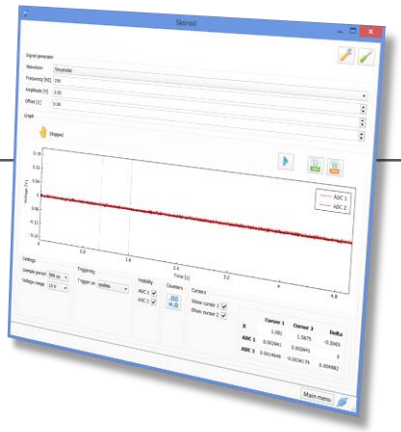
Synertronic Designs



Skorost-UI

Configurable user interface

Features

- Script-based configuration
- One-click measurements
- Data export
- Auto-connect

Applications

- Data logging
- Educational experiments

Overview

Skorost UI (a stand-alone application) is distributed as freeware. It provides a configurable user interface to communicate with all *Skorost* devices. Unlike other data logger user interfaces, *Skorost UI* can be configured via XML-based script files. With these scripts it is possible to select which user interface components to display and how measurements should be taken. For most cases, the user interface can configured to display a single button to run the experiment. This is especially useful for educational applications. Students don't need to learn a complicated user interface and can focus on performing their experiments.

Installing Skorost UI

The installer for *Skorost UI* can be downloaded from the Synertronic Designs web page (installer link). Download and run the installer.

If the target PC is not connected to the internet, it is advisable to pre-install the USB device driver. The USB device driver can also be downloaded from the Synertronic Designs web page (driver link).

Launching Skorost UI

The installer will create a desktop shortcut. Double this shortcut to launch the application.

Figure 1 Skorost UI shortcut icon.

Start-up page

When the application is launched the start-up page is shown. The application will search for any connected *Skorost* devices. When a device is found, a list of available script configuration files will be given. Click any one of these entries in the list to display the user interface defined by the script file.

Skorost UI will search all paths, as specified by the global application settings, for script configuration files. See "Global settings (development mode only)" for more information and how to specify search paths.

Script files must have the extension *.ssc*. The application will only list script files on the start-up page, which are compatible with the currently connected *Skorost* device. Script files with syntax errors will not be displayed.

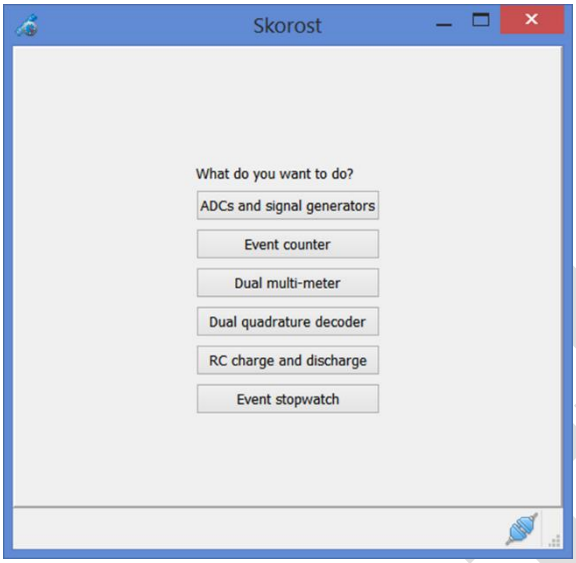


Figure 2 Start-up page.

Launching in development mode

In order to enable advanced features, *Skorost UI* must be launched in development mode. Follow these steps to launch in development mode:

1. Launch a windows terminal (Windows XP, Windows 7 and Windows 8)
    i. Press the *Windows* button on your keyboard or click the *Start* button
    ii. Simply type “cmd” and press *Enter*
2. Wait for the terminal window to open and then navigate to the *Skorost UI* program folder. This is done by using the *cd* command. For example: cd "c:\Program Files (x86)\Synertronic\Skorost"
3. Type the following to launch *Skorost UI* in development mode: Skorost –dev

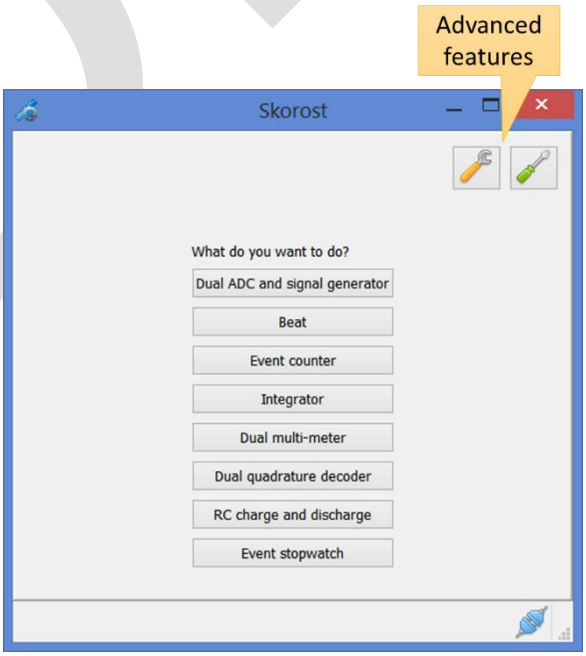


Figure 3 Start-up page in development mode.

The start-up will now provide some additional features:


Edit global settings.


Perform calibration (outside the scope of this document).

Global settings (development mode only)

Edit the global settings by clicking the  button on the start-up page. This opens the global settings dialog.

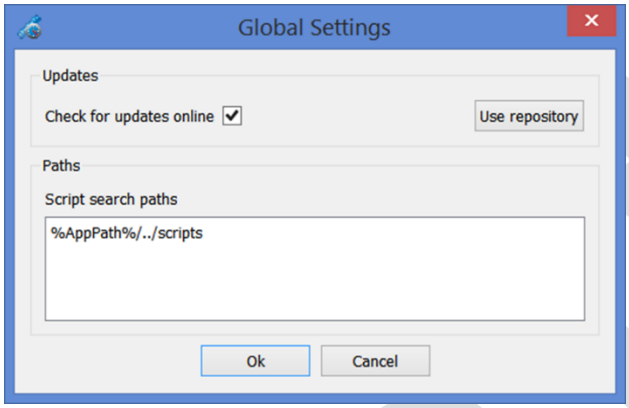


Figure 4 Global settings dialog.

In the *Updates* group, *Check for updates online* can be enabled or disabled. When enabled, *Glaz UI* will try to connect to the online update repository and download any new software and firmware. For more information on updating software and firmware see “Software and firmware updates”.

In the *Paths* group the script configuration search paths can be specified. More than one path may be specified and each path must be specified on a new line. It is possible to use either “\” or “/” as directory separators. The following keywords are provided for system directories:

- %AppPath% The path were *Skorost UI* is installed.
  For example: c:\Program Files (x86)\Synertronic\Skorost
- %Documents% The document path of the current user. On Windows this is the same as the *My Documents* or *Documents* directory.

Software and firmware updates

When check for online updates is enabled, *Skorost UI* will try to download the latest software and firmware updates from the online repository of Synertronic Designs. If a new version of *Skorost UI* is available, the user will be notified. When the application closes, the application will ask if the new version must be installed.

If new firmware is available, the user will be asked to update the firmware of the connected camera. Firmware updates can take between 3 to 5 seconds.


Under no circumstances, disconnect the device during a firmware update. If a firmware update fails due to a power failure or disconnecting the camera, the device must be returned to Synertronic Designs for reprogramming.

User interface configuration scripts

Script file XML structure

The configuration script files are XML files. They have the following general structure:

```
<!DOCTYPE SkorostScript>
<config>
    List of inline and nested root elements
</config>
```

The first line in the script file (`<!DOCTYPE SkorostScript>`) and the start `<config>` and end `</config>` tags are mandatory. Between the `config` tags a list of root configuration elements are given. An element consists of a tag and a list of optional and/or mandatory attributes. There are two methods to define an element:

Method A – Inline element definition

```
<TAG ATTRIBUTE1="VALUE1" ATTRIBUTE2="VALUE2" ATTRIBUTE3="VALUE3"/>
```

Method B – Nested element definition

```
<TAG1 ATTRIBUTE1="VALUE1" ATTRIBUTE2="VALUE2" ATTRIBUTE3="VALUE3">
    <TAG2 ATTRIBUTE4="VALUE4" ATTRIBUTE5="VALUE5" ATTRIBUTE6="VALUE6">
    </TAG2>
</TAG1>
```

In method B, element TAG1 is the parent of element TAG2. Element TAG2 is also called the child of element TAG1. This method is used to fine-tune the configuration or behaviour of the parent element. In this example, TAG2 will provide additional attributes to alter the behaviour of TAG1. An inline element definition may also be written as a nested definition. In nested format, it will not have any child elements. For example:

```
<TAG ATTRIBUTE1="VALUE1" ATTRIBUTE2="VALUE2" ATTRIBUTE3="VALUE3"/>
```

May be also be written as:

```
<TAG ATTRIBUTE1="VALUE1" ATTRIBUTE2="VALUE2" ATTRIBUTE3="VALUE3">
</TAG>
```

Element attributes consists of name-value pairs. In the examples above ATTRIBUTE1, ATTRIBUTE2 and so on are the names of the attributes. The attribute name is followed by an assignment symbol “=”. After this follows the value for the attribute. Always enclose values in inverted commas. Attributes may be omitted, in which case a default value for that attribute is used. The following attribute types may be encountered:

Attribute type	Description
String	Any combination of alpha-numerical letters
Integer	An integer number
Float	A floating-point number
Flags	A bit field. The value is constructed by optionally adding the values 1, 2, 4, 8, 16, 32, 64 and 128 together
Enum	An enumeration. This is an integer value between a given lower and upper bound. Each value represents a certain state.
Bool	“1” or “true” equals TRUE, “0” or “false” equals FALSE

Root configuration elements and document structure

The configuration file document structure is shown in the table below. Attributes are omitted and only the element tags are displayed. Each element and its attributes will be described in more detail later on in this manual. Note, that some elements are mandatory. Elements can be grouped in to three types:

- User interface (UI)
- Sampling
- Control

	Required	Type	Description
<!DOCTYPE SkorostScript>	Mandatory		
<config>	Mandatory		
<titel/>	Mandatory	UI	Name of the script
<ui>	Mandatory	UI	Define user interface elements
<graph>	Optional	UI	Graph widget
<presentation/>	Optional	UI	Graph widget presentation
<calculation/>	Optional	UI	Calculations to display
</graph>			
<number>	Optional	UI	LCD-style number widget
<presentation/>	Optional	UI	LCD-style number presentation
</number>			
<signalgenerator/>	Optional	UI	Signal generator control widget
<pwmcontrol/>	Optional	UI	PWM control widget
</ui>			
<sampling>	Mandatory[1]	Sampling	Normal sampling
<prestart>	Optional	Sampling	Actions performed just before sampling starts
Any control element	Optional	Control	
</prestart>			
<poststart>	Optional	Sampling	Actions performed just after sampling started
Any control element	Optional	Control	
</poststart>			
<poststop>	Optional	Sampling	Actions performed just after sampling completed
Any control element	Optional	Control	
</poststop>			
<loop/>	Optional	Sampling	Loops a measurement
</sampling>			
<stopwatch>	Mandatory[2]	Sampling	Stopwatch sampling
<prestart>	Optional	Sampling	Actions performed just before sampling starts
Any control element	Optional	Control	
</prestart>			
<poststart>	Optional	Sampling	Actions performed just after sampling started
Any control element	Optional	Control	
</poststart>			
<poststop>	Optional	Sampling	Actions performed just after sampling completed
Any control element	Optional	Control	
</poststop>			
</stopwatch>			
<trigger/>	Optional	Sampling	Trigger settings
<pwmfrequency/>	Optional	Control	PWM frequency setting
<pwm/>	Optional	Control	PWM settings
<pwm1/>	Optional	Control	PWM settings
<pwm2/>	Optional	Control	PWM settings
<digiout/>	Optional	Control	Digital high/low output settings
<digiout1/>	Optional	Control	Digital high/low output settings
<digiout2/>	Optional	Control	Digital high/low output settings
<dac/>	Optional	Control	Signal generator settings
<dac1/>	Optional	Control	Signal generator settings
<dac2/>	Optional	Control	Signal generator settings
<clear/>	Optional	Control	Counter clear action
</config>			

[1] Mandatory if a <stopwatch> element is not defined.
[2] Mandatory if a <sampling> element is not defined.

GUI elements

The GUI XML elements are used to add user interface widgets. There are two groups of widgets:

- Data display widgets
- Control widgets

The <graph> and <number> elements define data display widgets. They display measured data, received from *Skorost* devices.

The <signalgenerator> and <pwmcontrol> elements define control widgets. Many of the graph sub-widgets are also control widgets. With these widgets the *Skorost* device settings are changed. Control widgets are used when device settings must be changed dynamically by the user. When device settings are static (i.e. do not change for a given measurement or experiment), they may also be defined by control elements in the XML configuration file. In this case, control widgets are hidden from the user.

<titel> element

Supported devices: *All*

Parent: root element

Number of elements per file: 1 (mandatory)

This element is used to define the name or description of the script file. This name will be used on the start-up page when displaying the list of script files. The following attributes are supported:

Attribute	Type	Devices	Description
name	string	All	The name or description of the script file

<ui> element

Supported devices: *All*

Parent: root element

Number of elements per file: 1 (mandatory)

This element defines the contents of the user interface. User interface components are also called widgets. Each widget is defined by a child element of the <ui> element. The following widget elements may be used:

Tag	Description
graph	Graph widget.
number	LCD-style number display.
signalgenerator	Control widget for signal generators.
pwmcontrol	Control widget for PWM and digital high/low outputs.

The element does not support any attributes and only acts as a parent element for the widget elements.

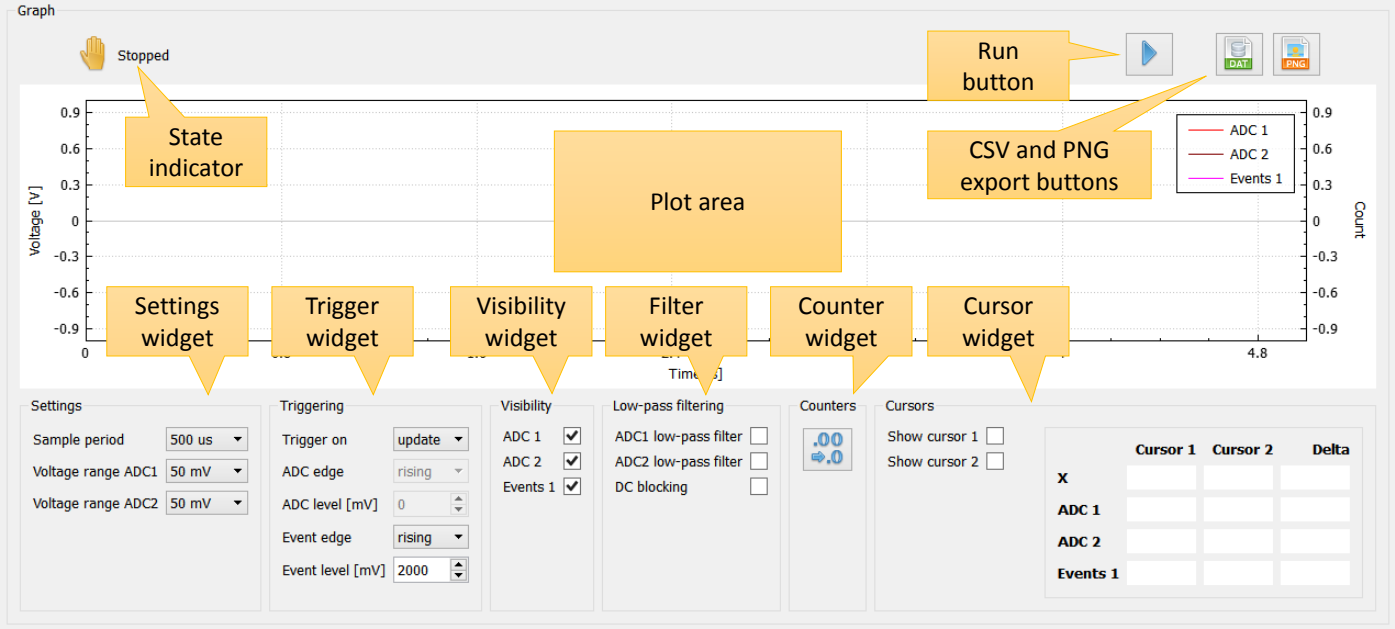
<graph> element

Supported devices: *All*

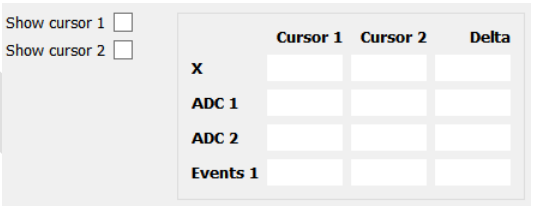
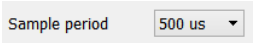
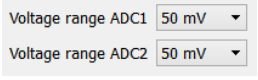
Parent: <ui> element

Number of elements per file: any number

Adds a graph to the user interface. As minimum every graph will display a plot area, state indicator, Run button and CSV and PNG export buttons. Several sub-widgets may be added to a graph as indicated in the screenshot below. The screenshot was taken with a connected *Skorost-II* device.



The following attributes are supported:

Attribute	Type	Devices	Description	Default
SampleSources	Flags	All	0 Nothing 1 ADC1 (analogue channel 1) *Skorost-I*, *II* 2 ADC2 (analogue channel 2) *Skorost-I*, *II* 4 Quadrature decoder 1 *Skorost-I* 8 Quadrature decoder 2 *Skorost-I* 16 Event counter *Skorost-I*, *II* Note: these flags must be equal to or a subset of the SampleSources defined in the <sampling> element.	0
ShowCursorWidget	Bool	All	 Shows the cursor widget. This will also enable the two cursors in the plot area. The cursor includes two check boxes to show or hide the cursors. It also includes a read-out table. The table shows the current X and Y values for both cursors and their differences.	false
ShowSamplingWidget	Bool	All	Shows the settings widget. A drop-down with the available sample periods is displayed. 	false
ShowADCRangeWidget	Bool	All	Shows the settings widget. *Skorost-I*: The two analogue channel voltage ranges cannot be set independently.  *Skorost-II*: The two analogue channel voltage ranges can be set independently. 	false

Attribute	Type	Devices	Description	Default
ShowTriggerWidget	Bool	All	Shows the trigger widget. *Skorost-I*: The trigger for starting a measurement can be set. *Skorost-II*: The trigger for starting a measurement can be set. In addition, when a measurement is triggered by an analogue channel (ADC) the level and slope can be specified. For event counting the event edge and level may also be specified.	false
ShowVisibilityWidget	Bool	All	Shows the visibility widget. Individual traces in the plot area can be shown or hidden.	false
ShowLPFilterWidget	Bool	*Skorost-II*	Shows the low-pass filter widget. This widget is only supported when a *Skorost-II* device is connected. The 300 Hz low-pass filter can be enabled for each analogue channel. A DC-blocking filter can also be enabled.	false
ShowClearCounterWidget	Bool	All	Shows a button to clear quadrature and event counters.	false
ShowCursor1	Bool	All	When set to TRUE, cursor 1 is shown by default. When set to FALSE, cursor 1 is hidden by default. Only has an effect when *ShowCursorWidget* is set to TRUE.	false
ShowCursor2	Bool	All	When set to TRUE, cursor 2 is shown by default. When set to FALSE, cursor 2 is hidden by default. Only has an effect when *ShowCursorWidget* is set to TRUE.	false
Antialiasing	Bool	All	When set to TRUE, anitaliasing is used for traces. Antialiased plotting performance may become slow for traces with a large number of sample points. Set to FALSE, to disable antialiasing and to improve plotting performance.	true
IncrementalUpdate	Bool	All	When set to TRUE, traces are continuously updated with new data from the connected *Skorost* device while a measurement is running. When set to FALSE, the trace is updated only when the measurement is completed.	true

Additional notes:

- The meaning of the default buttons:
  - Run measurement. When the measurement is started, the icon will change to .
  - Stop the measurement.
  - Export graph traces as CSV (comma separates values) data file.

Export graph as PNG image file.

- The different measurement states:

No measurement is currently running. The measurement was stopped.

The measurement was started and the device is waiting for a trigger before starting data collection

The measurement was triggered and data is being collected.

- It is not required to add sub-widgets. All settings controlled by the sub-widgets can also be triggered by control elements. With the control elements the detail of a measurement can be hidden and will simplify the user interface.
- The presentation of the graph widget can be fine-tuned by adding the <presentation> child element. Parameters that can be changed, include the axis ranges, trace colours, trace labels, trace scaling factors and others.
- It is possible to add calculations using the one or more <calculation> child elements. Calculations include add, subtract, multiply and division of two traces and integration of traces.

<number> element

Supported devices: *All*

Parent: <ui> element

Number of elements per file: any number

Adds a large LCD-style widget. This widget will automatically loop measurements and updates continuously. This is a useful widget for demonstrations using overhead projectors. The following attributes are supported:

Attribute	Type	Devices	Description	Default
SampleSources	Flags	All	0 Nothing 1 ADC1 (analogue channel 1) *Skorost-I*, *II* 2 ADC2 (analogue channel 2) *Skorost-I*, *II* 4 Quadrature decoder 1 *Skorost-I* 8 Quadrature decoder 2 *Skorost-I* 16 Event counter *Skorost-I*, *II* Note: these flags must be equal to or a subset of the SampleSources defined in the <sampling> element.	0
DecimalDigits	Integer	All	The number of digits after the decimal point. A number in the range [0..6].	3
ShowClearCounterWidget	Bool	All	Shows a button to clear quadrature and event counters. 	false

Additional notes:

- The presentation of the number widget can be fine-tuned by adding the <presentation> child element. Only a sub-set of attributes of the <presentation> element are supported by this widget. Parameters that can be changed, include the reading labels, scaling factors and a few others.
- The values of this widget will be updated after a measurement is completed. After this, the measurement is restarted. The displayed value is the average of the data points of the previous measurement. As a result, it is possible to achieve very accurate readings with this widget. The update interval is given by: *SampleCount · Effective sample period*. The *SampleCount* and *SamplePeriod* are attributes of the <sampling> element.

<signalgenerator> element

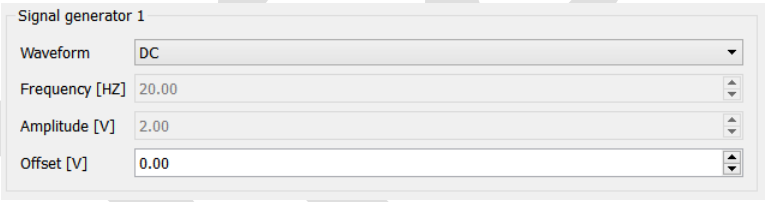
Supported devices: *All*

Parent: <ui> element

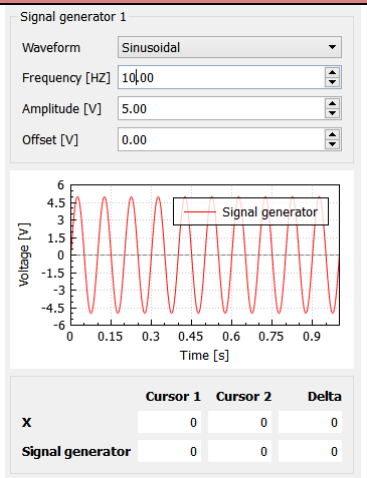
Number of elements per file: any number

The signal generator widget for the *Skorost-I* is shown below. It allows changing the waveform (DC, sawtooth or sinusoidal), frequency, amplitude and offset.

Skorost-II devices have two signal generators. The signal generator 1 widget for the *Skorost-II* is shown below. It allows changing the waveform (DC, sawtooth or sinusoidal), frequency, amplitude and offset.



The following attributes are supported:

Attribute	Type	Devices	Description	Default
ShowGraph	Bool	All	When set to TRUE a graph, showing the current signal generator output, is added below the signal generator control widget. An example is shown in the screenshot to the right. 	false
DACNumber	Integer	*Skorost-II*	The signal generator (DAC) number. This is either 1, for signal generator 1, or 2, for signal generator 2.	1

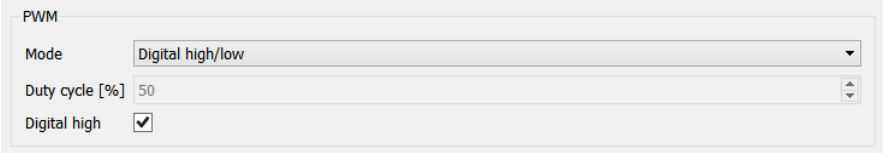
<pwmcontrol> element

Supported devices: *All*

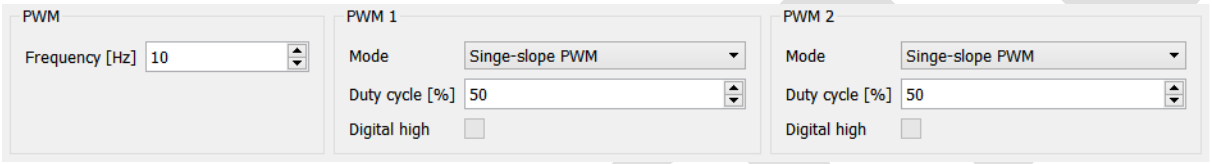
Parent: <ui> element

Number of elements per file: any number

Skorost-I devices have a single PWM/digital output. The PWM control widget for the *Skorost-I* is shown below. It allows changing the mode (digital high/low or PWM) and PWM duty cycle or digital high/low state.



Skorost-II devices have two PWM/digital outputs. The PWM control widget for the *Skorost-II* is shown below. By default, it consists of three group boxes: *PWM*, *PWM 1* and *PWM 2*. It allows changing the PWM frequency. In addition, the mode (digital high/low or PWM), PWM duty cycle and digital high/low state can be set for each PWM/digital output.



The following attributes are supported for *Skorost-II* devices only:

Attribute	Type	Devices	Description	Default
ShowFrequency	Bool	*Skorost-II*	When set to false, the *PWM* group box with the PWM frequency setting is hidden	true
ShowPWM1	Bool	*Skorost-II*	When set to false, the *PWM 1* group box with the mode and duty cycle settings is hidden.	true
ShowPWM2	Bool	*Skorost-II*	When set to false, the *PWM 2* group box with the mode and duty cycle settings is hidden.	true

<presentation> element

Supported devices: *All*

Parent: <graph> or <number> element

Number of elements per parent: 1 (optional)

The <presentation> element is used to fine-tune the graph and number widgets. A separate <presentation> must be specified for each sampled source. The following attributes are supported:

Attribute	Type	Widgets	Description	Default
SampleSource	Enum	graph, number	1 ADC1 (analogue channel 1) *Skorost-I*, *II* 2 ADC2 (analogue channel 2) *Skorost-I*, *II* 4 Quadrature decoder 1 *Skorost-I* 8 Quadrature decoder 2 *Skorost-I* 16 Event counter *Skorost-I*, *II*	0
Label	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	*Depends on SampleSource*

Attribute	Type	Widgets	Description	Default
YDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	*Depends on SampleSource*
YScaling	Float	graph, number	A scale factor for measured data. A negative value will invert the data.	1.0
YMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
YMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
YModulus	Integer	graph, number	This option should only be used for quadrature decoder measurements. For rotary encoding this value should be 1. For linear directional position encoding, where localised quadrature encoders are used, this value should be 4.	1
YAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0
Colour	String	graph	Colour of the trace.	*Depends on SampleSource*
Linearise	Bool	graph	Linearises data after the measurement is completed.	false

Additional notes:

- Automatic Y-axis selection will select the best axis for a combination of different measurements. For example, when both voltages and events counts are measured. The voltages will be displayed on the left axis and the event counts on the right axis. This behaviour can be modified by explicitly setting the Y-axis.

<calculation> element

Supported devices: *All*

Parent: <graph> element

Number of elements per parent: any number

The <calculation> element will add additional traces based on calculations referencing measured data. The following attributes are supported:

Attribute	Type	Description	Default
Operation	Enum	0: Normal expression 1: Integrates the result of the expression (*Skorost-II only*)	0

Attribute	Type	Description	Default
Expression	String	An expression that can reference measured sample sources. Measurements are referenced by using *#SampleSource* variables in the expression: #1 references ADC1 (analogue channel 1) *Skorost-I, II* #2 references ADC2 (analogue channel 2) *Skorost-I, II* #4 references Quadrature decoder 1 *Skorost-I* #8 references Quadrature decoder 2 *Skorost-I* #16 references Event counter *Skorost-I, II* For example: #1 + #2 Adds measurements of the ADC1 and ADC2 channels #4 - #8 Subtracts quadrature decoder 2 from quadrature decoder 1 The expression may contain: Operators +,-,/,*,%,^ Brackets (,) Functions min, max, abs, ceil, floor, round, exp, log, log10, sqrt, clamp, sin, cos, tan, acos, asin, atan, atan2, cosh, cot, csc, sec, sinh, tanh, d2r, r2d Constants pi Variables #1, #2, #4, #8, #16 (see above) x (measured x values, i.e. time)	1
Label	String	Data label displayed in the legend.	Math
YDescription	String	Data Y-axis description.	Math
YScaling	Float	A scale factor for measured data. A negative value will invert the data.	1.0
YMin	Float	The fixed minimum for the Y-axis range.	*auto scale*
YMax	Float	The fixed maximum for the Y-axis range.	*auto scale*
Colour	String	Colour of the trace.	grey
YAxis	Enum	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0

Additional notes:

- When integrating measured voltages, it is highly recommended to enable the low-pass filter and DC-blocking feature of the relevant analogue channels.

Sampling elements

The sampling XML elements are used to define the sampling process and settings. There are different types of sampling elements (methods) and only one element is allowed per XML file. The following sampling methods are provided:

- <sampling> Fixed sampling period. Supports all sampling sources
- <stopwatch> Event-based sampling. Supports only event counting.

Sampling elements also provide a mechanism to change settings of the target device. Settings can be changed when a measurement is started, when a measurement is triggered and/or when a measurement is completed. For this purpose, three elements are provided:

- <prestart> Child control elements are executed when a measurement is started.
- <poststart> Child control elements are executed when a measurement is triggered.
- <poststop> Child control elements are executed when a measurement is completed.
- <loop> Loops the measurement continuously (only when fixed sampling is used).

<sampling> element

Supported devices: *All*

Parent: root element

Number of elements per file: 1

The <sampling> element defines the settings for fixed sampling period measurements. This measurement method supports all sampling sources: analogue voltage, event counting and quadrature decoding measurements. The following attributes are supported:

Attribute	Type	Devices	Description	Default
SamplePeriod	Enum	All	10us *Skorost-II* 20us *Skorost-II* 50us *Skorost-II* 100us *Skorost-II* 200us *Skorost-II* 500us *Skorost-I*,*II* 1ms *Skorost-I*,*II* (*Skorost-II* default) 2ms *Skorost-I*,*II* 5ms *Skorost-I*,*II* (*Skorost-I* default) 10ms *Skorost-I*,*II* 20ms *Skorost-I*,*II* 50ms *Skorost-I*,*II* 100ms *Skorost-I*,*II* 200ms *Skorost-I*,*II* 500ms *Skorost-I*,*II* 1s *Skorost-I*,*II* 2s *Skorost-I*,*II*	
SampleCount	Integer	All	The number of samples to measure for one measurement run. Must be a number in the range [100 .. 50000]. The total measurement time is given by: T_{meas} = SampleCount · SamplePeriod	1000
SampleSources	Flags	All	Flags, specifying which sources to sample: 0 Nothing 1 ADC1 (analogue channel 1) *Skorost-I*, *II* 2 ADC2 (analogue channel 2) *Skorost-I*, *II* 4 Quadrature decoder 1 *Skorost-I*	0

Attribute	Type	Devices	Description	Default
			8 Quadrature decoder 2 *Skorost-I* 16 Event counter *Skorost-I*, *II*	
SamplesToClear	Flags	*Skorost-I*	Flags, specifying which counters to clear when a measurement is triggered: 0 Nothing 4 Quadrature decoder 1 8 Quadrature decoder 2 16 Event counter	0
ClearCounter	Bool	*Skorost-II*	When set to TRUE the event counter is cleared, when a measurement is triggered	false
Trigger	Enum	All	When a measurement is started, data acquisition will only start after the specified trigger is encountered. For *Skorost-I*: 0 Trigger on measurement start 1 Trigger on quadrature decoder 1 2 Trigger on quadrature decoder 2 3 Trigger on event counter For *Skorost-II*: 0 Trigger on measurement start 1 Trigger on event counter 2 Trigger on analogue channel 1 3 Trigger on analogue channel 2 Additional trigger settings are specified by the <trigger> element.	0
SampleADCRange	Enum	*Skorost-I*	Voltage range used for both analogue channels: 0 ±10V 1 ±1V	0
SampleADC1Range	Enum	*Skorost-II*	Voltage range used for analogue channel 1: 0 ±5V 1 ±1V 2 ±200mV 3 ±50mV	0
SampleADC2Range	Enum	*Skorost-II*	Voltage range used for analogue channel 2: 0 ±5V 1 ±1V 2 ±200mV 3 ±50mV	0
EventEdge	Enum	*Skorost-I*	Specifies the edge used for event counting: 0 falling edge 1 rising edge For *Skorost-II* devices use the <trigger> element.	1
LP1Filter	Bool	*Skorost-II*	When set to TRUE the 300 Hz low-pass filter is enabled on analogue channel 1.	false
LP2Filter	Bool	*Skorost-II*	When set to TRUE the 300 Hz low-pass filter is enabled on analogue channel 2.	false
DCBlocking	Bool	*Skorost-II*	When set to TRUE DC-blocking will be applied on all analogue channels.	false

<stopwatch> element

Supported devices: *All*

Parent: root element

Number of elements per file: 1

The <stopwatch> element defines the settings for timestamp-based measurement. This measurement method only supports event counting. The timestamp of each event is taken during a measurement run. This measurement method provides much better time resolution, than the fixed sampling period method. The following attributes are supported:

Attribute	Type	Devices	Description	Default
StopwatchDuration	Integer	All	The measurement duration in [ms].	5000
Trigger	Enum	All	When a measurement is started, data acquisition will only start after the specified trigger is encountered. For *Skorost-I*: 0 Trigger on measurement start 3 Trigger on event counter For *Skorost-II*: 0 Trigger on measurement start 1 Trigger on event counter	0
EventEdge	Enum	*Skorost-I*	Specifies the edge used for event counting: 0 falling edge 1 rising edge For *Skorost-II* devices use the <trigger> element.	1

<trigger> element

Supported devices: *Skorost-II*

Parent: root element

Number of elements per parent: 1 (optional)

This element is used to define advanced trigger settings for *Skorost-II* devices. The following attributes are supported:

Attribute	Type	Devices	Description	Default
EventEdge	Enum	*Skorost-II*	Specifies the edge used by the event channel, when it is used as a trigger: 0 falling edge 1 rising edge	1
EventLevel	Integer	*Skorost-II*	The trigger level for the event channel in [mV]. It must be a number in the range [0 .. 5000].	2000
ADCEdge	Enum	*Skorost-II*	Specifies the trigger edge, when a measurement is triggered by one of the analogue channels (see <sampling> element): 0 falling edge 1 rising edge	1
ADCLevel	Integer	*Skorost-II*	The trigger level for the analogue channel in [mV]. It must be a number in the range [-5000 .. 5000].	0

<prestart> element

Supported devices: *All*

Parent: <sampling> or <stopwatch> element

Number of elements per parent: 1 (optional)

This element only acts as a parent element (placeholder) and does not have any attributes. Child control elements will be executed when a measurement is started, but before the measurement is triggered. For example, it is possible to change the digital high/low output from a low to a high state when a measurement is started.

<poststart> element

Supported devices: *All*

Parent: <sampling> or <stopwatch> element

Number of elements per parent: 1 (optional)

This element only acts as a parent element (placeholder) and does not have any attributes. Child control elements will be executed after a measurement is triggered.

<poststop> element

Supported devices: *All*

Parent: <sampling> or <stopwatch> element

Number of elements per parent: 1 (optional)

This element only acts as a parent element (placeholder) and does not have any attributes. Child control elements will be executed when a measurement is completed or stopped.

<loop> element

Supported devices: *All*

Parent: <sampling> element

Number of elements per parent: 1 (optional)

When this element is added the measurement will loop continuously. The following attributes are supported:

Attribute	Type	Devices	Description	Default
WaitTime	Float	All	The time in [s] the device waits, before starting with the next measurement run. Must be in the range [0 .. 100].	5.0

Control elements

Control XML elements define the various device settings. Control elements can be specified under four different contexts:

- Default settings: These settings are defined by root control elements and are applied when a device is connected.
- Pre-start settings: Control elements are defined as children of the <prestart> element. These settings are applied when a measurement is started.
- Post-start settings: Control elements are defined as children of the <poststart> element. These settings are applied just after a measurement is triggered.
- Post-stop settings: Control elements are defined as children of the <poststop> element. These settings are applied when a measurement is completed or stopped.

<pwm>, <pwm1> and <pwm2> elements

Supported devices: *Skorost-I* (<pwm>, <pwm1>)
Skorost-II (<pwm1>, <pwm2>)

Parent: root element, <prestart>, <poststart>, <poststop>

Number of elements per file: Any

These elements define the PWM settings. Using one of these elements will automatically enable the PWM signal on the relevant output. *Skorost-I* devices only have one PWM output. Either the <pwm> or <pwm1> element can be used to configure this single PWM output. *Skorost-II* devices have two PWM outptus. The <pwm1> and <pwm2> elements are used to configure the PWM output 1 and 2, respectively.

The following attributes are supported:

Attribute	Type	Devices	Description	Default
DutyCycle	Integer	All	Defines the duty cycle in percent. This must be a number in the range [0 .. 100].	50
Trip	Enum	*Skorost-II*	This feature can be used to set the PWM duty cycle to 0%, when the event channel is triggered (see <trigger> element): 0 Trip feature is disabled 1 Trip the PWM (set the duty cycle to 0%) when the event channel is triggered When the PWM is tripped, an error message will be displayed by the *Skorost UI*. ⚠ Do not use the trip feature when the event channel is used to trigger a measurement. This may result in undefined behaviour.	0

<pwmfrequency>

Supported devices: *Skorost-II*

Parent: root element, <prestart>, <poststart>, <poststop>

Number of elements per file: Any

This element defines the PWM frequency. The following attributes are supported:

Attribute	Type	Devices	Description	Default
Frequency	Integer	*Skorost-II*	Defines the PWM frequency in [Hz]. This must be a number in the range [1 .. 100000].	10000

<digiout>, <digiout1> and <digiout2> elements

Supported devices: *Skorost-I* (<digiout>, <digiout1>)
Skorost-II (<digiout1>, < digiout2>)

Parent: root element, <prestart>, <poststart>, <poststop>

Number of elements per file: Any

These elements define the digital high/low output settings. Using one of these elements will automatically enable the digital high/low signal and disable the PWM signal on the relevant output. *Skorost-I* devices only have one digital high/low output. Either the <pwm> or <pwm1> element can be used to configure this single output. *Skorost-II* devices have two digital high/low outptus. The <pwm1> and <pwm2> elements are used to configure the digital high/low output 1 and 2, respectively.

The following attributes are supported:

Attribute	Type	Devices	Description	Default
State	Bool	All	Settings this attribute to TRUE will result in a high output state on the digital output. Settings this attribute to FALSE will result in a low output state on the digital output.	50

<dac>, <dac1> and <dac2> elements

Supported devices: *Skorost-I* (<dac>, <dac1>)
Skorost-II (<dac1>, < dac2>)

Parent: root element, <prestart>, <poststart>, <poststop>

Number of elements per file: Any

These elements define the signal generator (DAC) settings. *Skorost-I* devices only have one signal generator. Either the <dac> or <dac1> element can be used to configure this single output. *Skorost-II* devices have two signal generators. The <dac1> and <dac2> elements are used to configure the signal generator 1 and 2, respectively.

The following attributes are supported:

Attribute	Type	Devices	Description	Default
WaveformShape	Enum	All	Specifies the signal generator waveform: 0 DC (constant voltage) 1 Ramp/Sawtooth 2 Sinusoidal	2
Frequency	Float	All	Specifies the output frequency in [Hz] for ramp and sinusoidal waveforms. The allowed ranges: *Skrosot-I* : [1 .. 2500] *Skrosot-II*: [1 .. 20000]	1000.0
Amplitude	Float	All	Specifies the output amplitude in [V] for ramp and sinusoidal waveforms. The allowed ranges: *Skrosot-I* : [0 .. 5.0] *Skrosot-II*: [0 .. 4.5]	1.0
Offset	Float	All	Specifies the output offset in [V] for DC, ramp and sinusoidal waveforms. The allowed ranges: *Skrosot-I* : [-5.0 .. 5.0] *Skrosot-II*: [-4.5 .. 4.5]	0.0

<clear> element

Supported devices: *All*

Parent: root element, <prestart>, <poststart>, <poststop>

Number of elements per file: Any

Clears the event and/or quadrature counters. The following attributes are supported:

Attribute	Type	Devices	Description	Default
SampleSources	Flags	*Skorost-I*	Flags, specifying which counters to clear when a measurement is triggered: 0 Nothing 4 Quadrature decoder 1 8 Quadrature decoder 2 16 Event counter	28 $(11100)_2$

Writing XML configurations for different device types

When writing XML configuration files, they can be written for a specific device (e.g. only for *Skorost-I* devices) or the can be written as general configurations for several *Skorost* device types.

Writing XML configurations for specific devices

Elements and their attributes only need to comply with one device type. For example, it is possible to write XML configurations only for *Skorost-I* devices. In order to inform the *Skorost UI* that an XML configuration is targeted at a specific device, the *device* attribute must be defined in the <config> element:

Attribute	Type	Devices	Description	Default
device	Enum	All	0 All devices 1 Only *Skorost-I* 2 Only *Skorost-II*	0

The *Skorost* UI will only list those XML configuration files, which are targeted at any device or which are targeted to the currently connected device.

Writing XML configurations for several devices

It is possible to write XML configurations for several device types. The following mechanisms are provided:

- *Skorost UI* will ignore elements and attributes, which are not supported by the currently connected device. Always define all relevant attributes for all devices. Some attributes are only applicable to *Skorost-I* and some attributes are only applicable to *Skorost-II* devices. For example: define the *SampleADCRange*, *SampleADC1Range* and *SampleADC2Range* to make sure the analogue input ranges are correctly defined for both *Skorost-I* and *Skorost-II* devices.
- Different *Skorost* device types may have different number of input and/or output ports. For example: *Skorost-I* has one signal generator and *Skorost-II* has two signal generators. When writing control elements, always use a common denominator approach. For example, only use the <pwm1>, digiout1> and <dac1> elements, when defining PWM, digital output and signal generator control elements.
- Only use sample sources, which are supported by all devices. Currently, only analogue channels and event counting are supported by all *Skorost* devices.

Deprecated elements

Some of the XML elements have changed and have been deprecated. Skorost-UI still provides backwards compatibility for the old syntax, but it should not be used in new scripts.

<presentation> element

Supported devices: *All*

Parent: <graph> or <number> element

Number of elements per parent: 1 (optional)

A single <presentation> element was used to fine-tune the graph and number widgets. The new approach uses a generic presentation element for each sampled source. The following attributes are supported:

Attribute	Type	Widgets	Description	Default
Attributes for analogue channel 1				
ADC1Label	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	ADC 1
ADC1YDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	Voltage [V]
ADC1YScaling	Float	graph, number	A scale factor for measured data. A negative value will invert the data.	1.0
ADC1YMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
ADC1YMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
ADC1Colour	String	graph	Colour of the trace.	red
ADC1YAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0
Attributes for analogue channel 2				
ADC2Label	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	ADC 2
ADC2YDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	Voltage [V]
ADC2YScaling	Float	graph, number	A scale factor for measured data. A negative value will invert the data.	1.0
ADC2YMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
ADC2YMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
ADC2Colour	String	graph	Colour of the trace.	darkRed
ADC2YAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0
Attributes for quadrature decoder 1				
Quadrature1Label	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	Quad 1

Attribute	Type	Widgets	Description	Default
Quadrature1YDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	Count
Quadrature1YScaling				
Quadrature1YModulus	Integer	graph, number	For rotary encoding this value should be 1. For linear directional position encoding, where localised quadrature encoders are used, this value should be 4.	1
Quadrature1YMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
Quadrature1YMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
LineariseQuadrature1	Bool	graph	Linearises data after the measurement is completed.	false
Quadrature1Colour	String	graph	Colour of the trace.	green
Quadrature1YAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0
Attributes for quadrature decoder 2				
Quadrature2Label	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	Quad 2
Quadrature2YDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	Count
Quadrature2YScaling	Float	graph, number	A scale factor for measured data. A negative value will invert the data.	1.0
Quadrature2YModulus	Integer	graph, number	For rotary encoding this value should be 1. For linear directional position encoding, where localised quadrature encoders are used, this value should be 4.	1
Quadrature2YMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
Quadrature2YMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
LineariseQuadrature2	Bool	graph	Linearises data after the measurement is completed.	false
Quadrature2Colour	String	graph	Colour of the trace.	darkGreen
Quadrature2YAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0
Attributes for event counter 1				
Event1Label	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	Events 1
Event1YDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	Count
Event1YScaling	Float	graph, number	A scale factor for measured data. A negative value will invert the data.	1.0
Event1YMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
Event1YMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
LinearisEvent1	Bool	graph	Linearises data after the measurement is completed.	false

Attribute	Type	Widgets	Description	Default
Event1Colour	String	graph	Colour of the trace.	magenta
Event1YAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0
Attributes for stopwatch measurements				
StopwatchLabel	String	graph, number	Data label. *Graph*: Label displayed in the legend. *Number*: Label displayed before the LCD-style reading.	Stopwatch
StopwatchYDescription	String	graph, number	Data description. *Graph*: Y-axis description. *Number*: Description displayed after the LCD-style reading.	Count
StopwatchYScaling	Float	graph, number	A scale factor for measured data. A negative value will invert the data.	1.0
StopwatchYMin	Float	graph	The fixed minimum for the Y-axis range.	*auto scale*
StopwatchYMax	Float	graph	The fixed maximum for the Y-axis range.	*auto scale*
StopwatchColour	String	graph	Colour of the trace.	red
StopwatchYAxis	Enum	graph	0: Automatically determine the Y-axis 1: Left Y-axis 2: Right Y-axis	0

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