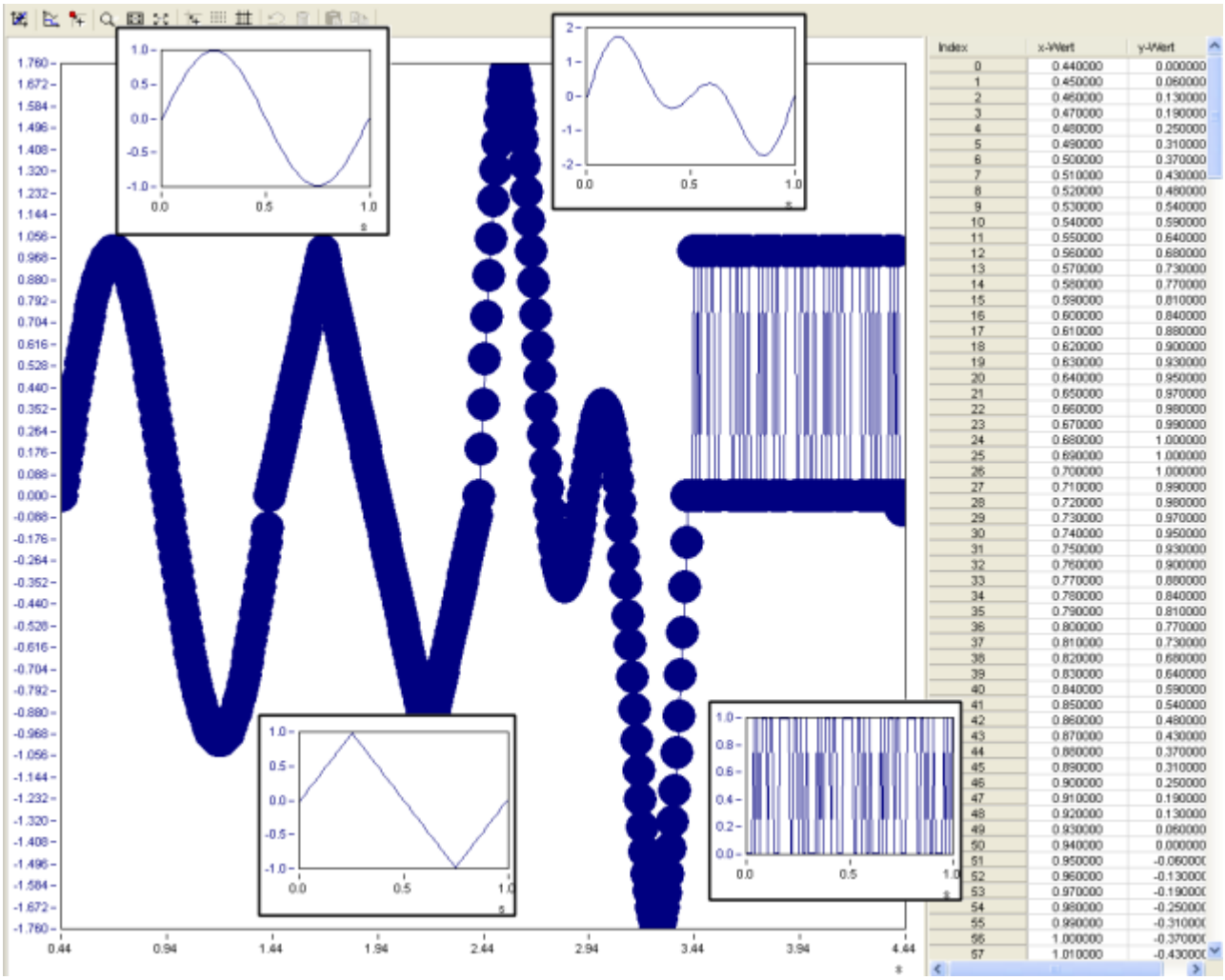
An Axiometrix Solutions Brand

imc CRONOS system family

Synthesizer Manual

Edition 2 - 2022-07-08



imc Test & Measurement GmbH • Voltastr. 5 • 13355 Berlin • Germany

Disclaimer of liability

The contents of this documentation have been carefully checked for consistency with the hardware and software systems described. Nevertheless, it is impossible to completely rule out inconsistencies, so that we decline to offer any guarantee of total conformity.

We reserve the right to make technical modifications of the systems.

Copyright

Open Source Software Licenses

Some components of imc products use software which is licensed under the GNU General Public License (GPL). Details are available in the About dialog.

A list of the open source software licenses for the imc measurement devices is located on the imc STUDIO/imc WAVE installation medium in the folder "*Products\imc DEVICES\OSS*" or "*Products\imc DEVICEcore\OSS*". If you wish to receive a copy of the GPL sources used, please contact our Hotline.

Table of Contents

1 General introduction

1.1 imc Customer Support / Hotline

If you have problems or questions, please contact our Customer Support/Hotline:

imc Test & Measurement GmbH

Hotline (Germany): **+49 30 467090-26**

E-Mail: hotline@imc-tm.de

Internet: https://www.imc-tm.com

International partners

For our international partners see https://www.imc-tm.com/distributors/.

Tip for ensuring quick processing of your questions:

If you contact us **you would help us**, if you know the **serial number of your devices** and the **version info of the software**. This documentation should also be on hand.

- The device's serial number appears on the nameplate.
- The program version designation is available in the About-Dialog.

1.2 Legal notices

Quality Management

imc Test & Measurement GmbH holds DIN-EN-ISO-9001 certification since May 1995. You can download the CE Certification, current certificates and information about the imc quality system on our website: https://www.imc-tm.com/quality-assurance/.

imc Warranty

Subject to the general terms and conditions of imc Test & Measurement GmbH.

Liability restrictions

All specifications and notes in this document are subject to applicable standards and regulations, and reflect the state of the art well as accumulated years of knowledge and experience. The contents of this document have been carefully checked for consistency with the hardware and the software systems described. Nevertheless, it is impossible to completely rule out inconsistencies, so that we decline to offer any guarantee of total conformity. We reserve the right to make technical modifications of the systems.

The manufacturer declines any liability for damage arising from:

- failure to comply with the provided documentation,
- inappropriate use of the equipment.

Please note that all properties described refer to a closed measurement system and not to its individual slices.

Guarantee

Each device is subjected to a 24-hour "burn-in" before leaving imc. This procedure is capable of detecting almost all cases of early failure. This does not, however, guarantee that a component will not fail after longer operation. Therefore, all imc devices are granted liability for a period of two years. The condition for this guarantee is that no alterations or modifications have been made to the device by the customer.

Unauthorized intervention in the device renders the guarantee null and void.

Notes on radio interference suppression

Any additional products connected to the product must satisfy the EMC requirements as specified by the responsible authority (within Europe[1]) in Germany the BNetzA - "Bundesnetzagentur" (formerly BMPT-Vfg. No. 1046/84 or No. 243/91) or EC Guidelines 2014/30/EU. All products which satisfy these requirements must be appropriately marked by the manufacturer or display the CE certification marking.

Products not satisfying these requirements may only be used with special approval of the regulating body in the country where operated.

Note

The EMC tests were carried out using shielded and grounded input and output cables with the exception of the power cord. Observe this condition when designing your experiment to ensure high interference immunity and low jamming.

[1] If you are located outside Europe, please refer the appropriate EMC standards used in the country of operation.

Cables and leads

Unless otherwise indicated, no connection leads may be long leads (< 30 m) as defined by the standard IEC 61326-1. LAN-cables (RJ 45) and CAN-Bus cables (DSUB-9) are excepted from this rule.

Only cables with suitable properties for the task (e.g. isolation for protection against electric shock) may be used.

ElektroG, RoHS 2, WEEE, CE

The imc Test & Measurement GmbH is registered with the authority as follows:
WEEE Reg. No. DE 43368136
valid from 24.11.2005

Reference

https://www.imc-tm.com/elekrog-rohs-weee/ and https://www.imc-tm.com/ce-conformity/

FCC-Notice

This product has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment on and off, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult our imc Hotline or an experienced technician for help.

Modifications

The FCC requires the user to be notified that any changes or modifications made to this product that are not expressly approved by imc may void the user's authority to operate this equipment.

1.3 Guide to Using the Manual

To look for WHAT?	Contents
You should really read the following chapters!	
Synthesizer 8	
Synthesizer configuration 10	General notes and remarks
Segment editor 25	Description of the segment editor and its functions
Program editor 33	Description of the program editor and its functions
Signal configurator 48	Description of the signal configuration
Synthesizer with Controller Functionality 58	Description of the controller and the controller features

Regularly updated information and up-to-date user's manuals can be accessed on https://www.imc-tm.com.

2 Synthesizer

The Synthesizer's task is to play back defined curve segments sequentially. The order in which the segments are played back is determined externally by certain bits or counter events.

The Synthesizer software can be divided into **two sections**:

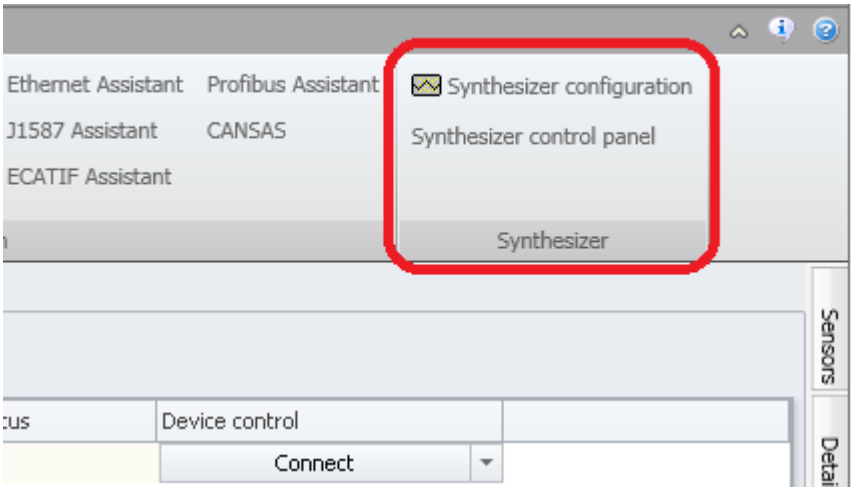
- In the first section, curve segments can be defined. The curve segment playback sequence can also be controlled by a program. We will refer to this section as the **Synthesizer Project**.
  Besides the specified curve segments, the programs also contain conditions which determine the curve segments that are played back. These conditions only take the form of placeholders in the synthesizer project's program and must be linked to real conditions.
- In the second section the pre-defined programs and segments are assigned to particular signals, which finally define the output. Up to eight synthesizer outputs SynthDAC_x can be used for the output. In a Signal Configurator 48 the links to the conditions mentioned above, as well as the outputs, are assigned.

2.1 Terms

Term	Definition
Flowchart	A Flowchart is a sequence that can be edited in the Program editor 33.
Program	The program specifies the order of playback for the segments, as well as the conditions attached.
Segment	A segment is a defined curve segment.
Signal	A signal contains a program, as well as certain parameters which monitor or process the output value computed by the program. This include the gain and offset, in the simplest case.
Signalname	Name of the synthesizer signal. Not to be confused with other signals such as analog input or output channels.
Branching	You can find a description/definition in chapter: Inserting a for-/ backward branch 35
Watchdog	Each Synthesizer comes with a Watchdog, which is configured with the device properties. The Watchdog-bit can then be selected in the branch for the Synthesizer card.

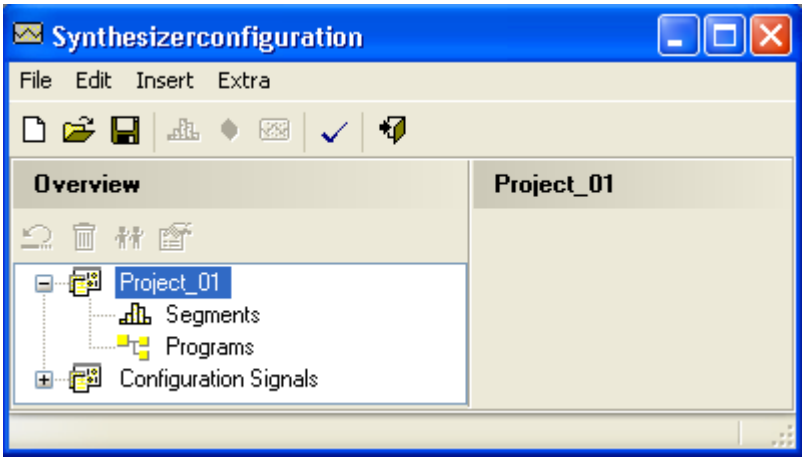
3 Calling the Synthesizer

In imc STUDIO, the synthesizer is called from the setup page:



This item is only visible for devices equipped with a synthesizer.

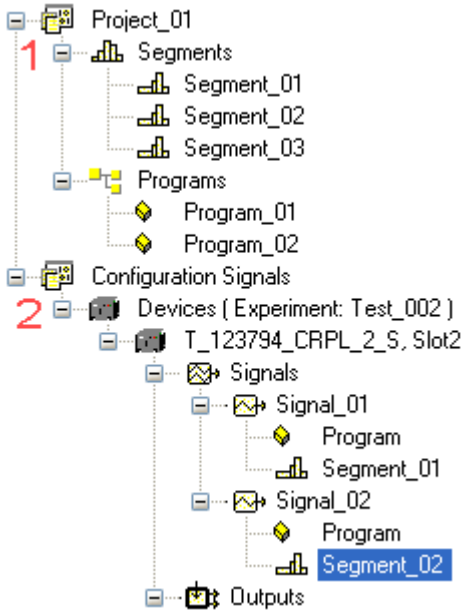
The assistant for the ***synthesizer configuration*** appears:



main dialog

4 Synthesizer configuration

The overview presents the two main sections of the **Synthesizer software**.

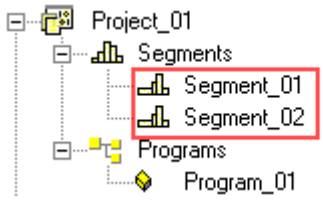


1 In the **Synthesizer Project**, it is possible to define *Segments* and *Programs* which are saved to the hard drive as a file. A project is **not** referenced to the Synthesizer hardware.

2 In the **Signal Configurator**, a project's individual programs are assigned to individual Synthesizer boards. Only then - there is a reference to particular hardware.

Once the tree diagram is expanded, it's possible to edit segments and programs in the Synthesizer project or to change the signals in the Synthesizer configuration.

4.1 Creating a new segment



Either the menu item *Insert / Segment / New* or the shortcut  (see description of shortcuts 11) can be used to define a new segment.

It first receives a default name which is composed of the character string Segment_ plus a consecutive number. In the overview, this segment then appears under the Segments header.

This segment can be changed by means of a segment editor Segment editor 25

4.2 Loading a segment

Under the menu item *Insert / Segment / Load*, a curve saved in FAMOS data format can be opened.

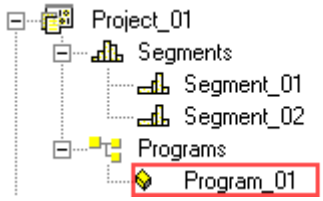
It is also possible to switch segments during the measurement. On this topic, observe the description in Managing the segment files 44 and the material on the option ***Reload*** 54 in the section Sub-dialog Parameter 50

.

Note

Curves in any other data formats can also be loaded. However, the format must first be converted to FAMOS data by a file import filter; refer to the FAMOS manual for more information. The file formats and their associated import filters can be registered with the synthesizer as per Options dialog 23 of this documentation.

4.3 Creating a new program



Using either the menu item Insert / Program or the shortcut , a new program can be defined. At first it receives the name Program_XX, where XX is a consecutive number. In the overview, this program appears under the Programs header. Find here a description 33 of creating a new program.

4.4 Toolbar, menu shortcuts in the main dialog

The actions initiated by means of the toolbar only affect the Overview section. It's possible to add individual elements such as segments, programs or signals to the overview, or to delete or rename them.

Below is a description of the actions symbolized by each shortcut.

Symbol	Definition
	Creates a new, empty synthesizer project.
	Opens an already existing synthesizer project. The project files have the ending "prj".
	Saves a synthesizer project.
	Restores any individual elements previously deleted (segments, programs, signals).
	Deletes a segment, program or signal.
	Duplicates the element currently selected in the overview.
	Properties dialog for name and comment of the element currently selected in the overview.
	Creates an empty new segment with a default name
	Creates an empty new program with a default name
	Creates a new signal with a default name. The associated Synthesizer board and program must be established
	Closes the Synthesizer software

4.4.1 File

New:

With this menu item, an empty new synthesizer project is created. The existing signal configuration is closed. The new project receives the name Project_XX, where XX is a consecutive number. This number is incremented with each new project and reset to 1 when the software is closed. After quitting a confirmation prompt, the old project can be saved (see at Save project 11 or Save project as 12). If the user declines the prompt to save, the changes are not saved but a new project is created.

New Project:

A new empty Synthesizer project is created. Existing signal configurations remain intact.

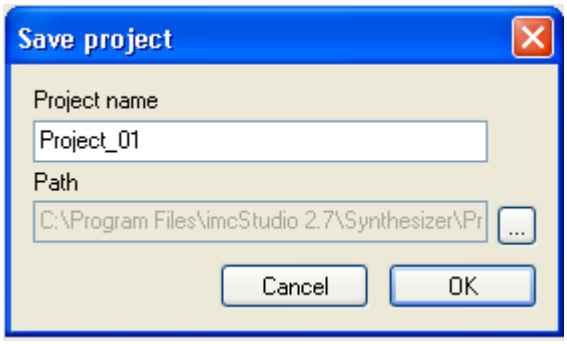
Open Project:

A dialog appears in which the desired project can be selected.
Project files have the extension "prj". If a previously edited project was not saved, a confirmation prompt appears.

Save Project:

Saves the active project - If the project was newly created, a dialog, in which to set the name appears.

Save Project as..:



The following dialog appears and prompts the user to specify the storage location.

In the input box Project name, the project's name can be specified. Below, the project's filepath is displayed.

Using the button ..., the filepath can be changed by making a different selection. A folder is created with the project's name under this path. This becomes the location to which the project's files are saved.

Configuration Signals - New:

The existing signal configuration and controller settings are deleted.

Configuration Signals - Import/Export:

Saving and loading of signal configurations

Menu item Close:

This menu item can be used to exit the software. If you did not save your changes, a confirmation prompt will appear.

4.4.2 Edit

Undo:

Segments, programs and signals can be deleted using the menu item Delete or the button . The restoration only applies to the deletion in the overview. Actions in the respective editor (segment editor 25, program editor 33, signal configurator 48) use their own toolbars.

Duplicate:

The selected element (segment, program or signal) is copied and inserted under the same tree diagram header, with the name Copy_Y_<element name>, where Y is an incremented number.

Delete:

The selected segment, program or signal is removed. Deletion can be reversed using the button or by the menu item Undo.

Properties:

With this menu item, the name and comment for the selected element (segment, program, signal) can be changed. Properties dialog 24 for details.

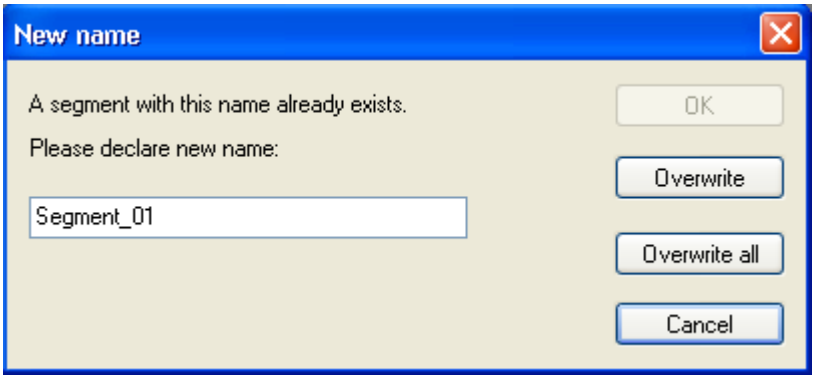
4.4.3 Insert

Segment->New:

- An empty segment with the name Segment_XX is inserted. XX is an incremented number. To change the name of the segment or if you want to browse the properties please consider the properties 12 or the properties dialog 24. Edit values of a segment can be learned by the Segment editor 25.

Segment->Load ... :

- Create a segment out of a file.
- Along with FAMOS files (RAW/DAT) it is also possible to load files in other formats. For this purpose, a file import filter must exist or the particular file format and be registered with the Synthesizer for use. For further information please consider Registering import filters 24.
- If one file contains multiple curves, all the curves will be loaded and inserted as segments.
- If the project includes segments having the same names as curves in the file, a prompting message appears:



OK: The button OK is enabled once the curve has been assigned a new name. Use this button and the segment is adopted into the project with the new name.

Overwrite: The segment in the project is overwritten with the data of the curve from the file.

Overwrite all: This and all other segments are overwritten.

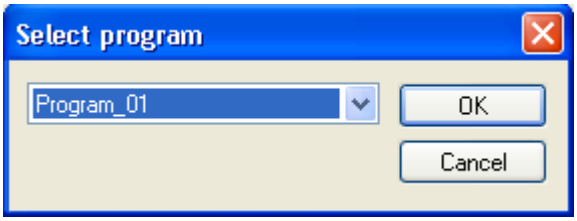
Program:

- A new program with a default name is created. This consists of the character string "Program_" plus a two-digit running increment. On editing the program's name, see properties 12 or properties dialog 24.

Changing the program data is performed using Program editor 33.

Signal:

- Before this item can be used, a device/slot must be selected. This determines for which device or Synthesizer board the signal is to be defined.



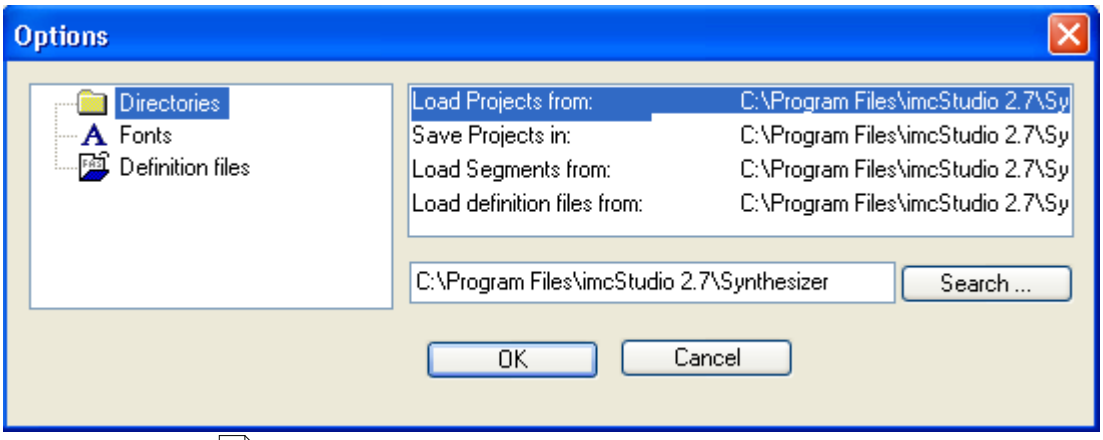
A dialog appears in which you are prompted to select a program from the synthesizer project which is to apply to the signal:

- The combobox offers the programs available for selection, and one of them must be selected.
- By clicking OK a new signal for the selected program is created and is automatically assigned a default name which consists of the string "Signal_" plus a two-digit incremented number.
- See properties dialog 24 on how the signal's name can be edited. Configuring the signal is accomplished in the signal configurator, Signal configuration 48
- Instead of using this indirect way, it's possible to use Drag & Drop to apply a program for creating a new signal. See Drag & Drop 22 for more info.

4.4.4 Extra

Options:

A dialog appears with which certain permanent settings options such as loading/saving paths, fonts etc. can be set:

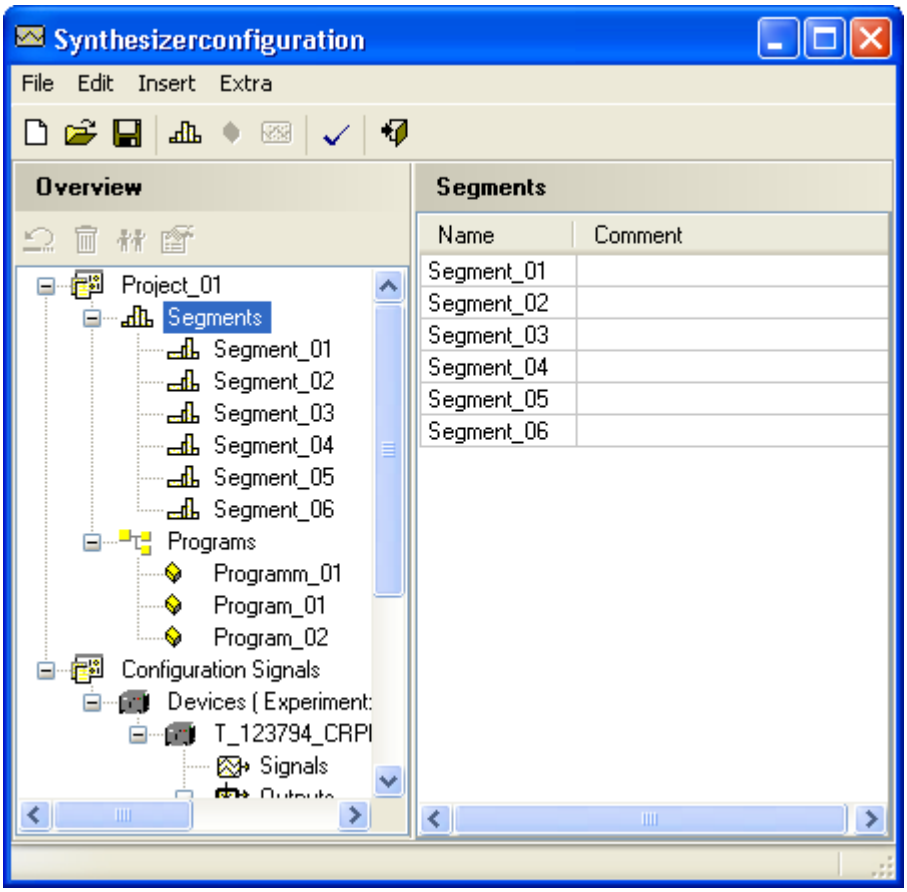


See Option dialog 23 for more information.

4.5 Tree diagram in the main dialog

Tree diagram selections Segments, Programs, Signals

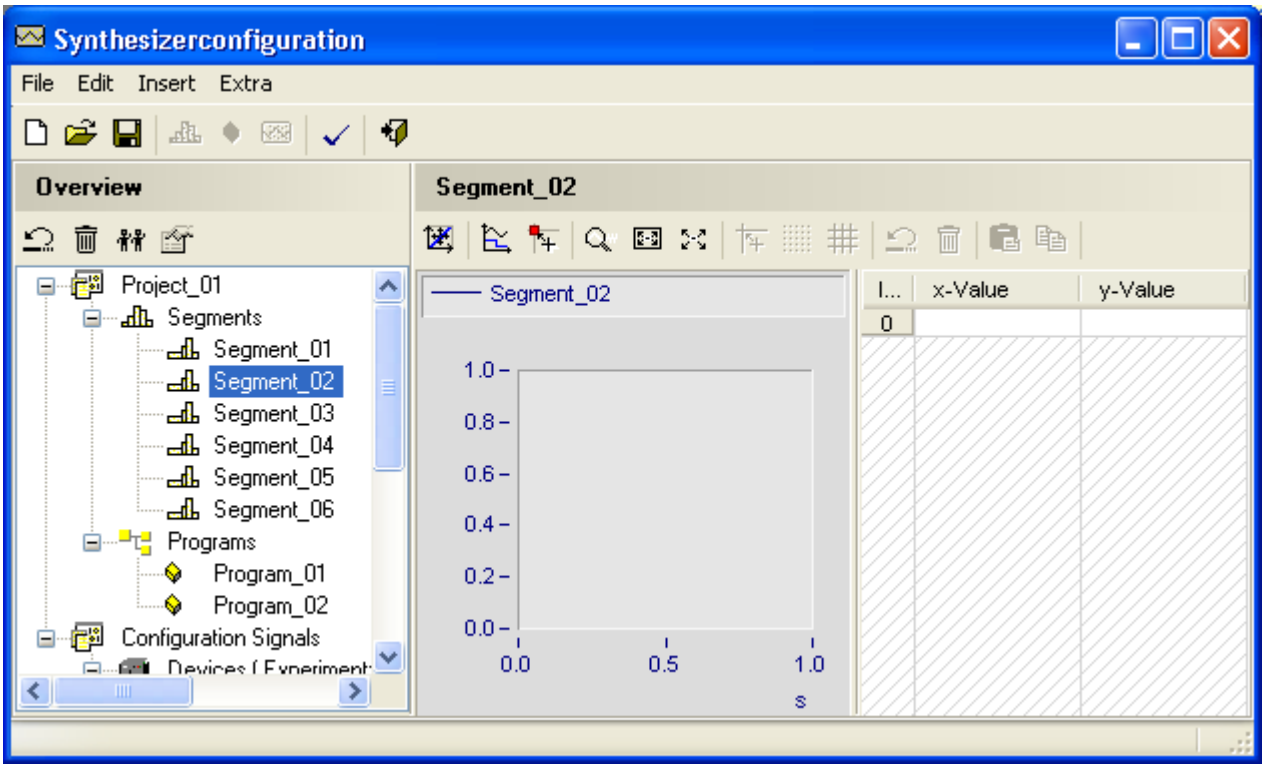
If one of the tree diagram elements Segments, Programs or Signals is selected, a list of showing names and comments on the associated elements appears.



In this list, the name and comment can be edited by selecting the associated element and entering new values from the keyboard. Name changes aren't reflected in the tree diagram right away, but only after selecting a different tree element.

Selecting a segment:

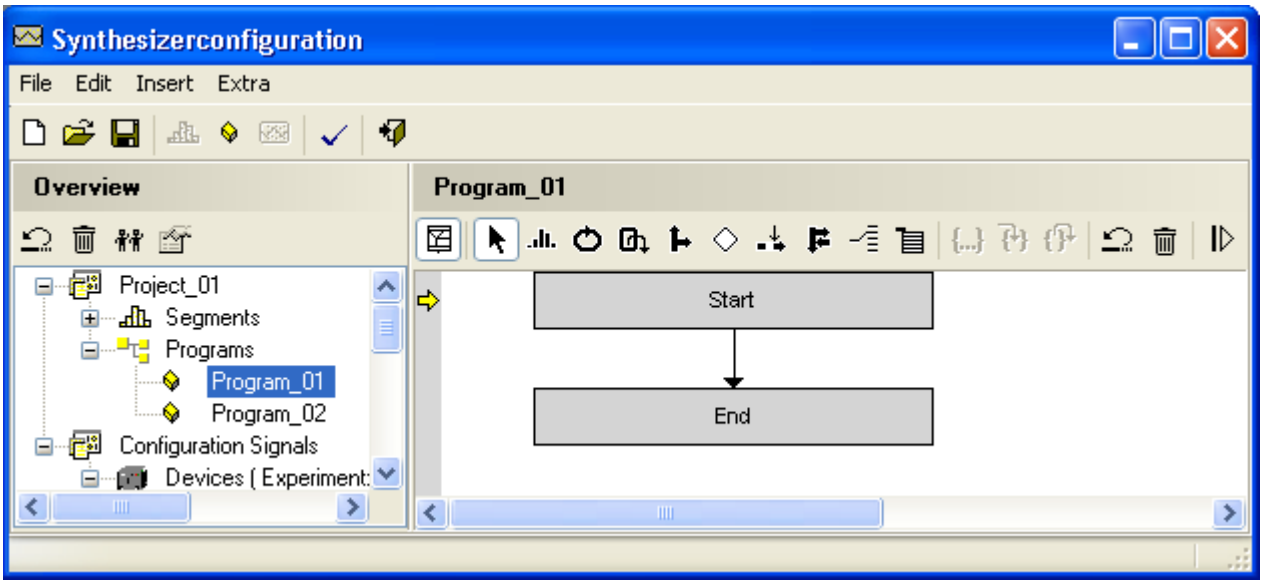
If you select a segment in the tree diagram, an editor with which the segment can be edited appears on the right side:



For details, please consider the chapter Segment editor 25.

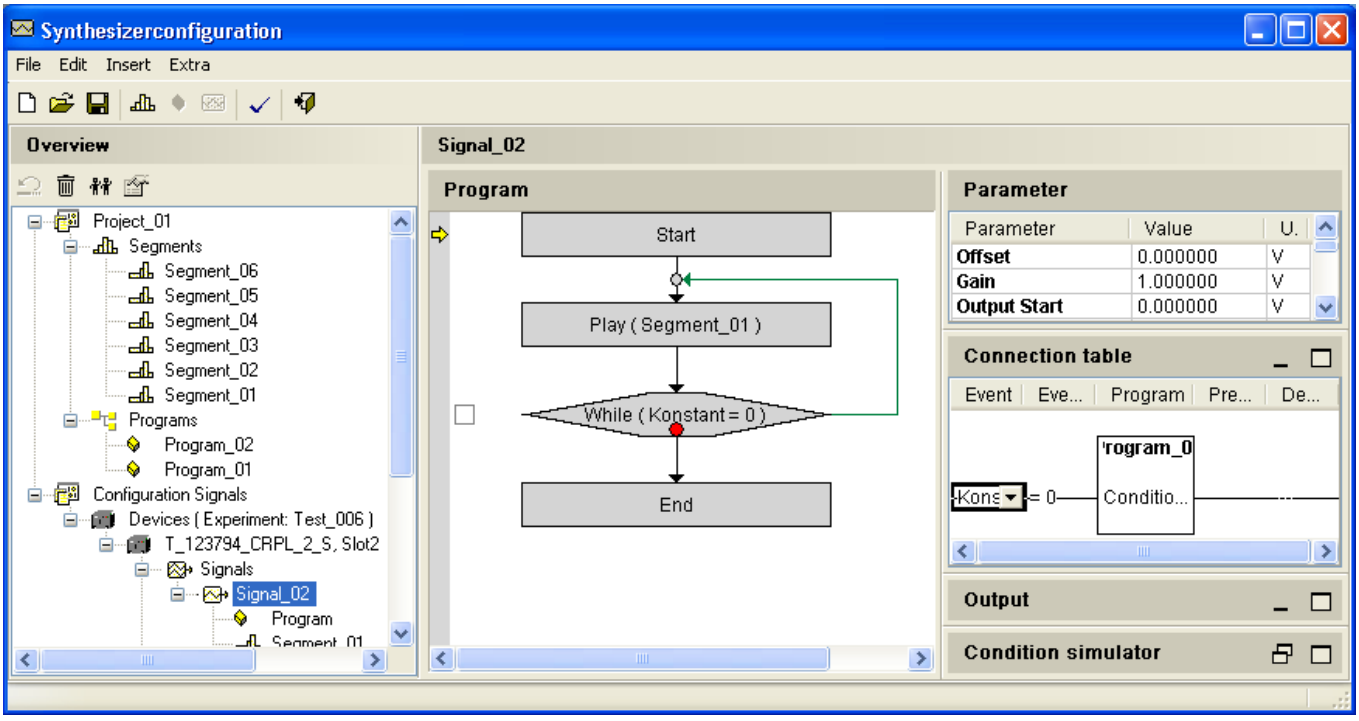
Selecting a program:

If you select a program in the tree diagram, a program editor appears on the right side of the screen. A description is presented in Program editor 33.



Selecting a signal:

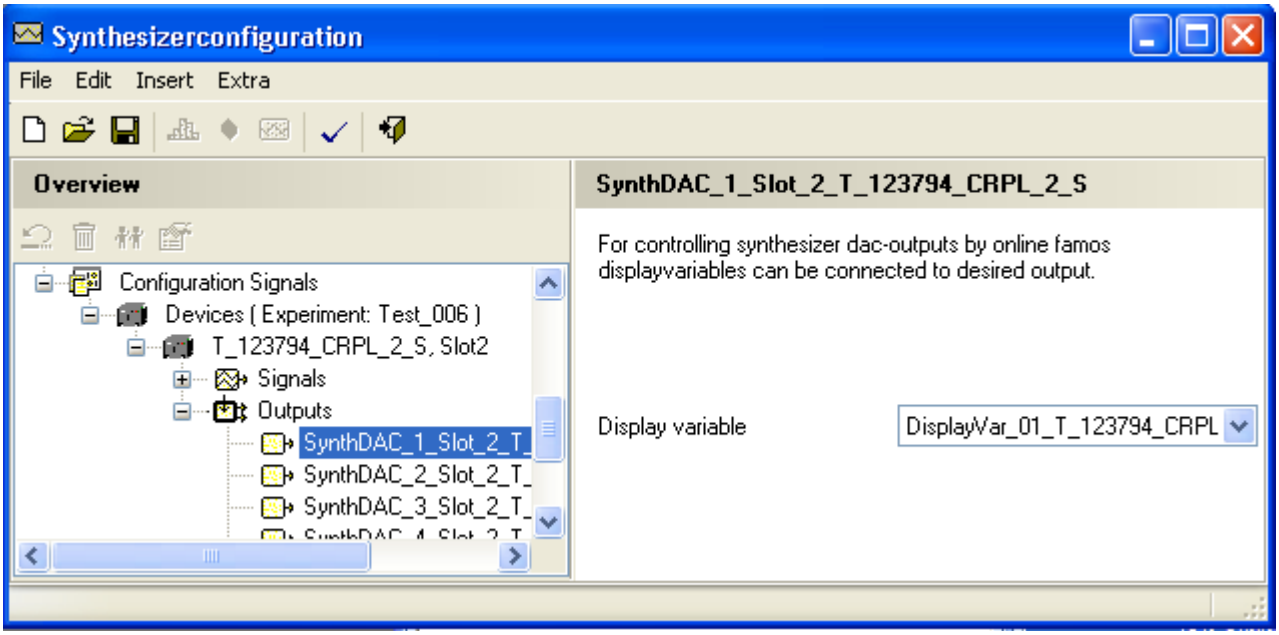
If you select a signal in the tree diagram, an signal editor appears on the right side of the screen.



For details see Signal configuration 48

Selecting a DAC output:

If you select a synthesizer output in the tree diagram, a configurator appears on the right side of the screen, with which a display variable can be assigned to the output. For details see Assigning a display variable to a synthesizer output 76.



4.6 Overview context menu

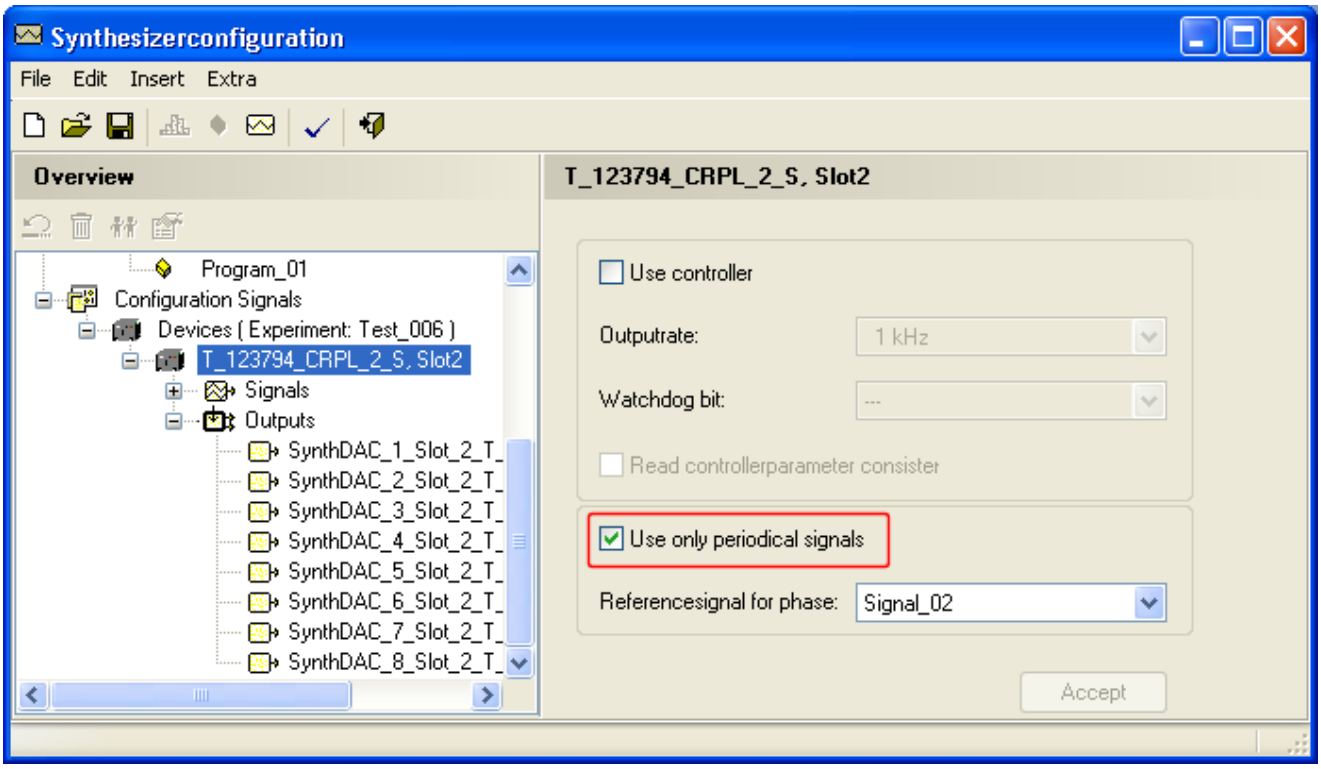
You can open the Overview context menu by right-clicking on a *segment*, a *program* or *signal*. This corresponds to the menu *Edit*. See *Edit* 12 for details.

4.7 Setting the phase during signal output

The phase between multiple signals can be governed by process vector variables.

Activating the option:

To activate phasing adjustment, select the option ***"Use only periodical signals"*** in the configuration dialog for the synthesizer slot.



Additionally, a reference signal must be selected in this dialog, whose phase is zero.

Limitations on the selections for "Use only periodical signals":

All segments must have the same time length. One single process vector variable controls the period length of all signals. It initially has the name pv.<DeviceName>_Slot<SlotNumber>_Period. This name can be edited.

Setting the phase length:

- To set the phase difference to the reference signal, two process vector variables are available.
- A process vector variable determines the number of periods by which the phase is to be reached. It is also possible to set fractions of periods. The name of these variables is comprised as follows: pv.<SignalName>_Slot<SlotNumber>_PhasePeriods.
- The second variable determines the phasing which the signal is to take on. The name is formed in the same way as the other variable and ends with PhaseValue.

Changing the offset, gain and period length

Offset, gain and period length can be changed **during signal output**. For this purpose, there are three process vector variables for each variable:

1. The desired **new value**
2. The **mode** according to which the desired new value is to be approached

3. The **time/rate**.

By default, when a value is changed, it changes abruptly.

When the **time/rate** changes, the value change is **ramp-shaped**. For example, this can serve to gently ramp-up or ramp down the gain.

The names of the process vector variables are defined as follows:

pv.<SignalName> _Slot<Slotnumber> _<Parameter>	Desired new **value**
pv.<SignalName> _Slot<Slotnumber> _<Parameter>VariationMode	**Mode** according to which the value is to be approached. **0**: Approach the value over the **time** specified by pv.*VariationParam. The time is stated in seconds. **1**: Approach the value at the **rate** specified by *pv.*VariationParam*. The rate is stated in 1/s. The default value is zero.
pv.<SignalName> _Slot<Slotnumber> _<Parameter>VariationParam	**Time/Rate** depending on the mode. The value must always be stated as positive. The program independently recognizes whether to increment or decrement, based on comparing the current value with the target value. The default value is zero.

Nomenclature:

Offset:

<Parameter>=Offset. The process vector variables are designated by:
pv.<SignalName>_Slot<Slotnumber>_Offset
pv.<SignalName>_Slot<Slotnumber>_OffsetVariationParam
pv.<SignalName>_Slot<Slotnumber>_OffsetVariationMode

Gain:

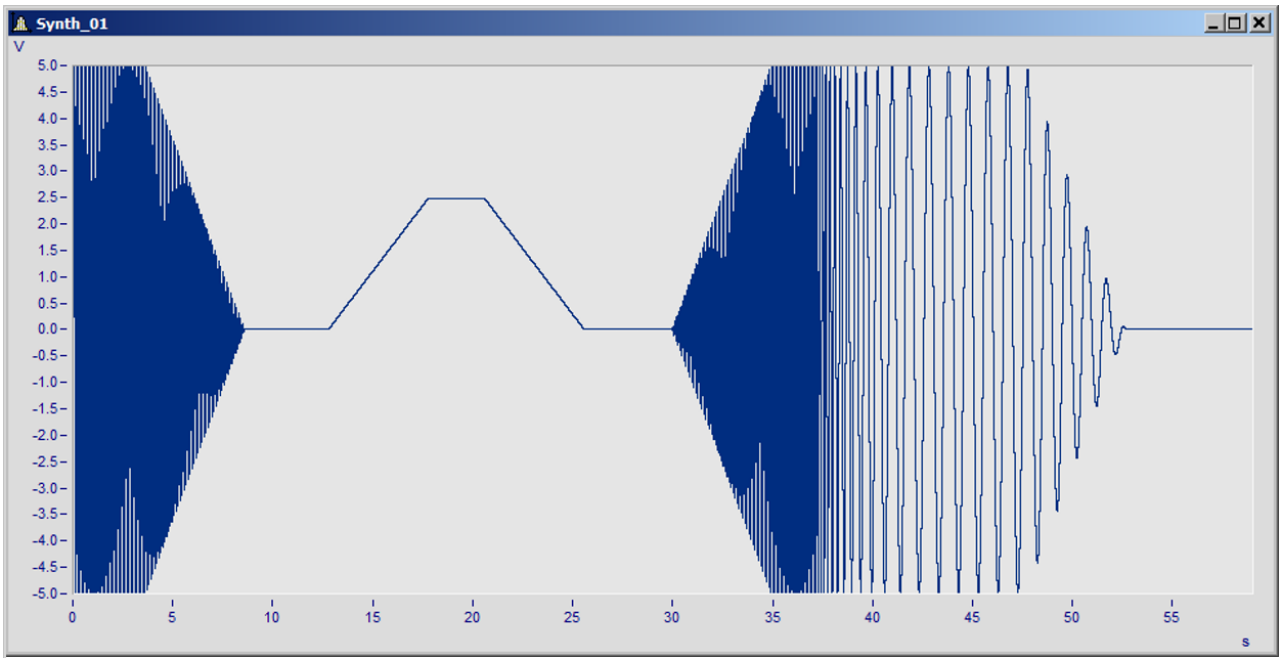
<Parameter>=Factor. The process vector variables are designated by:
pv.<SignalName>_Slot<Slotnumber>_Factor
pv.<SignalName>_Slot<Slotnumber>_ Factor VariationParam
pv.<SignalName>_Slot<Slotnumber>_ Factor VariationMode

Period length:

<Parameter>=Period. The process vector variables are designated by:
pv.<SignalName>_Slot<Slotnumber>_Period
pv.<SignalName>_Slot<Slotnumber>_ Period VariationParam
pv.<SignalName>_Slot<Slotnumber>_ Period VariationMode
On the topic of the variation of the period length, see the notes in the section "Setting the phase 19".

Note

The value is only incremented or decremented at each auxiliary point. Between the auxiliary points, the value remains constant. If this function is to be used, then auxiliary points at equal time spacings should be used, i.e. either XY-data whose time interval is constant, or deltaT/Y data.



4.8 Drag & Drop

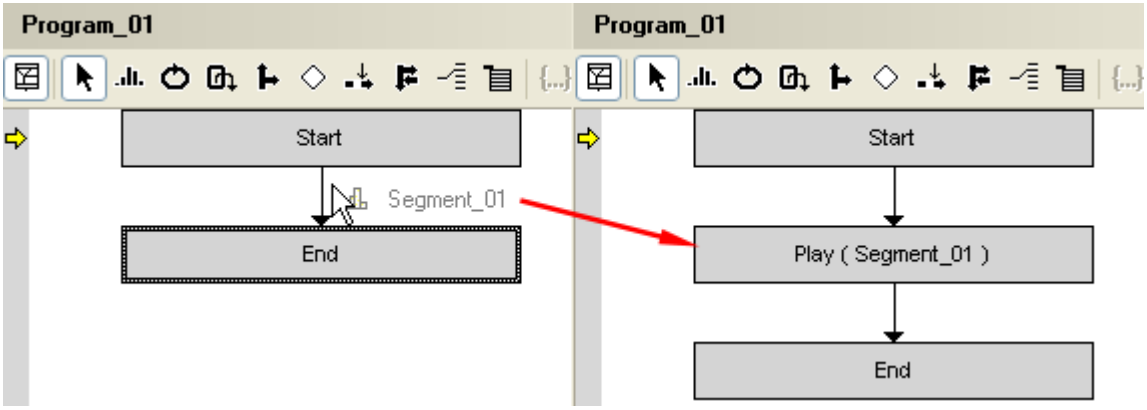
General:

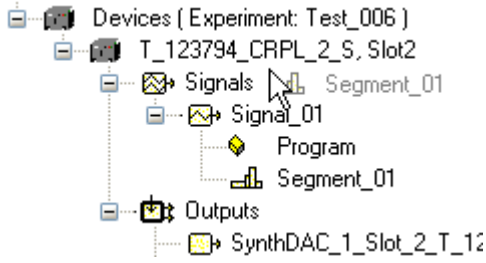
If a normal mouse cursor appears next to the element dragged, it's possible to drop it into the area over which it's located and thus carry out the associated action. On the other hand if the symbol appears, you cannot move the element.

Drag & Drop for segments:

A segment can be dragged and dropped into programs and devices/slots. Details on the actions which are carried out in this way are presented in the subsequent sections.

Using the Drag & Drop technique, segments can be taken from the Overview and inserted into the desired program. The following example shows that a segment "Segment_01" will be inserted into Programm_01. Once the left mouse button is released, the segment is included in the program.





If a signal containing only one segment is to be generated, a new signal can be created by using Drag & Drop to place a segment in one of the Overview's devices:

Once the left mouse button is released, a dialog appears. You can determine in this dialog how often the segment will be repeated. For details, refer to New signal with new segment dialog 24.

Using Drag & Drop in this way you can avoid the indirect approach of first creating a program containing one segment and a while-loop, which is then assigned to a new signal.

Drag & Drop programs:

A new signal can also be created by using Drag & Drop on a program. Once the left mouse button is released, a new signal is automatically created and receives the name "Signal_XX", where XX stands for an incremented number.

Applying Drag & Drop to signals:

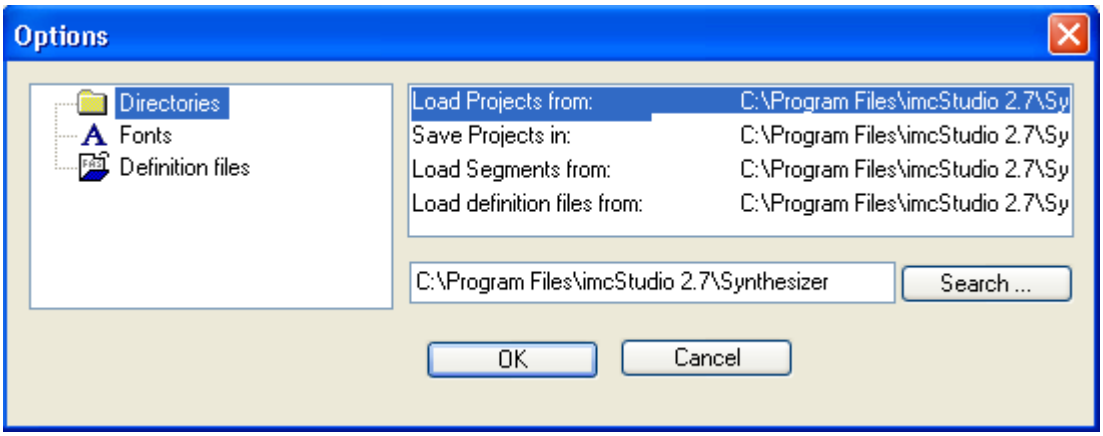
An existing signal can be copied by means of dropping it into another device/slot. (The signal's output terminal must be defined in the "*Connection table*", which is described in Signal configuration 48.) If multiple copies are to be made, the signal receives the name Copy_Y_<signal name>, where Y is an incremented number.

Drag & Drop from the Microsoft Explorer:

Segments from files in FAMOS-format can be placed in a project by applying Drag & Drop to move them from the MS-Explorer to the Overview, as well as using the sequence of menu items *Insert / Segment / Load...*.

4.9 Options dialog

You can configure several global settings, such as saving paths, fonts and the registrations of file import filters with the following dialog. Open the dialog via menu extra:



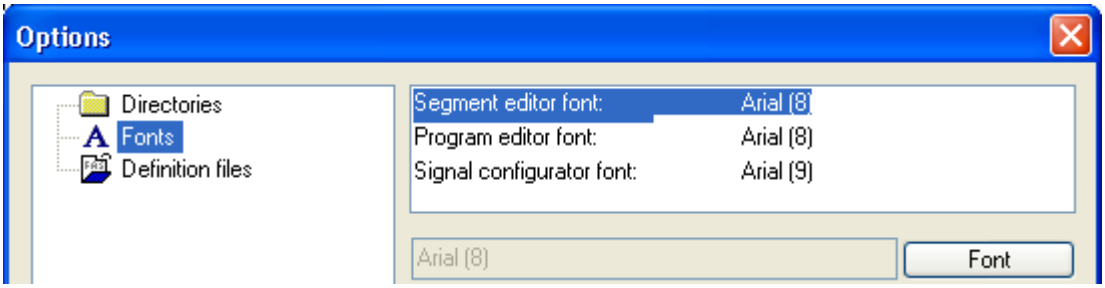
Specifying directories:

Selecting the tree diagram element Directories causes a list of default directories for specification to appear on the right side. When you select a particular line in this list, it's possible to set the associated path in the text box below, or by using the button *Browse*....

- **List line Load projects from**: Projects are usually stored together under a set path. The path specified here is initially suggested whenever you load project files.
- **List line Save projects in**: The path specified here appears as the first suggestion whenever a project is to be saved (refer to Save project 11).
- **List line Load segments from**: Default path for the segment loading dialog.
- **List line Load definition files from**: Default path for the file import filter loading dialog. For details on registering file import filters see Registering import filters 24

Setting fonts:

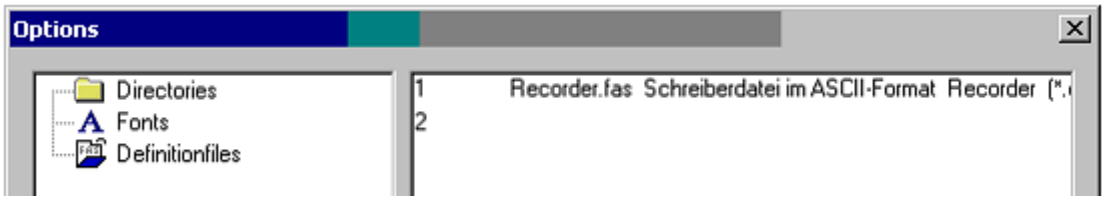
The item *Fonts* shows the fonts for the individual editors:



Clicking on the button *Font* calls the standard Windows dialog for fonts to choose any installed font.

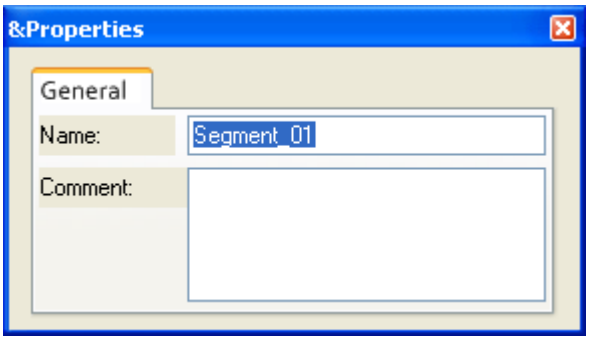
Registering import filters:

- Along with curve files in FAMOS-format, it's also possible to read in files which take other formats. The condition for this is that a file import filter exists (please consider the FAMOS manual for detailed information).
- If a filter does exist, it can be registered in the Synthesizer. Toward this end, the tree element Definition files is selected in the dialog Options. A list will appear on the right side.
- To define an import filter select the first line in the list. By clicking the button Add..., a load-dialog is opened from which an import filter can be selected. Import filters have the file extension FAS.
- When the FAS file is adopted, the file format is registered. Their endings and other information are automatically detected from the file by the Synthesizer software.



- An import filter once added is normally appended to the end of the list. If an existing import filter is to be replaced, select the corresponding line.
- By selecting a file format filter and clicking on Delete, the selection is deleted.

4.10 Properties dialog



The following dialog displays the properties of the selected element e.g. segment, program, signal.

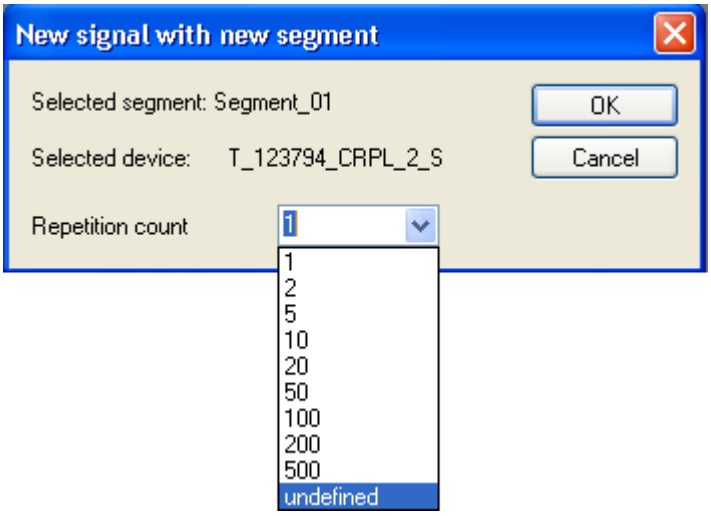
In this dialog, the respective element's name can be changed or a comment can be added to it.

Note:

If a certain name is already in use within a project, it will not be applied once you exit the dialog.

4.11 Dialog new signal with one segment

If you want to create a single-segment signal drag a segment and drop it upon a device/slot. Example: if you intend to output a fixed-frequency sine wave.

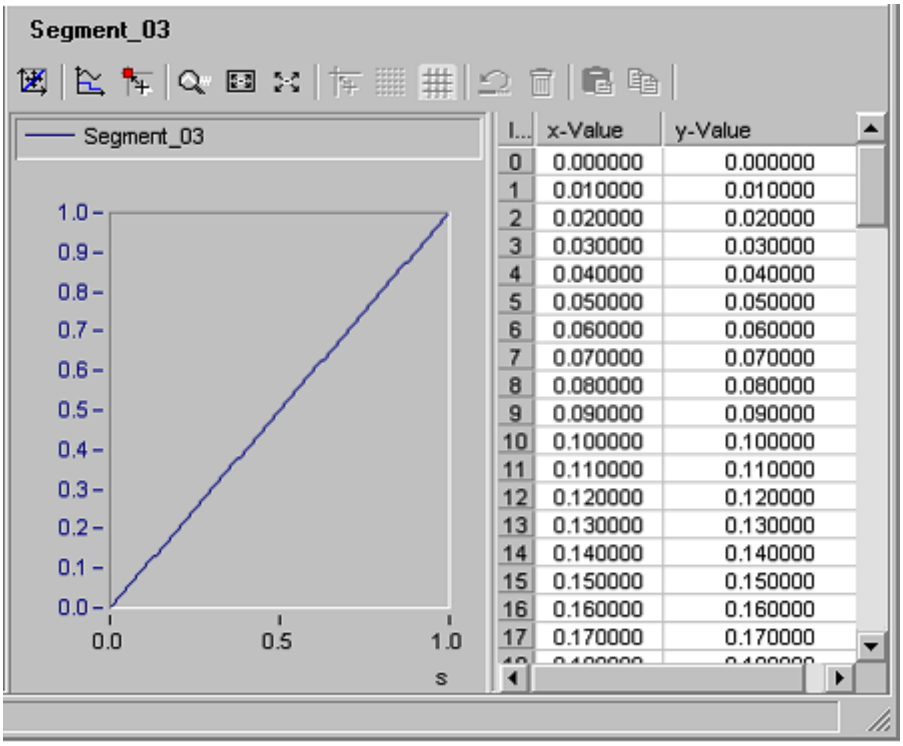


A dialog appears indicating which segment was selected and in which synthesizer board the new signal was created. There is a combobox to specify the repetitions. The choice "undefined" is also available.

Next, use the signal configurator 48 to set at which output the signal is to be outputted.

5 Segment editor

The segment is displayed in the curve window (left side). A list of the x- and y-values is displayed on the right side.



Segment editor in Display mode

5.1 Toolbar buttons

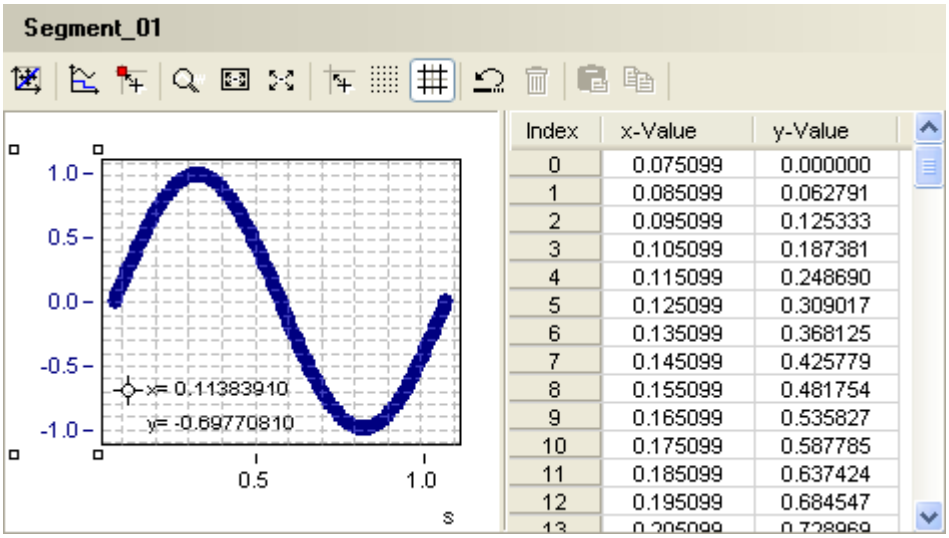
Symbol	Name	Definition
	Curve editor	Switches the curve display between plain display and a curve editor, in which data points can be changed using the mouse.
	Zoom	For zooming in - a part of the curve (only enabled in editor mode.)
	Unzoom	Reverses the zooming by one step
	Overall view	Displays the entire curve
	Snap	The curve points are oriented to a grid; fractions of grid units are rounded up/down
	Raster	Sets the curve editor's grid.
	Grid	Displays grid lines at multiples of the grid spacing.
	Undo	Reverses the last action (paste/delete, changing data points...)
	Delete	Deletes selected data points
	Paste	Inserts selected data points
	Copy	Copies selected data points
	Standard curves	A dialog for inserting standard curves (sine, ramp...) appears; see Standard curves 31
	New point	Adds a new point, e.g. the curve editor is in point-adding mode, so that a new point is generated with every click in the curve window (left mouse button). In order to exit this mode, the button must be hit again.

5.2 Curve window modes

There are two available modes.

- In **Display mode**, the window behaves like a normal imc curve window with all associated functions. An illustration appears above.
- In **Edit mode**, individual data points can be selected by mouse. They can then be moved, deleted, copied and pasted. In Edit mode the entire curve window's background is white.

If you select the curve editor button it is possible to switch between the two modes.



Segment editor in Edit mode

Zoom:

This button change the curve window in Zoom mode, as indicated by this appearance of the mouse pointer: . Use the mouse to drag open a rectangle over the desired range.

Re-zoom:

Restores the curve view's previous zoom-level.

Full size:

The curve window is scaled to provide a view of the complete curve.

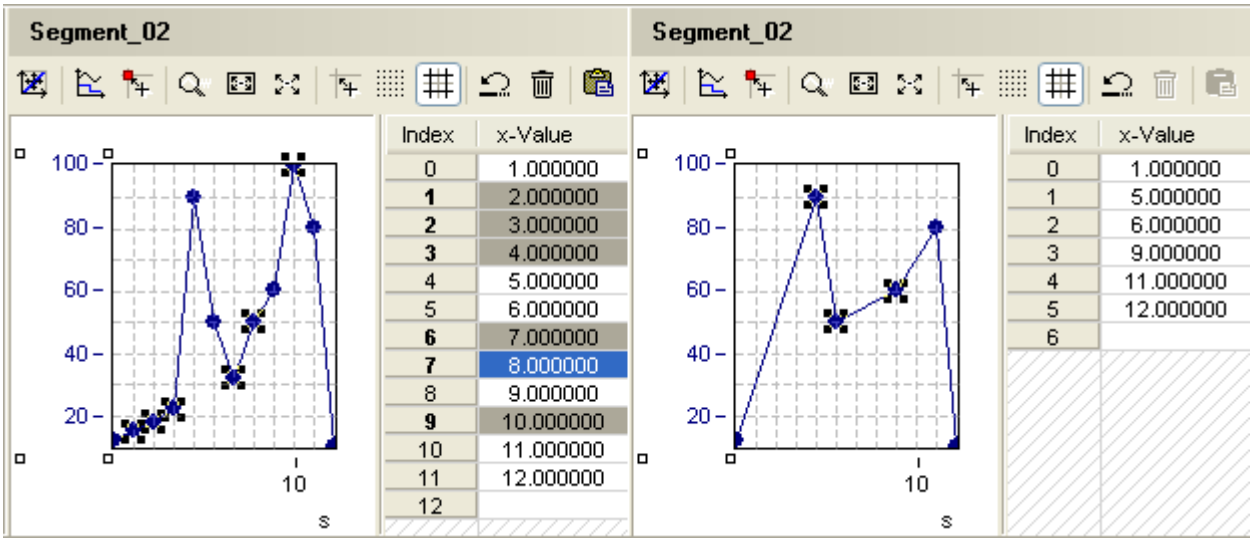
Undo:

Many changes of the curves can be reversed, step-by-step.

The following actions are affected: pasting, deletion and changing of data points. For pasted-in data points, it doesn't matter whether they are newly generated, previously copied from standard curves or from another source.

Delete:

Any point selected can be deleted using this menu item:



Copy:

The selected curve points are copied to the Clipboard. Using Paste, a duplicate of the copied data can be placed in another segment or another location.

The data points are held in the Clipboard as a text table and thus can be pasted into word processing programs (Word, Excel,...).

Paste:

The data in the Clipboard are pasted into a new location. In the process, the segment editor is switched to the Paste mode. This is indicated by this appearance of the mouse pointer: .

- Clicking the left mouse button in the curve window pastes the data points at the pointer's location.
- Clicking the left mouse button in the value table causes the data points to be pasted above the selected line.

Stretch/Compress:

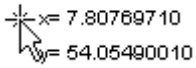
It's possible to change the selected curve points by specifying a factor and offset, for instance, as an easy way to change the amplitude or frequency of a segment's sine signal. If this menu item is selected, a dialog appears in which the factor and offset in x- and y-direction each can be specified. For details, refer to Dialog Stretch/Compress 30

5.3 Adding curve points (at end of list)

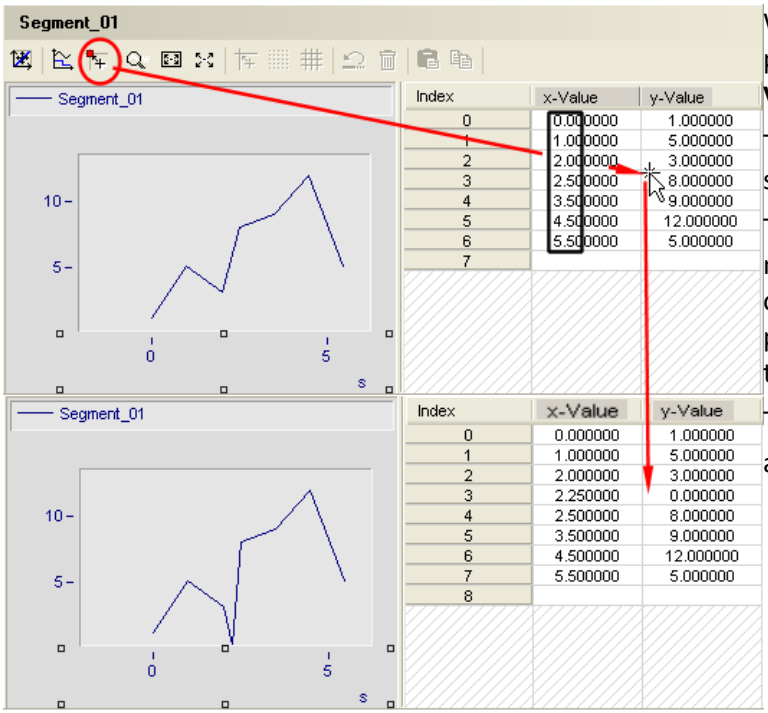
Value list:

- In order to add values, an x- or y-value in the last (empty) line must be selected in the value table. Then, to generate a new point, enter a number into the value table. If a y-value is entered, then the associated x-value is set to the previous x-value +1, as shown in the black rectangle (Figure: Inserting curve points). When an x-value is added, the associated y-value is always inserted with the value 0. both values can still be edited and modified after having been entered. But if the x-value is lower than the x-values appearing before it in the list, then it is incremented by one from the preceding value.
- It is also possible to insert external values into the value table by means of Copy and Paste, e.g. values from existing experiments or even other projects. Observe that the values to be supplied must be expressed with a period before the decimal places, e.g. x-value: 2.45 and y-value: 6.8.

Curve editor:

- The data points to be changed can be selected with the mouse, which is indicated by the appearance of small rectangles around the affected data points. By moving the mouse while the left button is held down, the data point is moved correspondingly within the curve window. However, the movement is limited to the coordinates of a defined grid.
- The button  is used to add new points in the curve editor. The segment editor is then in the *Point adding* mode, as indicated by the appearance of the mouse pointer: 

Each click of the left mouse button in the curve window creates a new data point at its position:



How to change coordinates

When in this mode, it is also possible to add points by clicking the left mouse button in the **Value list**.

The insertion position is always right before the selected row.

The insertion point's default x-value is the midpoint between the x-value of the table row on which you click, and the x-value of the preceding row. The default y-value is zero, since the position is not known.

The Point adding mode can be exited by clicking again on the  button.

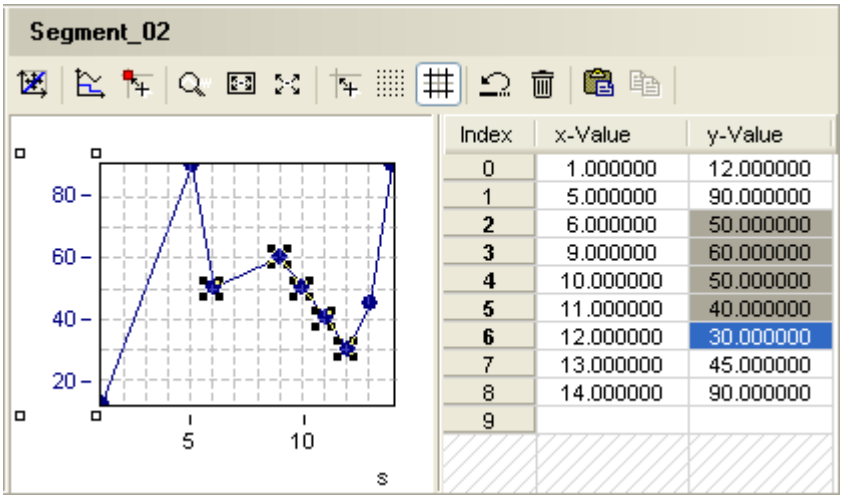
Note

- The point can only be moved by increments corresponding to the grid spacing.
- The change only affects the y-coordinate. It is not possible to move to a position either before or after another point.
- There is always one grid-unit's worth of space between points. How the grid is defined is discussed in Configure grid 30.
- In order to move multiple data points, it is necessary to also keep the CTRL-key held down.

5.4 Selecting curve points

It is often desirable to change multiple curve points at once. For this reason the segment editor allows you to select multiple curve points and to change them simultaneously.

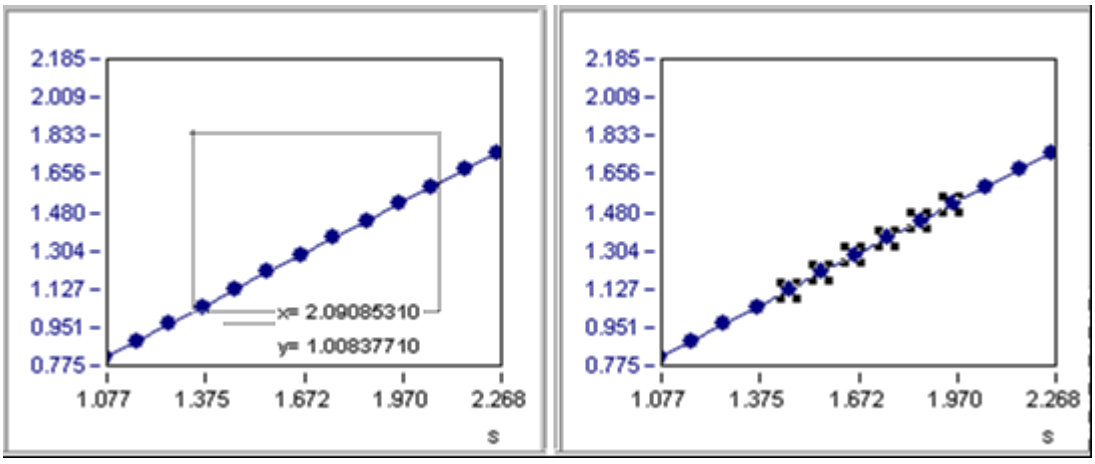
Value list: Individual curve points can be selected by clicking on the corresponding row. Any already selected row will then be de-selected. In the value list, click on one element with the left mouse button and sweep the mouse either upwards or downwards through the list, with the button held down.



Index	x-Value	y-Value
0	1.000000	12.000000
1	5.000000	90.000000
2	6.000000	50.000000
3	9.000000	60.000000
4	10.000000	50.000000
5	11.000000	40.000000
6	12.000000	30.000000
7	13.000000	45.000000
8	14.000000	90.000000
9		

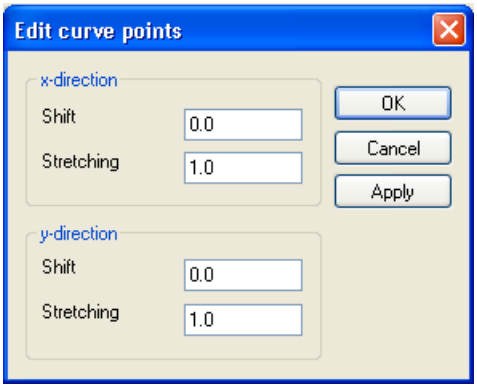
Curve editor: Individual curve points can be selected by clicking on the particular point in the curve window. The selection is highlighted by small squares around the selected data points.

Multiple data points: There are two ways to select multiple data points. Either the mouse can be used to highlight a rectangular segment in the graph, in which the desired points are contained, or it is possible to click on each desired point in succession while holding down the CTRL key. The latter option enables selection of non-contiguous points.



5.5 Dialog Stretch/Compress

It is often desirable to change individual curve points or even the entire curve by multiplying the amplitude or frequency of, for instance, a sine signal by some factor. In this case it is possible to manipulate the distance of point to point with *Stretch* and *Compress*. See following text:



The offset is referred to as a *Shift* and its initial value in each direction is 0. We refer to the factor here as *Stretching*, whose initial value is 1.

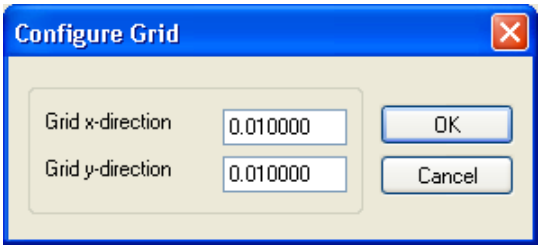
If the *Apply* button is clicked, the selected data points are adapted according to the parameters set. By repeatedly clicking the *Apply* button it's possible to repeatedly alter the points in the same way specified by the parameters.

For instance, if the value for the control *Stretching* for the y-direction is set to 3, and the *Apply* button is clicked twice, then all selected curve points are multiplied with a factor of 9.

5.6 Configure grid dialog

Since manipulation of the data points using the mouse can be imprecise, depending on the screen resolution, the option of using a grid is provided. When the positions of data points are changed, they are changed by whole increments of this grid. If a data point has the initial y-position 0.11 and the grid spacing is 0.1, then this point can only take the position values 0.01 + n*0.1 (thus 0.11, 0.21, ...)

Selecting the menu item *Configure* grid it is calls the following dialog:



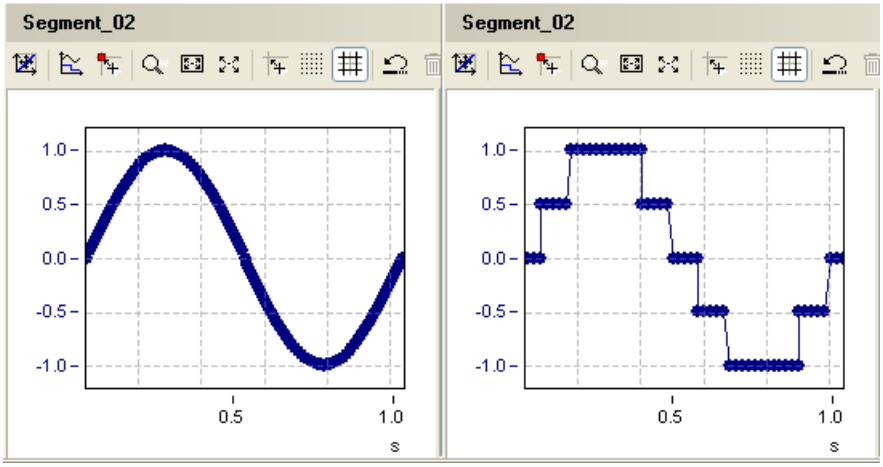
Snap to grid:

This function rounds the curve points' x- and y-values up or down to align them to the grid. One situation where this would be needed is when loading a curve, since the curve points will not always coincide with the established grid.

Attention:

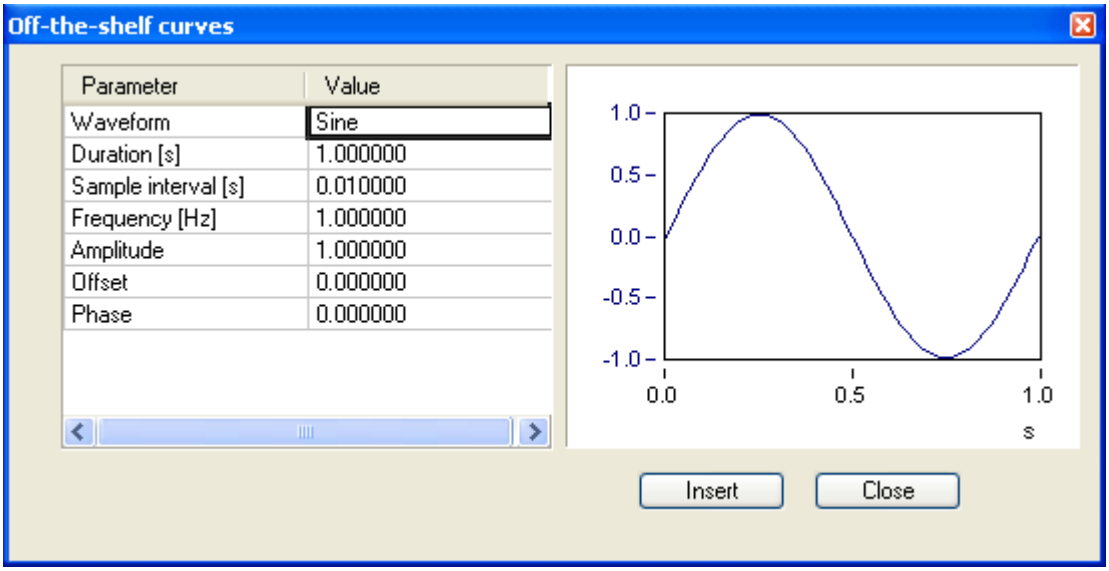
This function should be used with caution if the grid spacing is large, since it can lead to unpredictable results.

The following image shows what effect the function has when the grid spacing is 0.1 in both directions:



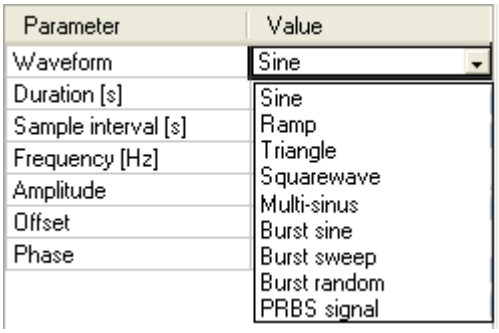
5.7 Standard curves dialog

Are you looking for a standard curve type? For the purpose of generating curves such as sine, sawtooth, triangle waveforms and squarewaves, the *Standard* curves dialog is provided:



The user can specify the curve form, frequency, amount of data points etc.

Curve form:



The following curve forms can be selected.

Duration:

Here, the number of auxiliary points is set in conjunction with the interval between data points:

$$Number\ of\ data\ points = \frac{Duration}{Sample\ interval}$$

Sample interval:

This parameter must be adapted to the time window and frequency selected.

Frequency:

Here, the curve's frequency is adjusted. If only one period is to be generated, the time window must be adapted according to the formula time window = 1/frequency. To obtain a set amount of data points, the data points' interval must also be adjusted.

Amplitude:

This sets the maximum value that the curve can have.

Offset:

A constant value added to those of the curve points.

Phase:

Phase shift for the curve form sine by an angle between 0 and 360°.

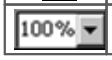
Assigning a segment:

The selected curve can be placed in a segment by the simple means of Drag & Drop. To do this, click on the curve window in the dialog Standard curves and hold the left mouse button down. Then, by moving the mouse it is possible to select the position in the segment editor's curve window or value list, in which to insert the curve. Releasing the mouse button deposits the curve at the desired position.

6 Program editor

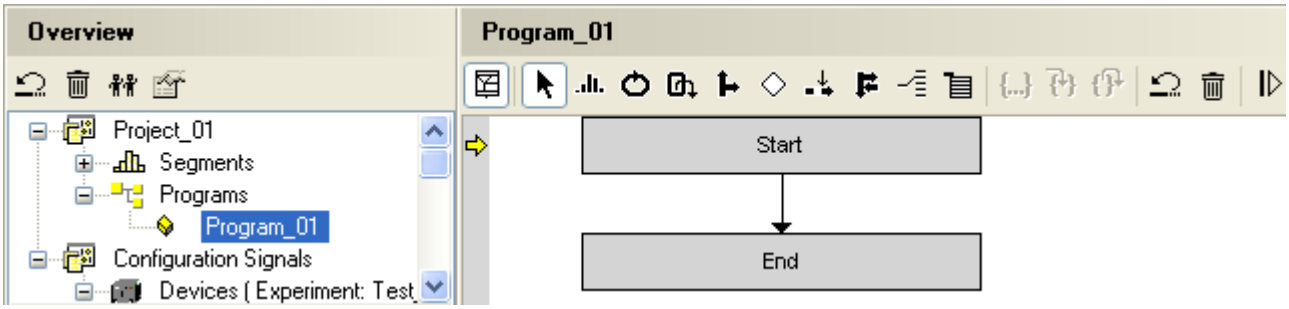
Programs specify in what order and upon what conditions segments are played back. For this purpose, an editor is needed, as described below.

6.1 Toolbar

Symbol	Name	Definition
	Flowchart	Switches between flowchart and program text display
	Edit	In this mode, program steps can be selected, moved or grouped together.
	segment	Inserts a segment
	backward branch	Inserts a backward branching
	Break	Insert break instruction
	forward branch	Inserts a forward branching
	Action	Insert action instruction
	Synchronization	Insert synchronization instruction
	Generic skip	Insert generic skip 37
	Merge program steps	Selected program steps can be grouped to a sub-program, to increase transparency
	Into sub-program	If a sub-program is selected, it can be opened by means of this button and edited.
	Out of sub-program	This button lets you return from a sub-program to the overall program
	Undo	Reverses deletion of program steps
	Delete	Deletes program steps
	Run mode	Simulates a program run
	Procedure mode	Carries out a single step in a simulation. Subprograms are either run completely or skipped.
	Step mode	Carries out a single step in simulating a run. If a sub-program starts, the simulation proceeds to each single sub-program step, as well.
	Debug window	Displays a curve window in which the output is simulated.
	Magnify	Magnifies the flowchart view.
	Shrink	Shrinks the flowchart view.
	Zoom	Set view of the flowchart per listbox

6.2 Empty program

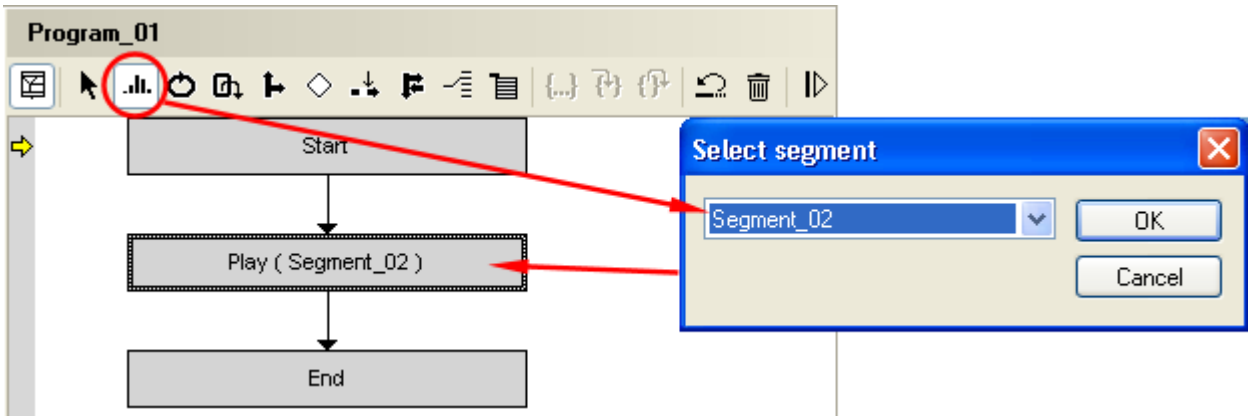
Once a program has been created by means of the menu item *Insert / Program*, it consists of only a Start-block and an End-block. Clicking on the program's tree diagram opens the following window:



In between, it is possible to place segments and forward or backward branches.

6.3 Inserting a segment

Segments defined in the project are inserted using this button: from the toolbar of the program editor. The mouse pointer goes into the mode Insert segment: . When you select a line in the flowchart, a dialog appears in which the desired segment can be selected:

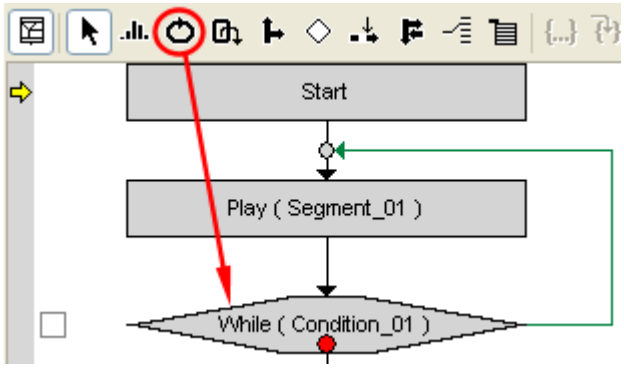


After selecting a segment from the combobox, it is inserted into the desired position.

Then the program editor returns to the normal Edit mode.

Another technique for inserting a segment, by means of Drag & Drop, is described in Drag & Drop 22.

6.4 Inserting a forward/backward branch

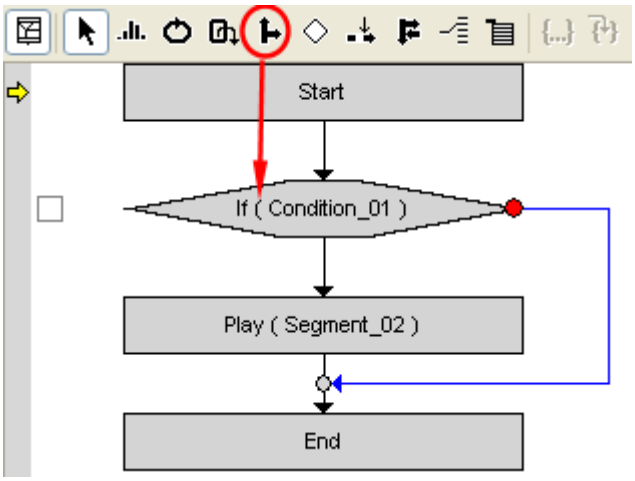


When this button is clicked, the program editor change in the special mode *Insert backward branching*. This is indicated by a new appearance of the mouse pointer:

If you select with the left mouse button on a line in the flowchart, the backward branch is inserted at exactly this spot (here: clicking on a point before the segment):

This actually inserts two program steps, one in which the branching occurs and the other which marks the backward branch's entry point.

The program step will only be integrated if you select the check box.



This button leads the program editor into Insert *forward branching* mode. The mouse pointer changes its appearance to

By clicking on a line in the flowchart, a forward branch is inserted at the corresponding point (here: before the segment):

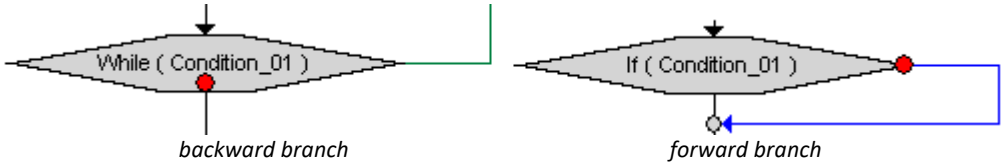
Afterward, the program editor returns to the normal Edit mode.

The entry marker can be moved using the mouse.

This name can be edited, as described in Changing a segment to be used, changing names 41.

Terms:

The term **branching** denotes the two symbols below:



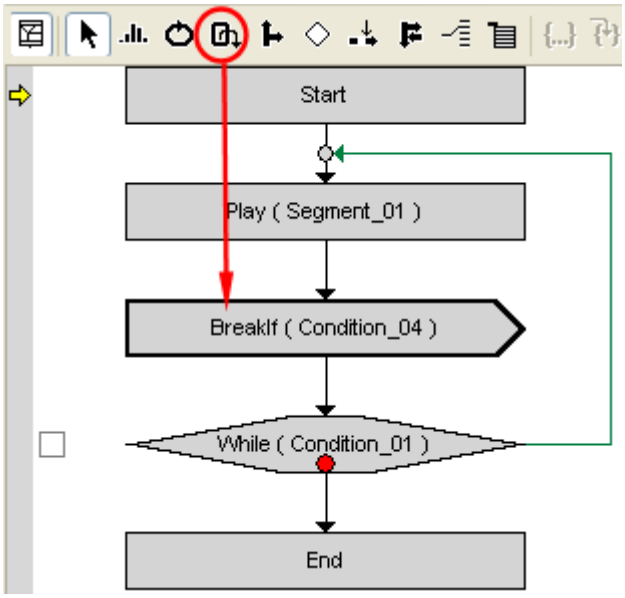
- entry marker:
- Depending on whether a certain condition is set, the program proceeds either to the entry marker or to the next program step.
- Depending on where the entry marker points, you can generate either a **forward** or a **backward** branch. This means: if you insert the entry marker before the program step, you obtain a **backward branch** which is called a **While** loop. Conversely, if the entry marker is located after the program step, it is a **forward branch** which is called an **If** loop.

Note

- The forward- and the backward branch should not cross each other. This means, you can neither select a forward nor a backward branch on the same program step, if there are several program steps in between a branch. This is only possible by inserting a generic skip 37.
- The branching point contains a condition. This condition initially gets a default name constructed from the string "Condition_" plus an incremented number. This name can be edited, as described in Changing a segment to be used 41.
- The entry marker as well as the branching can be moved using the mouse. For details refer to Inserting a generic skip 37

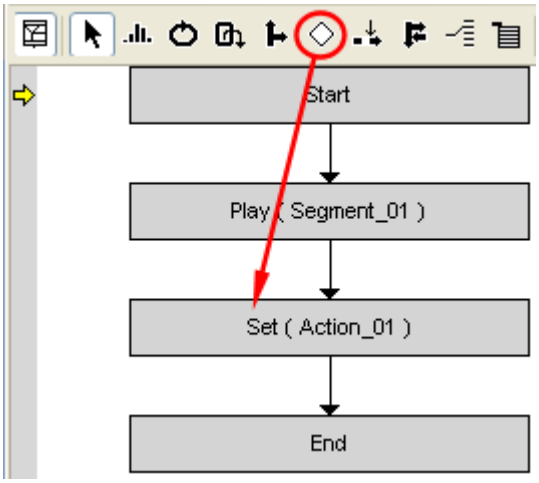
6.5 Inserting a break condition

A break condition can only be inserted into a While-loop. You will leave the loop with this break condition.



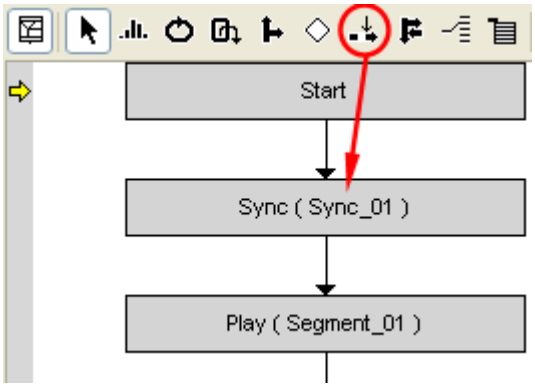
6.6 Inserting an action element

An action element can be used to set bits or display variables whose states or values are needed later in the program.



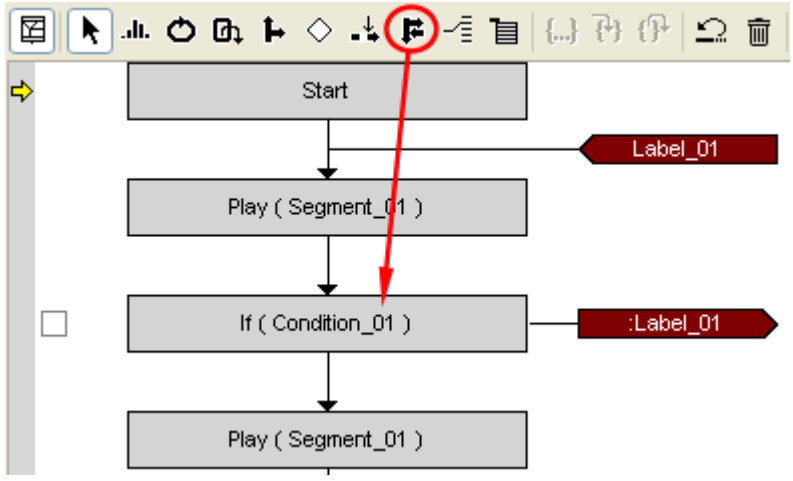
6.7 Inserting a synchronization element

Synchronization elements are only needed if multiple signals are outputted. The element ensures that segments are started simultaneously. For instance, suppose a test rig is to synchronize the control signal for a velocity with a load signal, and that the velocity must first be run up to a certain level. An extra segment is needed for the run-up phase. Only once the desired velocity has been reached, the velocity signal is to start together with the load signal. This is accomplished by providing a synchronization element for each signal; one before the velocity signal's segment and another before the load's segment.



6.8 Inserting a generic skip

A generic skip can skip to any location within the program, similarly to a "Goto" instruction. The skip is carried out depending on a condition. By this means, for example, output of a signal can be stopped.

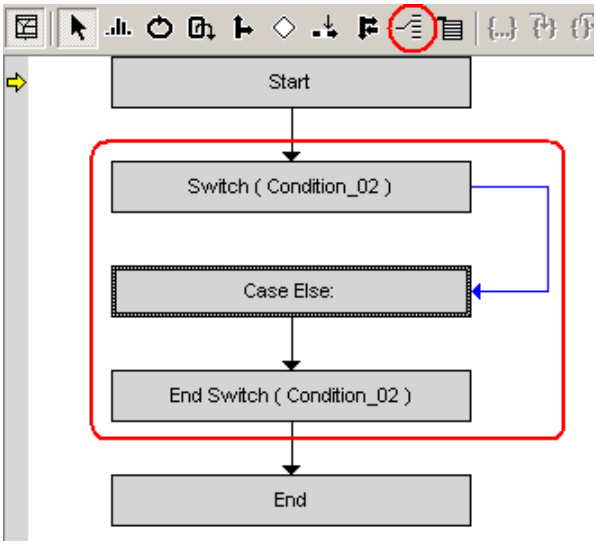


6.9 Inserting a switch and case command

	Switch	Insert Switch command
	Case	Insert Case command

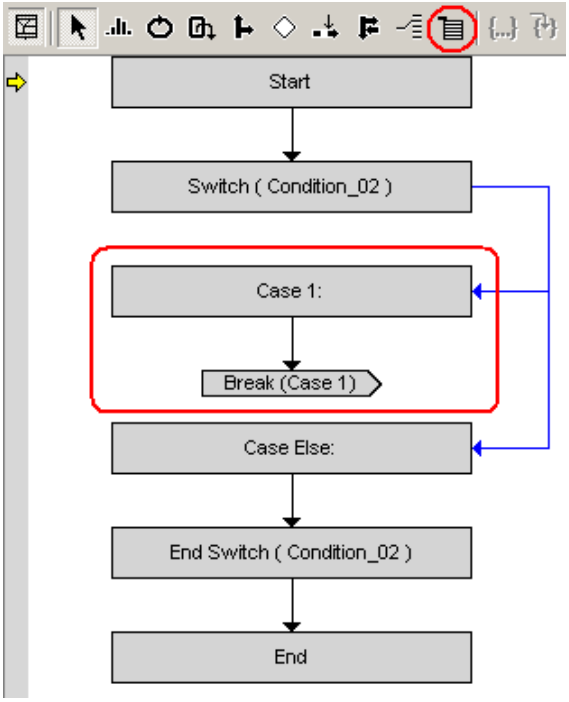
Inserting a switch command:

A Switch command makes it possible to use a variable (e.g. Display variable) to skip between different program segments or routines. For this purpose, the command must first be inserted. Within this command there are associated Case commands. The Case branches in turn contain the segments or program routines. Depending on the value of the Switch command's variables, individual "Cases" will be crossed.

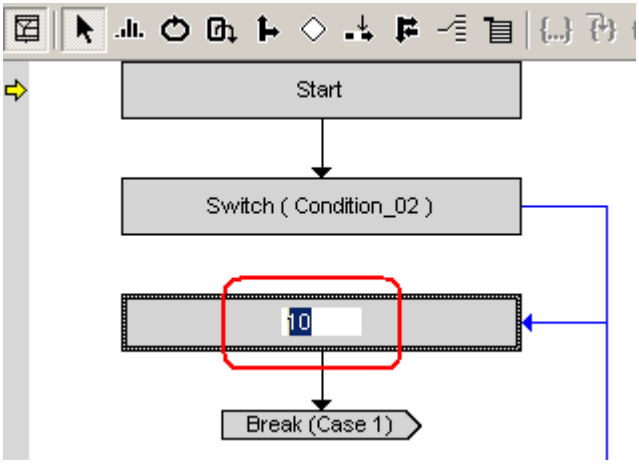


Inserting a Case command:

The Case command determines what to do when the Switch command's variable takes a particular value. Case commands can only be inserted into Switch commands.

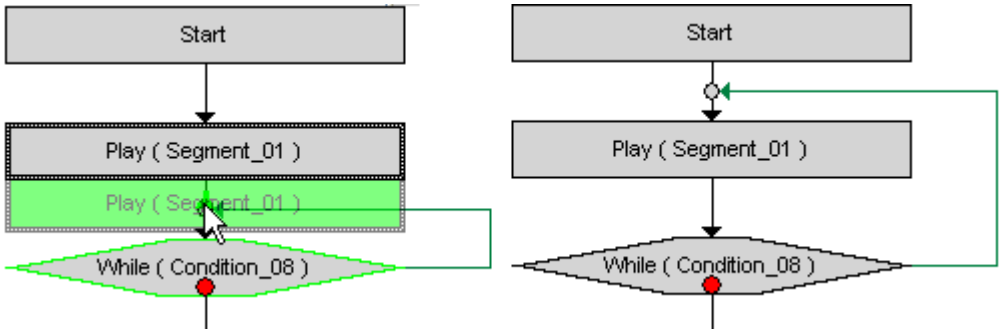


The value can be set after double-clicking on the Case command:



6.10 Moving program instructions

You can move a program-elements by Drag & Drop, for example the segment "Segment_01" is relocated within the backward branch:



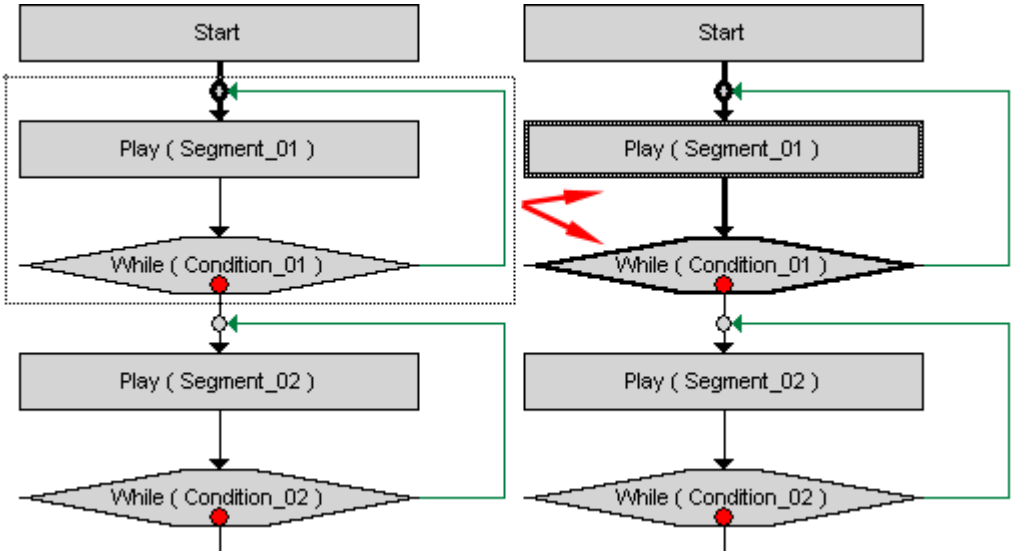
There are certain limits to the ability to move program elements:

- An entry marker and a branching point may not be inserted in such a way that its branch crosses an existing branch. This is indicated by the cursor changing its appearance to  .
- The marker for a backward branch may not be inserted after the corresponding branching point, and conversely, the branching point may not be inserted before the corresponding entry point.
- The same goes analogously for forward branching, where the branching point must naturally precede the entry point.

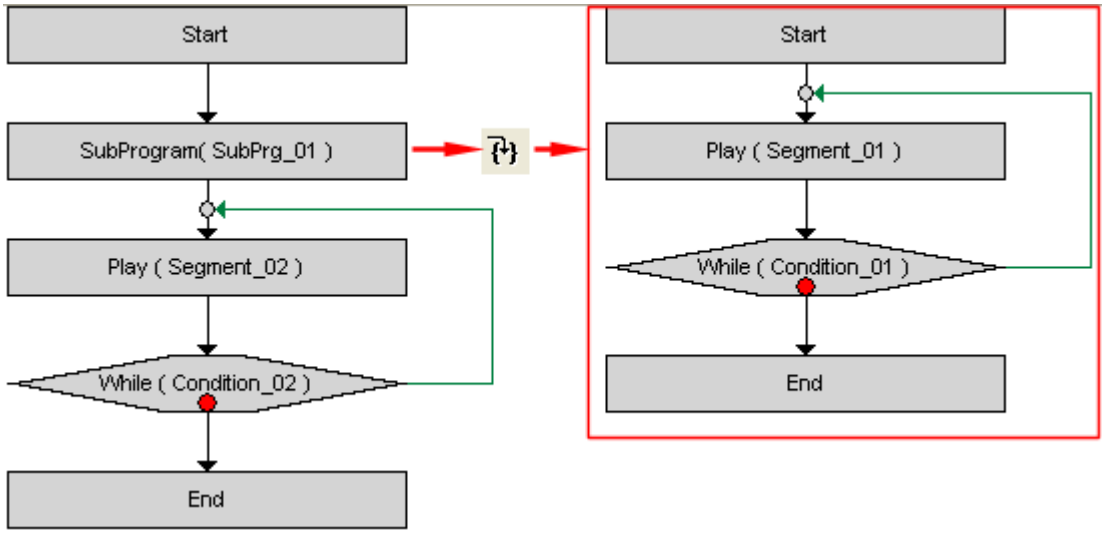
6.11 Merge program instructions to sub-programs

It is sometimes convenient to group multiple program steps to build a block. This block is then a sub-program. The rule is that both the branching point and the entry point, whether of a forward or backward branch, must be included in the sub-program. However, this is automatically taken into account when the program steps are selected.

For starters, a number of program steps must be selected. Use the mouse to draw a rectangle around the desired steps.



Once the desired program steps have been selected, the button Merge program steps ( ) can be used to join them to a sub-program. It contains a standard name, composed of the string "SubPrg_" plus an incremented number:

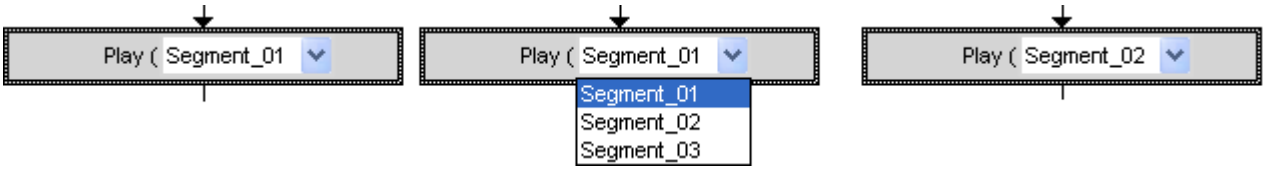


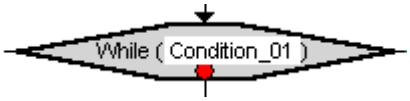
Program with sub-program *Sub-program of the main program*

By selecting the sub-program and using this button: , the sub-program becomes visible in the Editor and can be edited like the main program. This button: returns to the higher-level program. Sub-programs can include subordinate programs. Recursive operations, however, are not possible.

6.12 Changing a segment to be used, changing names

Segments which are played back in response to the command Play can be altered. To do this, click the mouse on the instruction, which causes a combobox to appear:





The names of sub-programs and conditions can be changed. To do this, double click on the appropriate instruction to select it. If you have done this the name can be changed. If chosen name already exists for a different sub-program or a different condition, an error message appears.

The entered name is adopted when you click *Enter* or select a different instruction.

6.13 Debugging programs

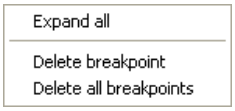
In order to ensure that your program is working correctly, it is necessary to debug errors. Before starting your measurement device it is possible to display the entire program in the debug mode. Possible errors in the program can be eliminated.

On the left side of the flowchart is a grey bar. Along this bar, a yellow arrow denotes at which step the program is currently located. There is a debug program that enable the possibility to display the **single steps** of the program with e.g. functions of segments.

In order to quickly reach the location in question, you can set **breakpoints**. Breakpoints are denoted by a red circle: . When the program is run, the debugger stops at these points. Then it's possible to run the program further step-by-step.

6.13.1 Setting / deleting breakpoints

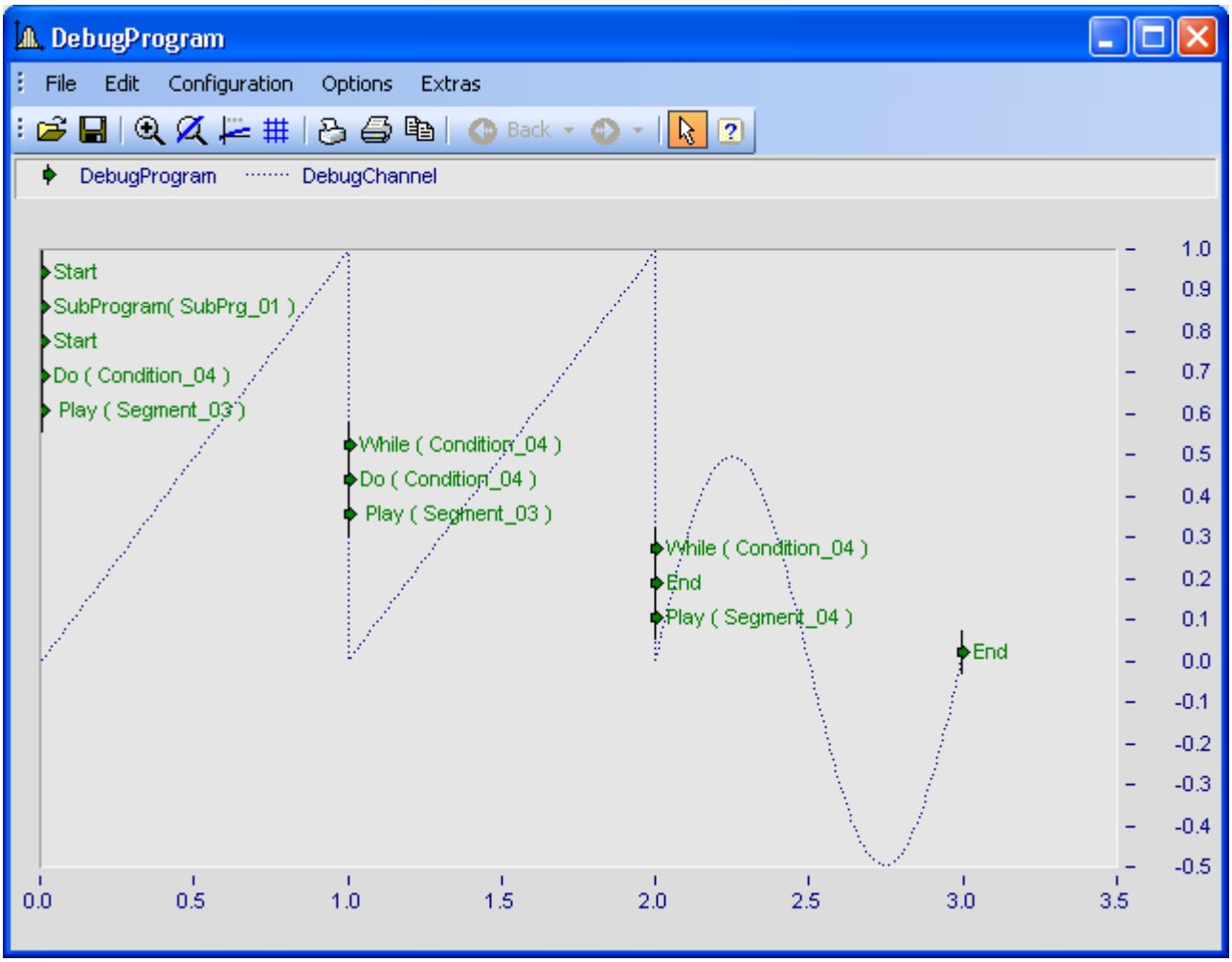
Set / Delete a breakpoint by right-clicking on the grey bar appearing at the left margin.This calls a context menu.



The command *Insert breakpoint* inserts a breakpoint at the current location, as indicated by the addition of the symbol .

6.13.2 Debug curve window

Whenever a debug procedure is started, a curve window displaying a simulation of the output appears. Additionally, the associated program steps are written out at their respective positions.



Here, a ramp and a sine are played in sequence repeatedly. The ramp appears in a sub-program.

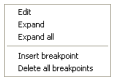
6.13.3 Debugging in Run mode, Procedure mode, Step mode

There are a few different techniques for debugging a program.

- This button:  , for Run mode, causes the entire program to be run. When a breakpoint is reached, the run is stopped and the system will wait until one of the Debug buttons:  ,  or  are pressed.
- In *Procedure mode*  , the program is always run one step at a time. In this context, sub-programs are treated as a single procedure step and run in their entirety, unless there is a breakpoint within the sub-program.
- For a detailed test, the *Step mode*  is available. Here, the debugger enters the sub-programs and performs each step there one-by-one.

6.14 Subprogram context menu

Right-clicking the mouse above a subprogram calls this context menu:



Below, the menu's individual functions are described.

Edit:

Edit corresponds to the button , Into sub-program. The subprogram is displayed in the editor and can be edited.

Expand:

By grouping multiple program steps together, a subprogram is created. *Expand* reverses this action, which means that all program steps belonging to the subprogram are returned to the level of the overall program.

Expand all:

All program steps belonging to all subprograms are returned to the level of the overall program.

Insert breakpoint:

This menu item is only enabled if there is no breakpoint yet at the selected location. For details on the functioning of this item, see Setting / deleting breakpoints 41

Delete breakpoint, Delete all breakpoints:

These items provide ways to remove one or all breakpoints. The former item is only enabled if a breakpoint exists at the selected location. For details on the functioning of this item, see Setting / deleting breakpoints 41.

7 Managing the segment files

Segments can be changed during the measurement if the option ***Reload*** 54 is activated in the sub-dialog Parameter 50.

This option is only supported by imc CRONOS devices as of the serial number 14000.

Changing files

In order to move from one file to another, use ***pv-variable <SignalName*** 8 ***>_Slot<Number>_NextFile***.

The segment files are designated in the device storage by the filenames fileindex0, fileindex1, etc. In order to play back the segment fileindex3 next in Slot 2, for instance, the pv-variable pv.signalname_Slot.2_NextFile must be set as = 3.

The condition for this is that the file with the index 3 had previously been copied for the signal to the device by means of imc STUDIO. Initially, the Synthesizer outputs the current segment to its end, even if the pv-variable has been set to 3 in the meantime. At the end of the running segment, the system switches to the new file having Index 3.

The running file is displayed with the ***pv-variable <SignalName>_Slot<Number>_CurrentFile***.

Synchronized switching among multiple signals

If multiple signals' files are to be exchanged and the exchange is to be synchronized, a special command pv-variable is used:

Initially, the command ***17*** is written to the signal's command pv-variable (***pv.<SignalName>_Slot<Number>_Command***). In this way, this signal is added to a "synchronization pool".

The signals in the "synchronization pool" only switch to the next file once all signals have been reloaded.

In case of repeated synchronized switching, the command 17 must be applied to the respective signals each time.

Note

- If the files are of different lengths or if the signals in the files are the same length but at different positions, synchronized switching may not be possible.
- Make sure process vector variables have been activated.

Removing from the synchronization pool

A signal is removed from the synchronization pool using the command 18. This makes immediate switching possible once more.

List of commands:

1 Start signal output

2 Stop signal output

3 Pause

4 Reset or start of signal output after pause

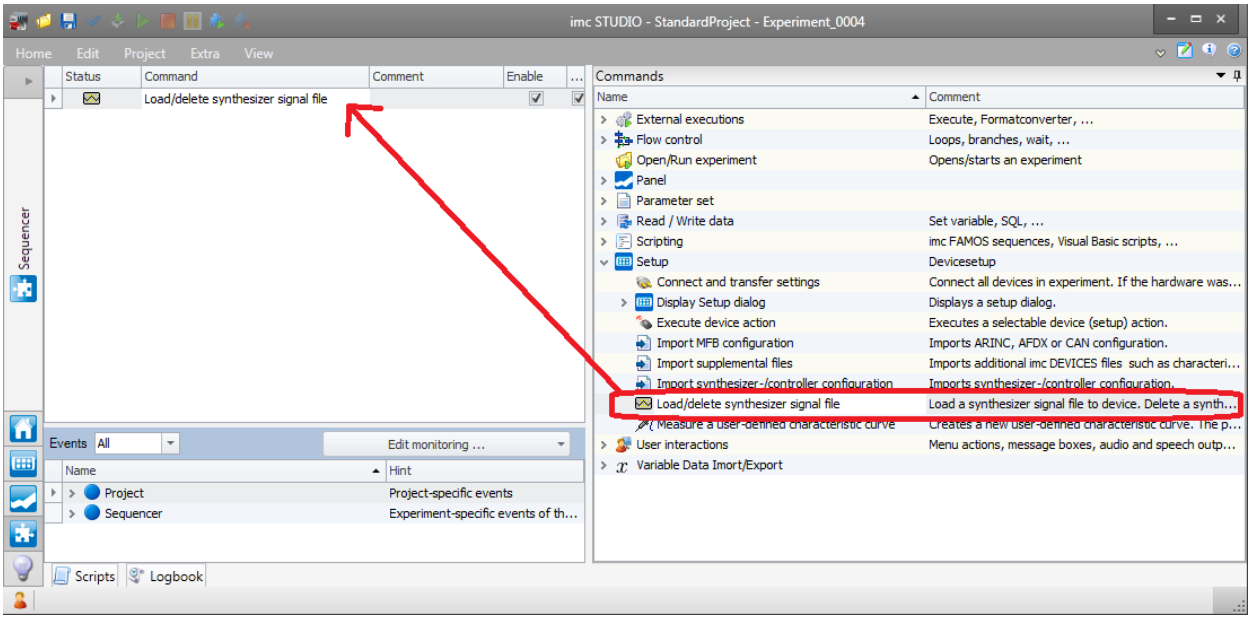
17 Add signal into "synchronization pool"

18 Remove signal from "synchronization pool"

7.1 Writing files to the device with imc STUDIO

Segment files are written to the device with imc STUDIO as of version 4.0.

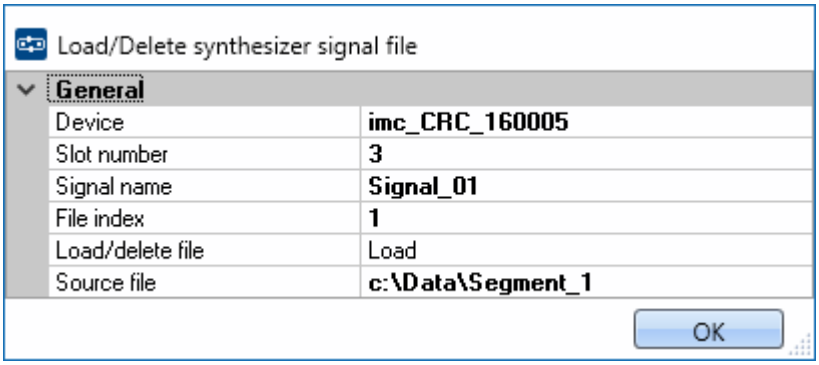
Toward this end, the command "*Load/Delete a Synthesizer signal file*" is provided. The command can be called either by means of a widget (e.d. button) or from the STUDIO Sequencer.



Command to transfer a segment file from the PC to the device disk

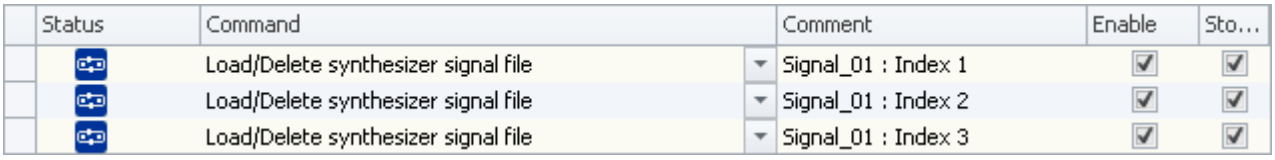
In the Properties you select the Synthesizer output. The outputs are distinguished by their respective slot numbers and signal names.

Parameter	Description
Device	Here, select the target device. The device requires a Synthesizer-module
Slot number	Synthesizer slot in the device
Signal name	See the Synthesizer-configuration for the signal name.
File index	A signal's segments can be changed during the measurement. By means of the parameter, the segment is assigned an index in reference to which it is accessed. The index indicates the number of the segment which is selected by means of the pv-variable 44 <SignalName>_Slot<Number>_NextFile. For an detailed description, see the section "*Managing the segment files*" in the "*Synthesizer Manual*".
Load/delete file	• **Load:** Transfers the source file to the device • **Delete:** Deletes the file having the index specified from the device
Source file	Here, select the file you wish to import.



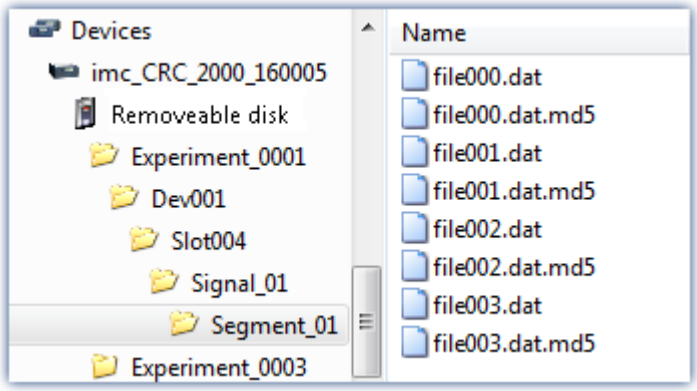
Assignment of the segment to the synthesizer output

It is thus possible to transfer multiple segments in succession:



Transfer of three segments

The segments are set up in the device's data carrier as follows:

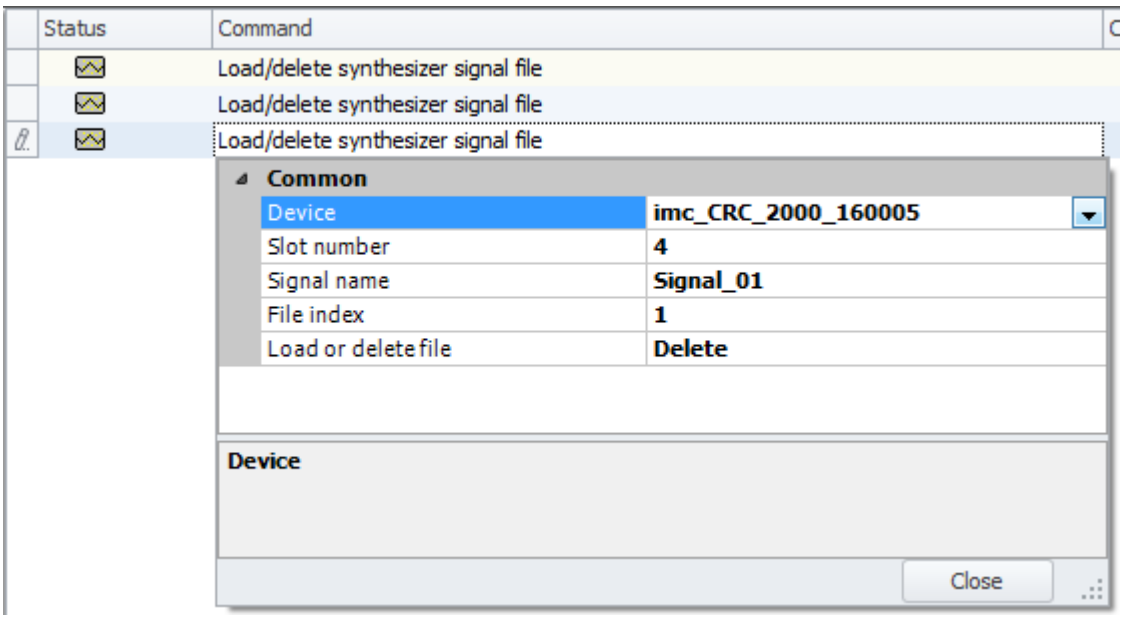


File structure of segments on the device disk

Note

The data are saved on the device drive in a special format. It is **not** possible to copy the data manually!

Segments can be deleted with the same command if "*Delete*" is entered under "*Load or delete file*".



Delete a segment from the device disk

8 Signal configurator

You can determine the setting of a signal [48] with the **Signal configurator**. A signal is formed by a program which determines the order in which the segments are to be outputted.

In the process of making such a program, conditions are positioned within it, which initially only represent placeholders, but which must be defined to depend on the value of particular bits, or to be based on iterative conditions by the time the signal is outputted. It is also necessary to specify at which of the synthesizer board's DAC outputs the signal is to be outputted.

The necessary assignments are accomplished by means of the connection table as described in greater detail in The Connection table window [54].

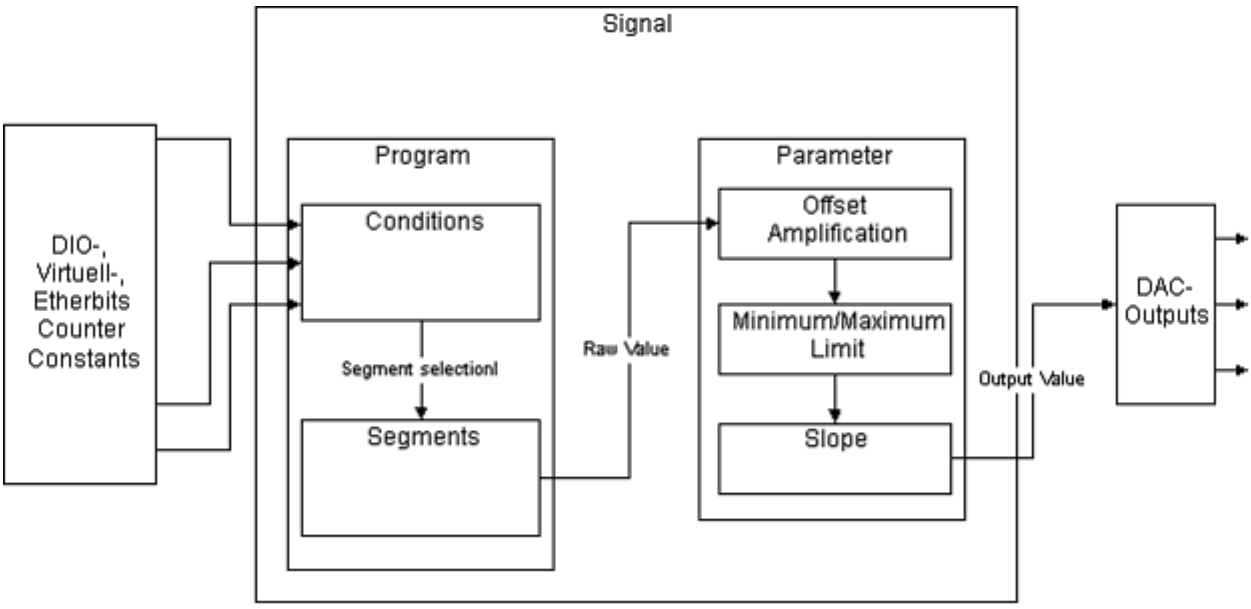
Other necessary parameters to be specified include the minimum and maximum values, maximum slope, and the voltage balance in certain situations such as the values before and after the signal output. These parameters are set in the sub-dialog Parameter [50].

The sub-dialogs Conditions Simulator and Output are provided for debugging purposes. Conditions Simulator enables individual DIO-, virtual- and Ether-bits to be set. In the dialog Output, the overall result which would appear at the output is displayed.

8.1 Structure of a signal

A signal's structure is governed by its program. A segment to be outputted is determined on the basis of conditions stated in the program. While the segment is being played, there is no checking of the conditions. Once the segment has been outputted, the system finds the next segment according to the sequence of program steps and the conditions they specify.

The values contained in a segment are outputted sequentially at the specified interval. These segment values are raw data processed by the signal's parameters in order to reflect particular scaling and safety aspects. The basic structure of a signal and how output values are generated is shown in the following diagram.



8.2 Sub-dialog Program

Here you can see an illustration of a program sequence.

Clicking on a branch with a condition highlights the corresponding logical input in the Connection table, a window on the right side of the signal configurator. Conversely, selecting a control input in the Connection table highlights the corresponding branching point on the flowchart:

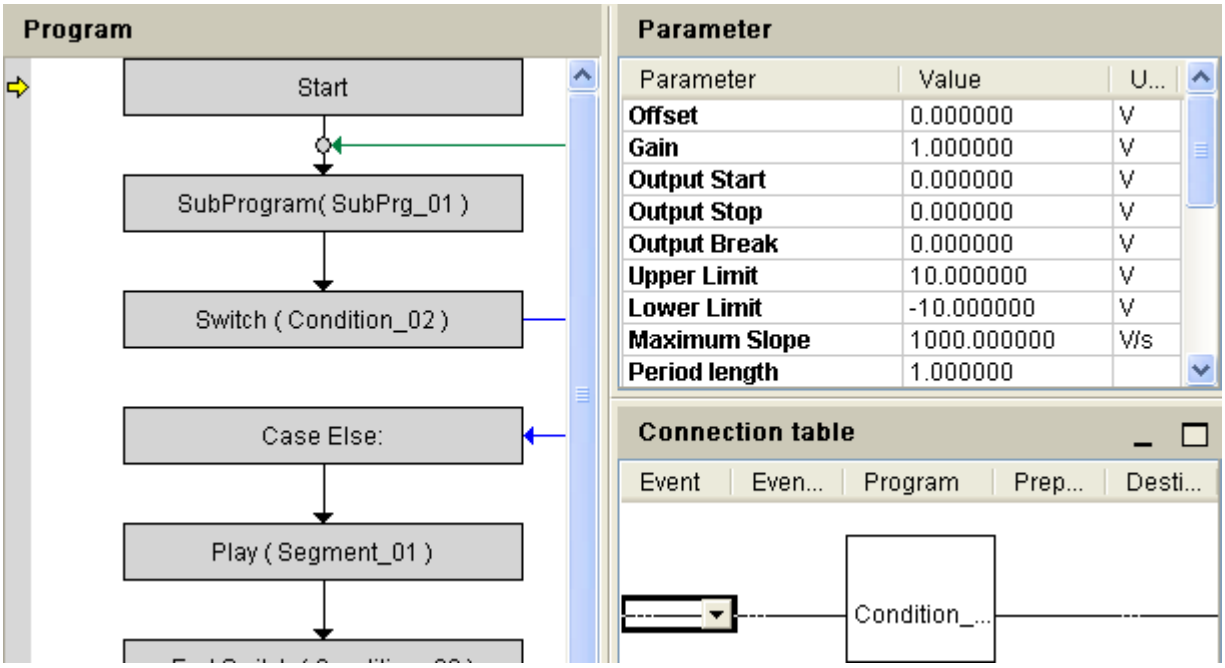


Illustration of program sequence, Parameter and Connection table

Every single part is described in the following chapters: The Parameter sub-dialog 50; The Connection table window 54; Program Editor 33

8.3 Sub-dialog Parameter

These parameters affect the output directly: the data stream created with the segments can be globally rescaled or have an offset added to it, or its frequency can be changed, etc.

The individual parameters are each described in a section below.

The window Program 49

Parameter

Parameter	Value	Unit
Offset	0.000000	V
Gain	1.000000	V
Output Start	0.000000	V
Output Stop	0.000000	V
Output Break	0.000000	V
Upper Limit	10.000000	V
Lower Limit	-10.000000	V
Maximum Slope	1000.000000	V/s
Period length	1.000000	
Compression	0.000000	
Interpolation	S-Curve	-
Ratio Line to S-Curve	0.000000	-
Scaling	1.000000	V/V
Unit	V	-
Display	Output value	
Offset control	pv.Signal_01_Slot2_Of...	
Gain control	pv.Signal_01_Slot2_Fa...	
Period length control	pv.Signal_01_Slot2_P...	
Reload	none	

Part of illustration in chapter The window Program

Offset:

This *offset* is added to all segments. Depending on the value of the parameter Display, this offset can be specified as a voltage value or, with an appropriate scaling factor, as a physical value.

Gain:

The raw value is multiplied by a factor specified as the Gain. Thus, all segments are globally multiplied with this factor.

Output start:

Signals can be linked to triggers. Output of a signal or segment can be programmed to begin only upon fulfillment of a particular trigger condition. The signal output's state until the triggering moment is defined by this parameter. It is set when an experiment is prepared and remains constant until the trigger is released.

Output stop:

The signal output's state after the program has run all the way through.

Output break:

The signal output can be interrupted and later continued from the point of interruption. In the meantime, the Synthesizer outputs this value.

Transition time change of state ("State transition time"?):

Transition time between states, e.g. from the stopped state until the first segment value.

Transition time greater than 0: When the status changes, there is a linear transition within this specified time to the next value.

Example: If the signal is stopped, the synthesizer transitions smoothly within this time frame from the last outputted value to the specified stopping value. Upon re-starting, the synthesizer proceeds to the first segment value to output within this time frame.

Transition time equal to 0: The maximum signal slope is used for the status transition.

Upper / Lower limit:

This parameter sets an upper and a lower limit for the voltage output, which the signal cannot exceed.

Maximum slope:

This parameter sets a maximum slope for voltage output, which the signal cannot exceed. If the raw data have a stronger slope, then the output curve proceeds in the direction of the raw data at the maximum slope specified here. As a result, if the raw data constantly exceed the maximum slope, the output curve may deviate strongly from the raw data. If no output values become visible, it could be that the specified maximum slope is too weak.

Period length:

The data points are separated by a constant time interval. This parameter multiplies this interval by a factor. Thus, the signal output can be accelerated or slowed down.

If unexpected results occur, e.g. triangles instead of a sinusoidal signal, it may be that the value for the parameter "maximum edge steepness" was set too high. In that case, the parameter's value should be adjusted accordingly.

Ensure that the maximum aggregate output rate is not exceeded.

The aggregate output rate is the sum of the output rates of all of the signals used.

A signal's output rate is calculated as follows:

The smallest time distance between consecutive points among all segments in the signal is determined.

This point distance is to be multiplied by the period length.

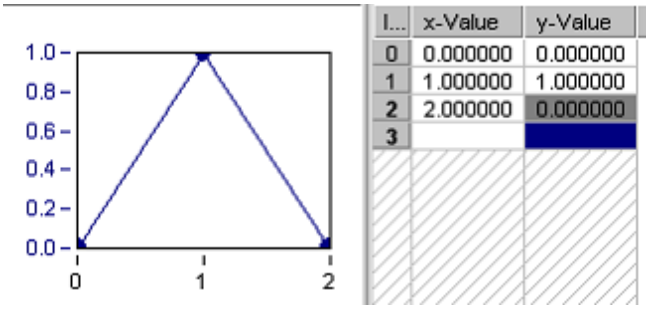
The output rate is the reciprocal of this value.

Compression:

In order to reduce the amount of data to be transferred, the technique Transitional Recording (see manual imc STUDIO chapter imc Online FAMOS) is applied to the segment data to a degree specified by the value of this parameter. Zero means no reduction, and 1 means strong reduction.

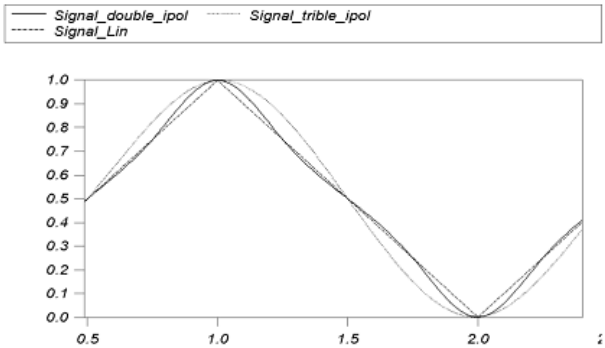
Interpolation:

Normally, data points are connected by straight lines. Here, it's possible to construct a smoother signal by means of splines. In the graphs below, you can see the raw data versus the results:



I...	x-Value	y-Value
0	0.000000	0.000000
1	1.000000	1.000000
2	2.000000	0.000000
3		

Source segment

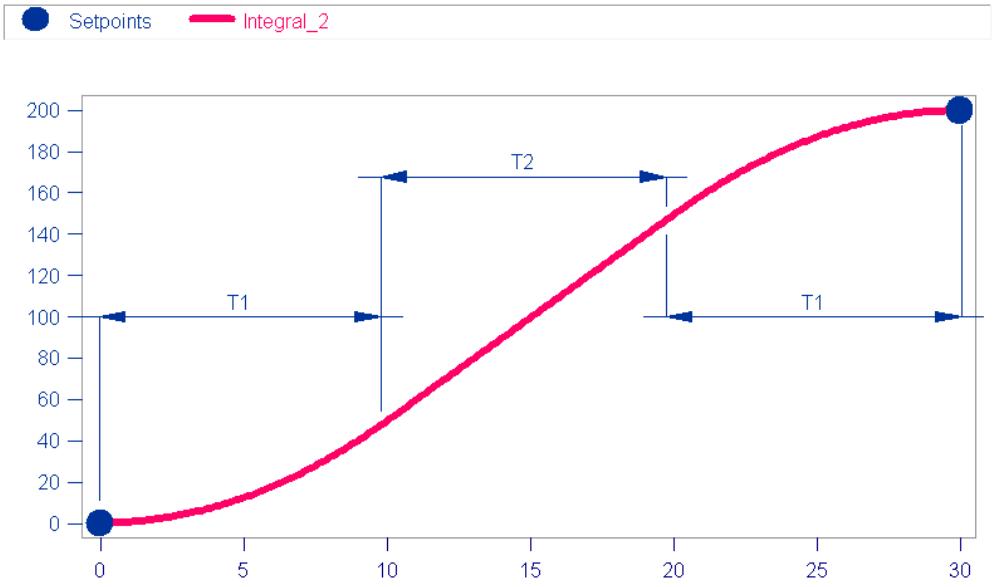


Output using different interpolation algorithms
linear; double-integral; triple-integral

S-Curve:

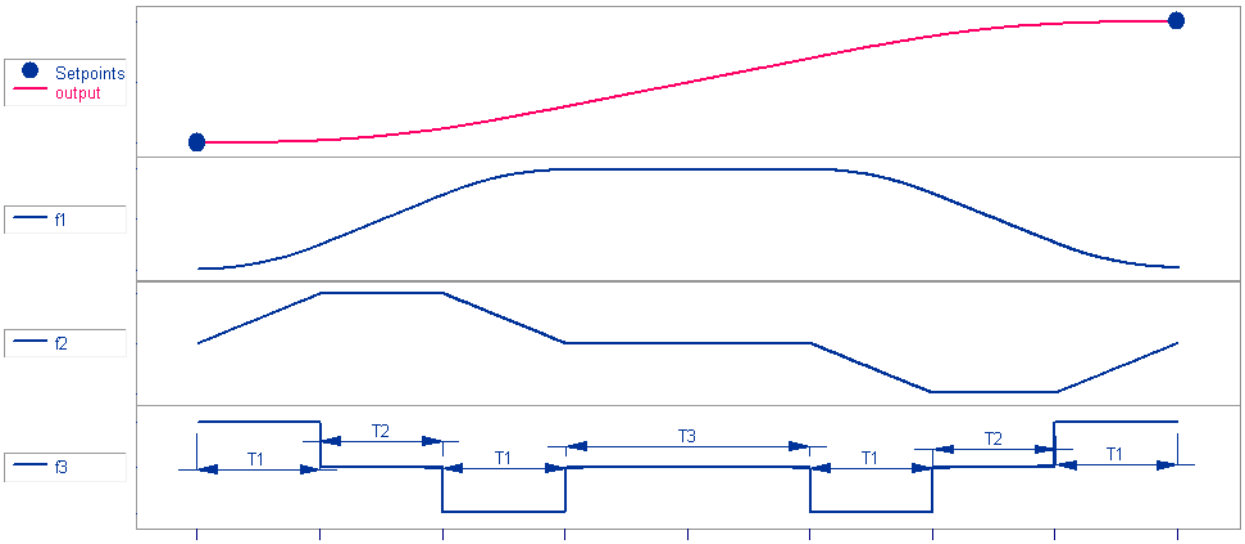
With the S-Curve setting, it is possible to specify a signal by means of inflection points. The segment should be created in FAMOS and adopted as a time-stamped data set. The Synthesizer calculates the specified curve in such a way that the individual data points always begin and end with 0 slope.

- ***S-Curve (double integral):*** The straight-line section (T2) is specified for the setpoints under ***Ratio: linear part to total time***, which appears when the *S-Curve* is selected.



Double integral with setpoints as input

- ***S-Curve (triple integral):*** With the triple integral, as well, the control ***Ratio: linear part to total time*** appears. In addition, there is the parameter ***Ratio: parabolic part to total time***.



Reference

See also Use holdtime for S-Curves 69.

Scaling:

The Synthesizer board could be used to control test stations. For instance, the outputted voltage could drive an hydraulic cylinder. In such cases, there is usually a fixed proportionality factor. This factor can be entered here, so that in conjunction with the parameters Unit and Display it can be used to automate the conversion between physical quantities and voltages.

Unit:

Specifies the physical unit corresponding to the outputted voltage.

Display:

This parameter enables the user to select the display type. The available selection Output value signifies that the displayed results reflect the original voltage values. Unit, conversely, means that the quantities displayed reflect the scaling factor which the user can specify in the parameter Scaling. The list's third column states the applicable unit.

Offset control, Gain control, Period length control:

- During output of the signal, the offset and gain can be changed. Control is then exercised by means of a link with pv-variables. After experiment preparation, the pv-variable concerned is set to the specified initial value. Afterward, the variable can be changed in the Panel widgets with imc STUDIO or using imc Online FAMOS.
- If it is not necessary to control theses parameters, set the value to "---". In this case, no process vector variable is created.

Note When changing the period length, observe these rules:

1. The period length's value may never be zero.
2. The period lengths of all signals may never be so small that the maximum aggregate output rate is exceeded. The calculation of the aggregate output rate is described along with the parameter "Period length".
3. If the signal output plot has an unexpected shape, for instance if a small period length causes a sine signal to produce a triangular signal of small amplitude, then the cause may be that the selected maximum edge steepness may be too small. In that case, adjust the value according.

Reload:

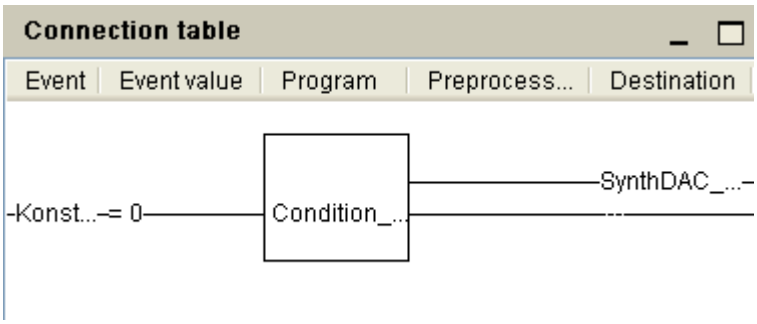
Large segments can be saved to the measurement device's hard drive at the start of the measurement. By means of the imc COM interface or imc STUDIO, it is possible to upload additional segments during measurement.

This option is only supported by imc CRONOS devices as of the serial number 14000.

- ***complete:*** For supplemental uploading via the imc COM interface. The Synthesizer only plays back the next segment in the routine once it arrives at the end of a segment buffer. If the new segment is not yet **completely** loaded, the old segment is played back again.
  This setting is particularly suitable for the exchange of short, periodic outputs, for example for switching from a sine wave to a square-wave .
- ***streaming:*** Also for the purpose of reloading via the imc COM interface. With this option, very large files are transferred as a segment. Output begins already while the segment is still being transferred to the device. If the transfer can not keep pace with the output, the respective last output value is kept until additional data have been loaded.
- ***Hard disk (HD):*** For supplemental uploading using commands in imc STUDIO. The data are saved to the internal device hard drive.
- ***Removable disk:*** Also for supplemental uploading by means of imc STUDIO. The segments are saved to the removable disk.

8.4 Sub-dialog Connection table

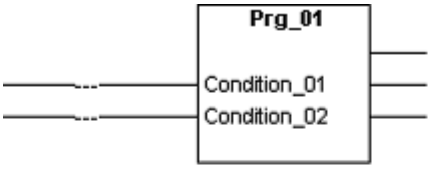
In this dialog, the placeholders for the program's conditions are filled with real bits, counters or constants. You can also set at which of the Synthesizer's outputs the signal is to be outputted.



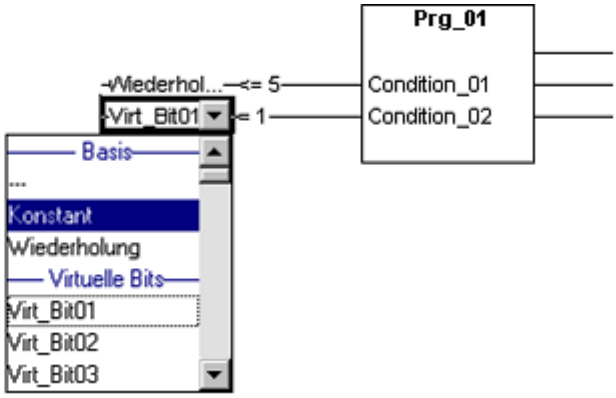
8.4.1 Setting conditions

The controls for setting conditions are on the left side of the Connection table. "Conditions" refers to inputs which affect the signal output.

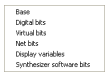
At first any new defined signal has no condition. This is denoted by the "—" symbol.



Selection of a condition



Opening the selection box

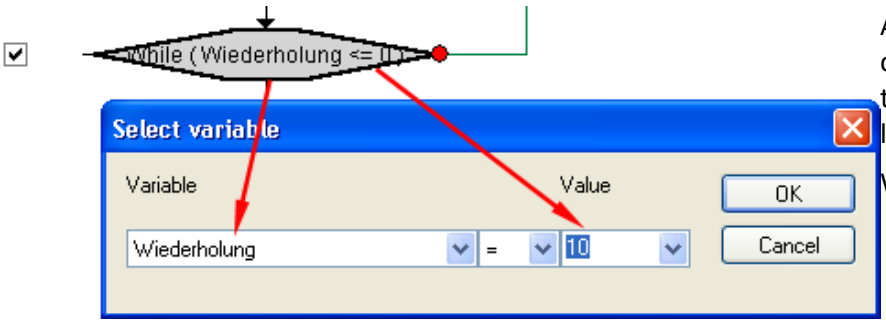


In this list box, a variety of condition types are available. Depending on the type, an event value may also need to be specified. The conditions selected are ordered in groups according to their manner of working. Right-clicking over the list box calls a context menu which enables to skip from the top of one group to the top of another.

With *DIO-*, *virtual* and *Etherbits*, you can specify whether the condition is true when the bit is set or not. Correspondingly, the event value can be set to "1" or "0", respectively.

If a Display variable is selected, a number must be specified as the event value. The condition is true if the display variable equals this value.

It is also possible to define a counter. In this case, select *Counter* from the list box. Here you must specify a number of repetitions as the event value. In order to skip over branchings or to carry out unlimited repetitions, select list option *Constant*.

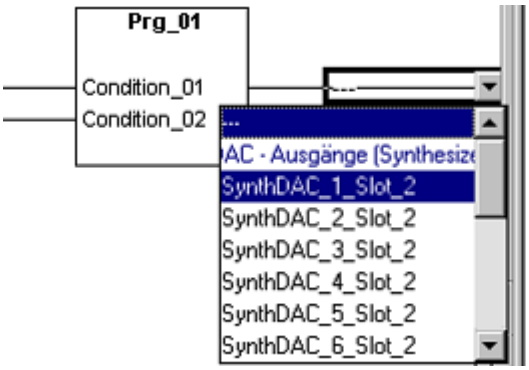


Alternatively, double-click on the condition in the flowchart and enter the condition in the dialog depicted at left.

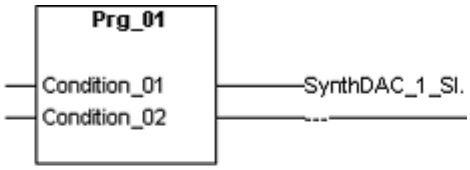
Wiederholung = repititions

8.4.2 Specifying the output

The user must specify from which of the synthesizer's outputs the signals values are to be outputted. The target output on the right side exists for this purpose. By clicking on the target output denoted by "---" and selecting one of the available choices from the pop down box, a new synthesizer output can be added. By selecting an already defined output, a pop-down box for it appears, from which a different selection can be chosen.



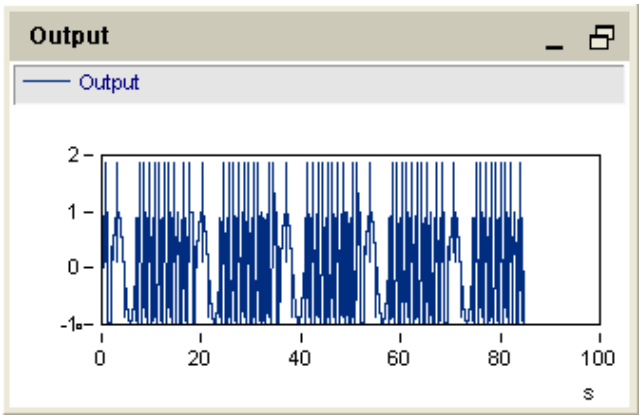
Selection of the target output and opening the pop-down selection list



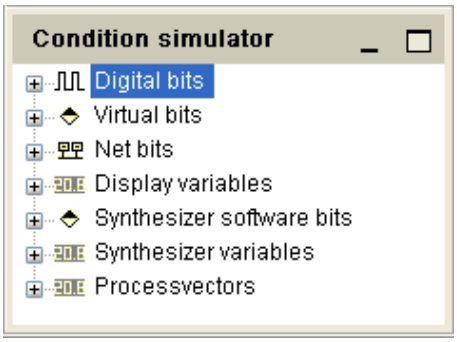
Adding synthesizer Output 1

8.5 Sub-dialog Output

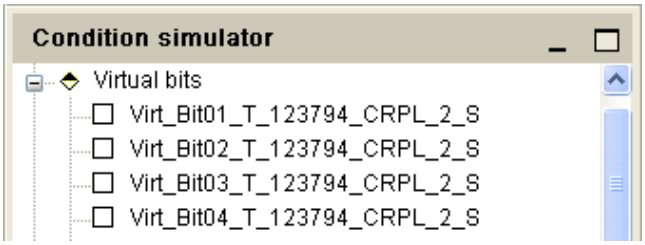
The curve plot displayed in the Output window reflects the settings which are valid in the window Conditions Simulator.



8.6 Sub-dialog Condition Simulator



This is a tool for simulating the functioning of conditions (setting DIO, virtual and Ether bits). The results can be viewed in the Output window.



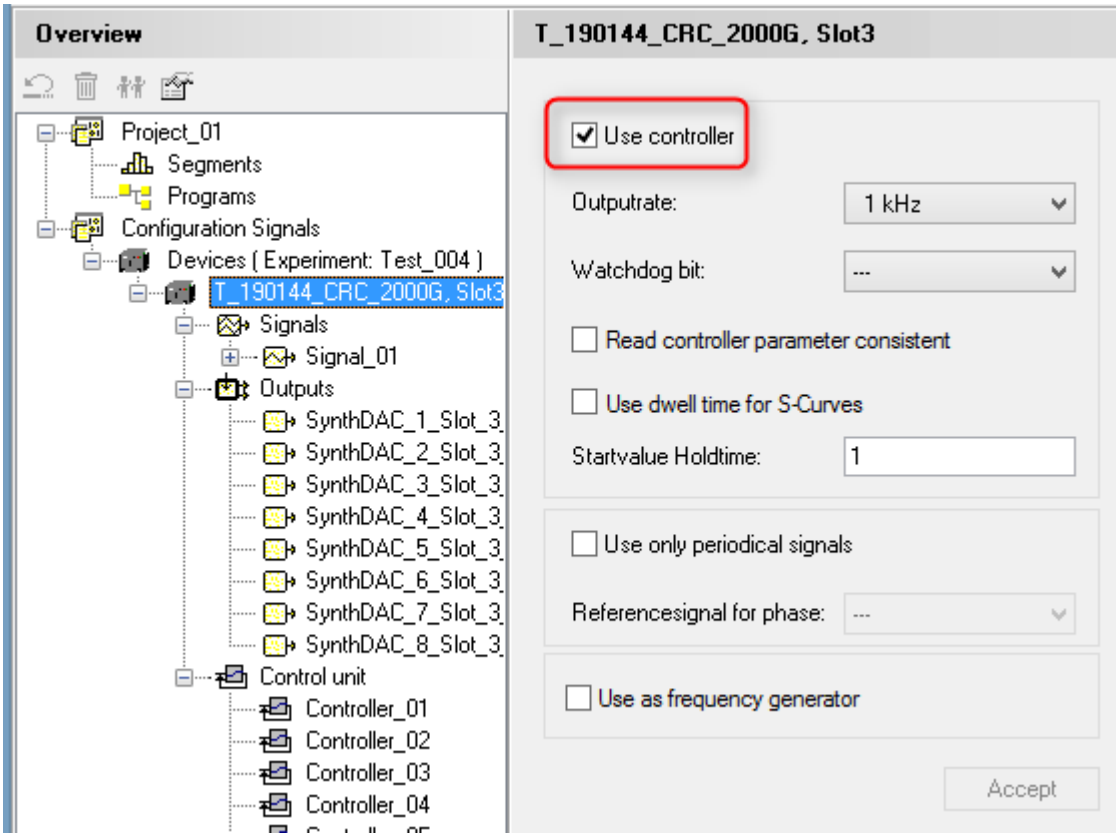
Virtual Bit 1, for instance, is set by expanding the entry Virtual Bits in the tree diagram and clicking on VirtBit_01. Whether or not the box in front of the bit's entry is checked indicates whether that bit is set.

9 Synthesizer with Controller Functionality

The prerequisite for controller functionality is a firmware version (imc DEVICES) as of 2.6 R1.

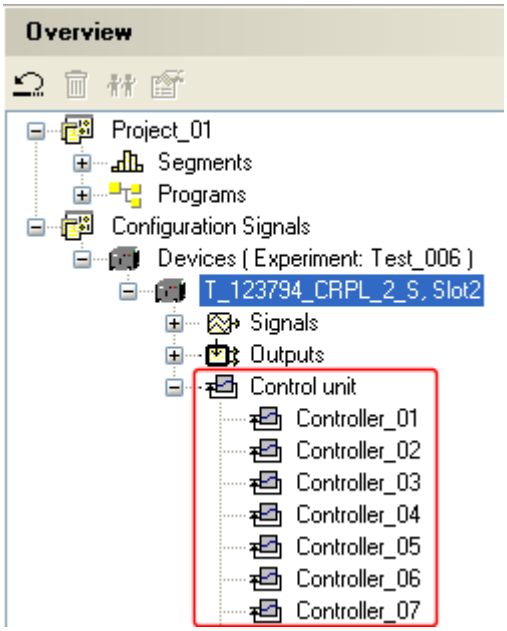
9.1 Activating the controller functionality

Select a synthesizer - slot in the tree diagram view, a configuration dialog appears on the right side:

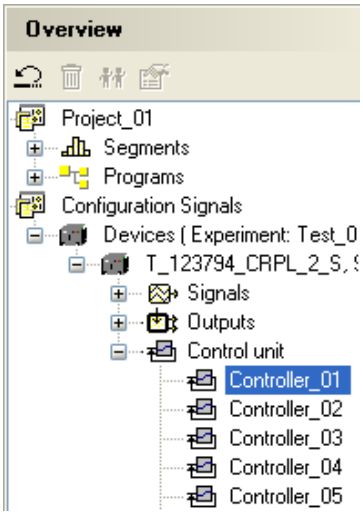


The controller is activated by check-marking the option box *Use controller*. It's not absolutely necessary to click on the button *Apply*; selecting another entry in the tree diagram will also cause the change to be applied.

Eight controllers appear in the tree diagram:

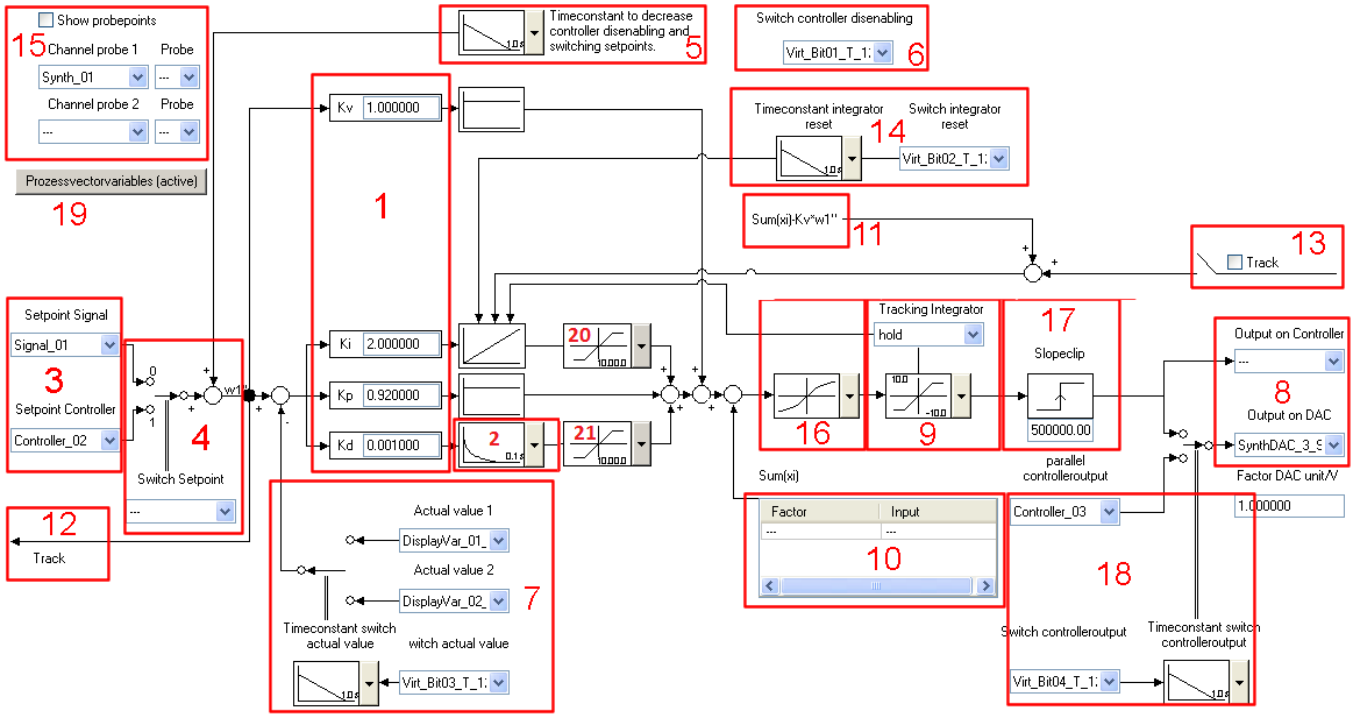


9.2 Configuring a closed-loop controller



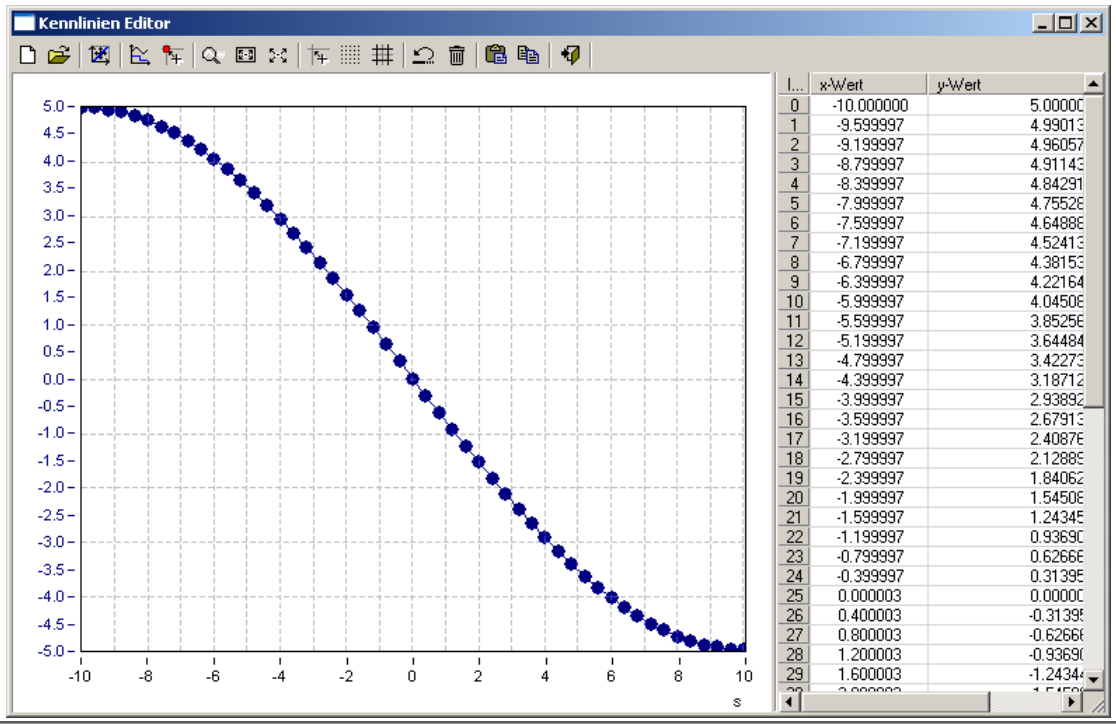
Once a controller has been selected, it can be configured by means of an editor.

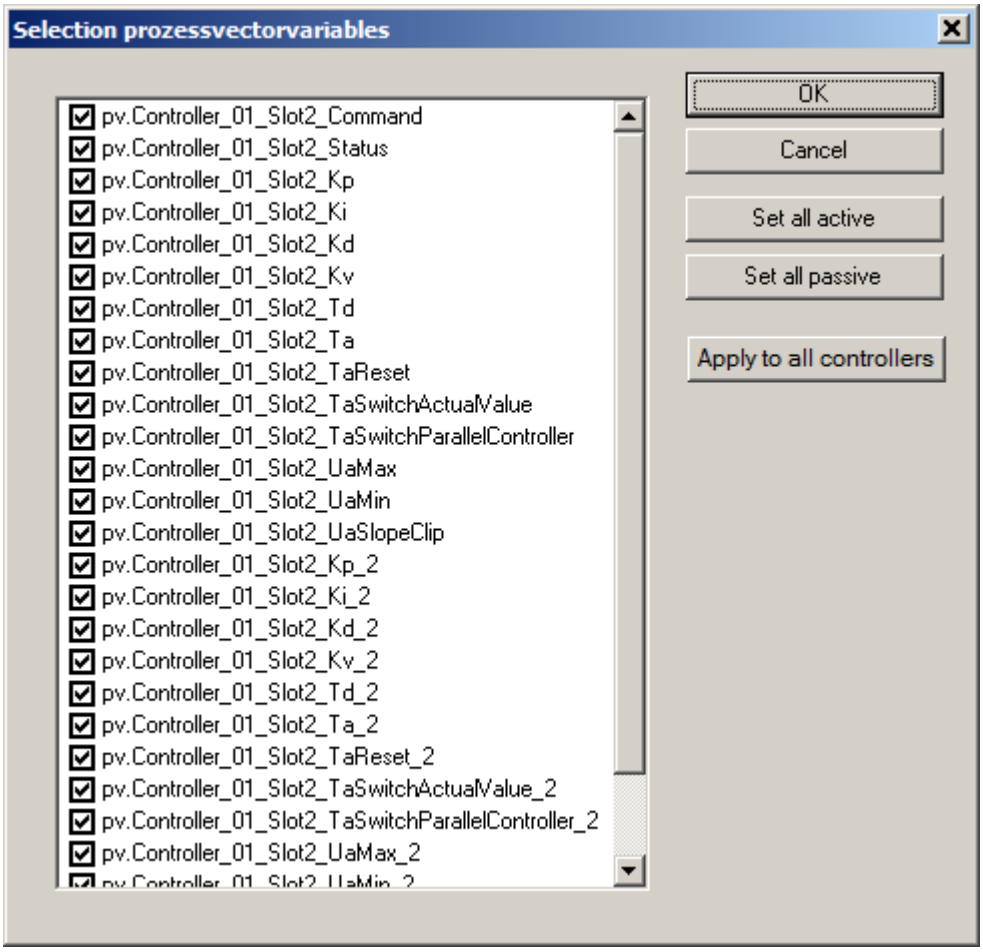
A schematic presentation of the controller overview is shown in a control circuit.



The following **parameters** can be set:

1)	the gain factor *Kv*, the Integral component *Ki*, the proportionality factor *Kp*, the differential component *Kd*
2)	the decay time of the differential component
3)	the target value
4)	The target value switch offers a choice between the target values for a signal and those of a different closed-loop controller. The latter is needed for creating a cascading multiple control system.
5)	the behavior of a target value switch: A time constant can be set at which the switching process between the two target values, or the controller disenable, is carried out. If the value zero is set, it is switched without delay.
6)	the controller disenable
7)	the actual value During the closed-loop control process, it is possible to switch between two actual values (7). For this purpose, the switching condition and the transition time can be set. (7 see switch actual value)
8)	the output channel, Here it is possible to distinguish between whether to output the actuating variable at a DAC output channel, or whether to apply it to a closed-loop controller downstream. The latter is needed for creating a cascading multiple control system.

9)	the control boundary, The control boundary determines the closed-loop control's maximum output values. It is also possible to set how the controller's integral component responds when the control boundary is reached. The following settings are possible: • "hold" – the Integral component is held at the maximum value Ua_{Max}. • "by difference Ua_{Max} – actuating variable" – the integral component is reduced by the actuating variable value. • "hold - if regulation error and actuating variable are equal" – the integral component is held at the maximum value Ua_{Max} if the sign of the regulation error and actuating variable are the same.
10)	additional summation of weighted measured values, There is a possibility to influence the controller by means of weighted state variables like other measured values. However, these must be previously included in the integration process and for that reason are also inputted into the controller under (11).
11)	see 10)
12)	Track
13)	Track
14)	Integrator Reset
15)	Probe points Certain "internal quantitiesc of the controller can be displayed in a channel.
16)	Characteristic curve The controller output can be a characteristic curve. By clicking on the button to the right of the symbol, a curve editor appears with which one can define characteristic curves in the same way as defining the signal segments. Additionally, it is possible to load any characteristic curves which are stored in FAMOS format on the PC. 
17)	Slope limiting
18)	Parallel controller xxxx During the control process, a DAC output can be switched from one controller to another. The transition time and the switching condition can be defined by the user.

19)	Process vector variable (active): Activation/deactivation of process vector variables for closed-loop controllers Each controller is associated with a large number of process vector variables. If an experiment involves multiple controllers, it is easy for the maximum count of process vector variables to be reached. Clicking on this button opens a dialog by means of which targeted variables can be activated, according to actual need:  With the button "*Apply to all controllers*" the current settings are applied to all of a slot's controllers. This saves the trouble of making the settings for each controller separately. If at least one process vector variable has been activated, the button is captioned in the controller's configuration dialog with *Process vector variable (active)*; but if all are passive, with *Process vector variable (passive)*.
20)	Integral component boundaries: Using this quantity, the integral component can be bound symmetrically around zero.
21)	Differential component boundaries: Using this quantity, the differential component can be bound symmetrically around zero.

9.3 Limitations caused by the controller function

When the controller function is active, the synthesizer operates with reduced signal quality: The output rate is only 10 Hz to 10 kHz instead of 400 kHz. Setting the output rate is accomplished by selecting the desired synthesizer-slots in the tree diagram. If overload of the synthesizer can be anticipated, an error message will be posted.

9.4 Governing a controller via imc Online FAMOS

The controller parameters can be changed by means of process vector variables, for instance by using imc Online FAMOS.

The names of the process vector variables begin with *pv.<ControllerName>_Slot_<SlotNumber>_*.

For each parameter, there are two process vector variables. For this reason it is possible to modify a parameter set and to switch to a different, secure parameter set in case the controlled system's behavior is unstable.

The table below lists the names and meanings of the individual process vector variables:

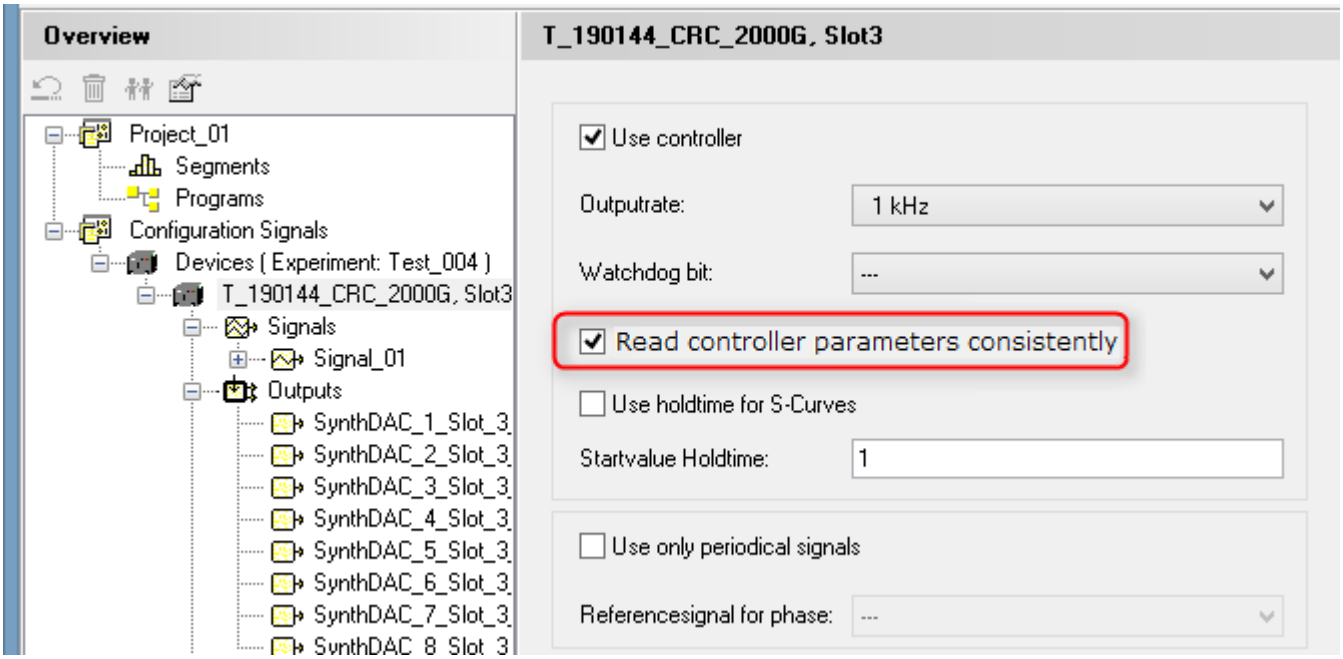
Name	**Meaning**
pv.<ControllerName>_Slot_<SlotNumber>__Kv pv.<ControllerName>_Slot_<SlotNumber>__Kv_2	Value for the gain factor (Kv)
pv.<ControllerName>_Slot_<SlotNumber>__Kp pv.<ControllerName>_Slot_<SlotNumber>__Kp_2	Value for the proportional component (Kp)
pv.<ControllerName>_Slot_<SlotNumber>__Ki pv.<ControllerName>_Slot_<SlotNumber>__Ki_2	Value for the integrator component (Ki)
pv.<ControllerName>_Slot_<SlotNumber>__Kd pv.<ControllerName>_Slot_<SlotNumber>__Kd_2	Value for the differential component (Kd)
pv.<ControllerName>_Slot_<SlotNumber>__Td pv.<ControllerName>_Slot_<SlotNumber>__Td_2	Time constant for the differential component
pv.<ControllerName>_Slot_<SlotNumber>__Ta pv.<ControllerName>_Slot_<SlotNumber>__Ta_2	Time constant controller disenable and input signal
pv.<ControllerName>_Slot_<SlotNumber>__TaReset pv.<ControllerName>_Slot_<SlotNumber>__TaReset_2	Time constant of integrator reset
pv.<ControllerName>_Slot_<SlotNumber> __TaSwitchActualValue pv.<ControllerName>_Slot_<SlotNumber> __TaSwitchActualValue_2	Time constant of actual value changeover
pv.<ControllerName>_Slot_<SlotNumber> __TaSwitchParallelController pv.<ControllerName>_Slot_<SlotNumber> __TaSwitchParallelController_2	Time constant of switchover to a parallel controller
pv.<ControllerName>_Slot_<SlotNumber>__UaMax pv.<ControllerName>_Slot_<SlotNumber>__UaMax_2	Maximum allowed setpoint signal
pv.<ControllerName>_Slot_<SlotNumber>__UaMin	Minimum allowed setpoint signal

Name	Meaning
pv.<ControllerName>_Slot_<SlotNumber>__UaMin_2	
pv.<ControllerName>_Slot_<SlotNumber>__UaSlopeClip pv.<ControllerName>_Slot_<SlotNumber>__UaSlopeClip_2	Maximum allowed rate setpoint signal
pv.<ControllerName>_Slot_<SlotNumber>__ParamSet	Currently used parameter set 1: Parameter set 1 is used 2: Parameter set 2 (with suffix "_2") is used.
pv.<ControllerName>_Slot_<SlotNumber>__IntLimit pv.<ControllerName>_Slot_<SlotNumber>__IntLimit_2	Max/Min allowed integrator value symmetrically around zero.
pv.<ControllerName>_Slot_<SlotNumber>__DiffLimit pv.<ControllerName>_Slot_<SlotNumber>__DiffLimit_2	Max/Min allowed differential value symmetrically around zero.

Setting an individual value/Consistent setting:

Setting of the controller parameter can be accomplished in two ways: **Individual value setting** or **Consistent setting**. The easiest way is individual value setting. The parameter value is written to the associated process vector variable and is immediately imported and applied by the synthesizer processor. However, the resulting combinations of parameters can cause the controlled system to behave erratically. In order to avoid this, there is an option to alter the controller parameters consistently. In this case, it is possible to write to the controller parameters without applying them immediately. Only once all the controller parameters have their new values, they can be applied by being written to an additional process vector variable.

In order to be able to use the consistent read/write mechanism, it is necessary to put a check in the box *"Read controller parameters consistently"* for the respective slot:



In order that the parameters can be read and written consistently, the process vector variables having the endings *Command* and *Status* are provided.

If *Command* and *Status = 0*, then the command can be set or parameters can be set/read.

For the purpose of consistent writing/reading, there are two commands available. With Command 130, the synthesizer processor writes the controller parameters belonging to the selected parameter set into the corresponding variables. With Command 131, the synthesizer processor reads the controller parameters belonging to the selected parameter set and writes them to the corresponding controller. It acknowledges reception of the command by writing the command to the variable *Status*.

Once parameters have been written/read, the processor resets the variables *Command* and *Status* to 0 and is ready to accept a new command.

In the following imc Online FAMOS program, the controller's proportional component is set to 5 if the Virtual Bit 2 is set to 1.

```
; Initialization prior to the first measurement
OnInitAll
    ControllerState = 0 ; Status of the controller upon transferring parameters
End

; continuous execution
OnAlways
End

; execution at the start of measurement
OnTriggerStart(BaseTrigger)
End

; execution during measurement
OnTriggerMeasure(BaseTrigger)
    ; GET_PARAMETERSET = 130 /* command for setting a controller's set of parameters */
    ; SET_PARAMETERSET = 131 /* command for getting a controller's set of parameters */
    ; Sets a new set of parameters using OnlineFAMOS with control commands
    if ( Virt_Bit02 = 1 )
        if (ControllerState = 0 ); the controller's parameters are unknown
            if ( pv.Controler_01_Slot1_Command = 0
                and pv.Controler_01_Slot1_Status = 0 )
                pv.Controler_01_Slot1_Command = 130
                ControllerState = 1
            end
        end
        if (ControllerState = 1 )    ; the controller's parameters are determined,
                                     ; once controller is again in Idle,
                                     ; write new parameter
            if ( pv.Controler_01_Slot1_Command = 0
                and pv.Controler_01_Slot1_Status = 0 )
                pv.Controler_01_Slot1_Command = 131
                pv.Controler_01_Slot1_Kp = 5.0
                ControllerState = 2
            end
        end
        if (ControllerState = 2 )  ; Parameters were written, once controller
                                   ; again in Idle, they were adopted
            if ( pv.Controler_01_Slot1_Command = 0
                and pv.Controler_01_Slot1_Status = 0 )
                ControllerState = 0
                Virt_Bit02 = 0
            end
        end
    end
End
```

Switching between the parameter sets:

By means of the interface command/status, it is also possible to switch between the two parameter sets.

If the value **135** is written to the command variable, the first parameter set is used and consistent reading/writing is applied to this parameter set.

With the value **136**, the second parameter set (with the suffix "_2") is used and consistent reading/writing is applied to this parameter set.

Switching between the parameter sets, the target values and the actual values:

In special situations, it is necessary to change the control system, along with the target values and actual values, within one sampling step.

For this purpose, the commands 137 and 138 are available.

With the command **137**, the first parameter set is used; simultaneously the system switches to the top target value and the top actual value.

With the command **138**, the second parameter set is used, simultaneously the system switches to the bottom target value and the bottom actual value.

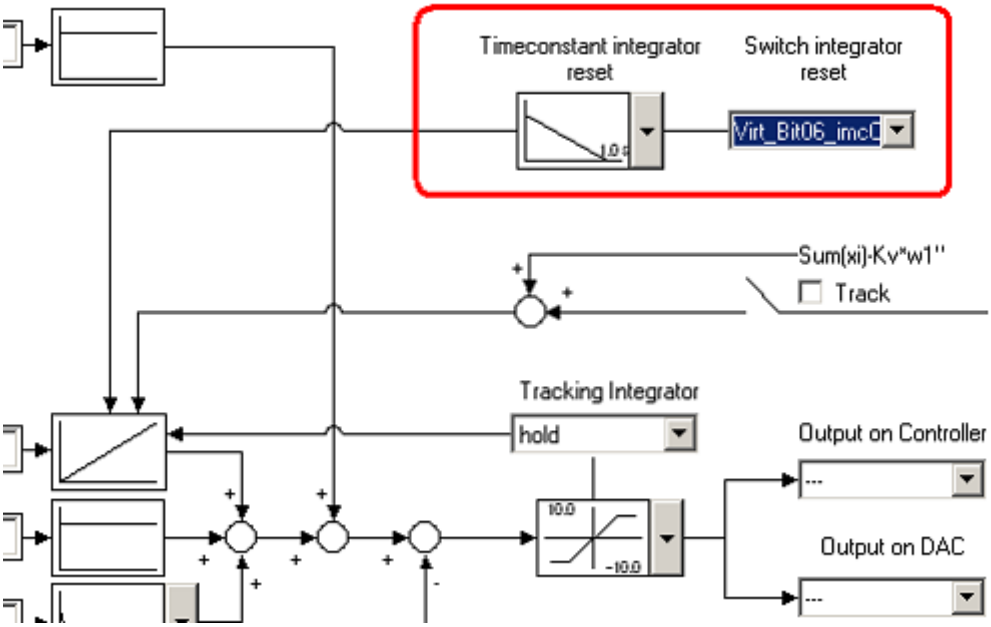
9.5 Integrator Reset

If you want to change the closed-loop control parameter during the process use the "Synthesizer-Controller". If the integration factor Ki becomes zero, the integration portion remains as an offset of the controller output. This may not be desirable, since it means that the controller is not functioning as expected. The function "Integrator Reset" allows the integration portion to be set to zero.

Using a previously configured virtual bit taking the value 1, the integration portion declines in a sloped pattern at a previously set time constant.

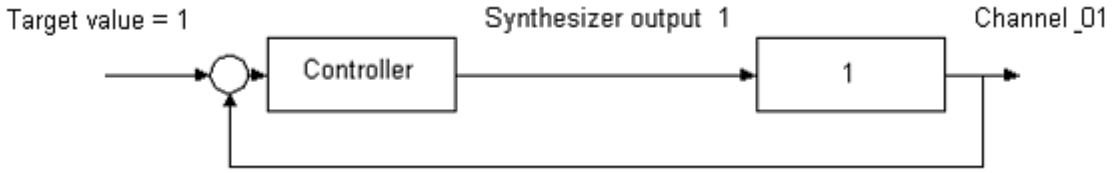
- If the bit is set to 0 during the decline, the integration portion is frozen at the value currently in effect, or
- if the value of Ki is not zero, the integrator resumes activity.

9.5.1 Feature in the controller dialog



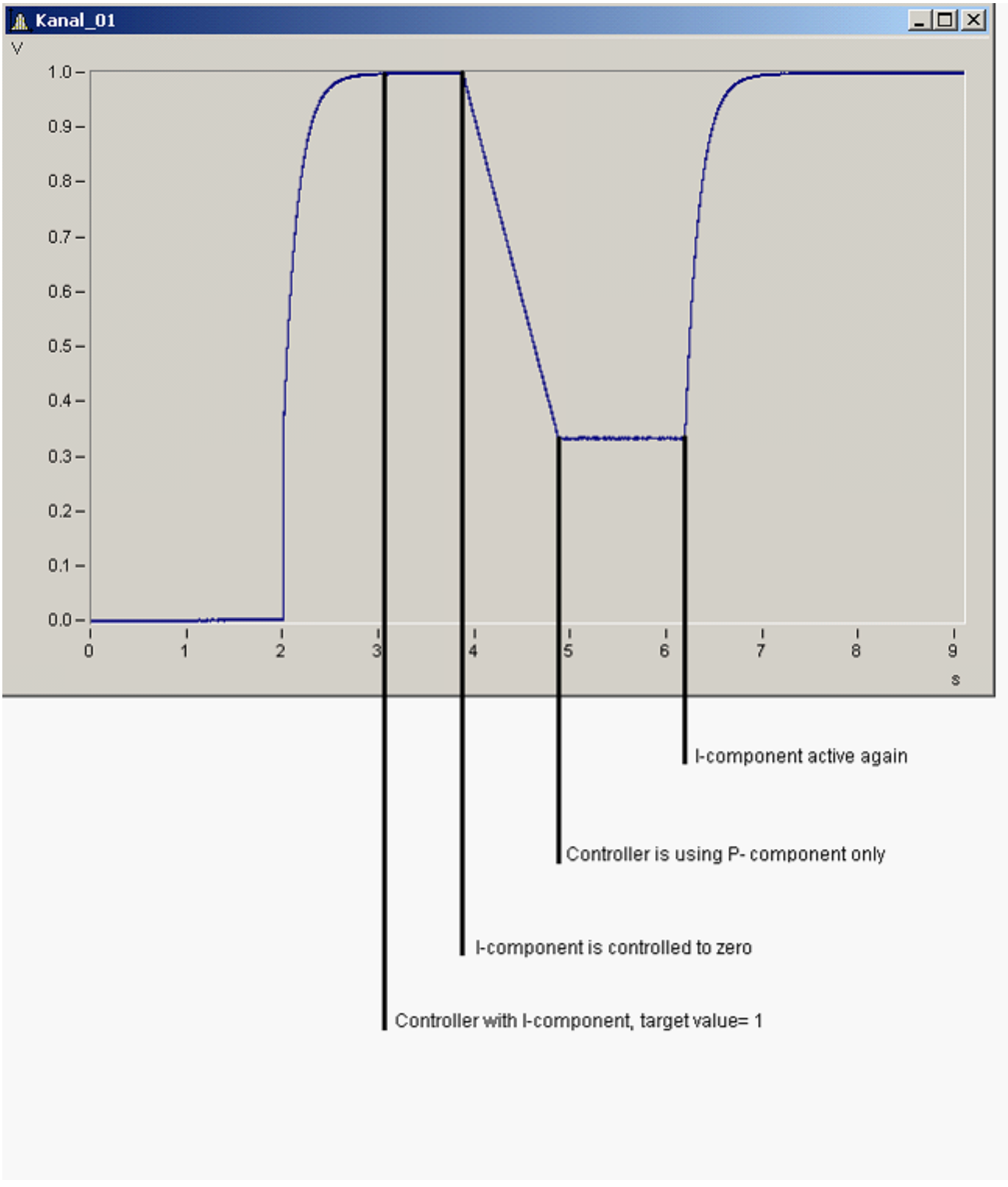
Example of an expected response:

The controller output is simply coupled back as the actual value, so the closed-loop control system is a simple proportional element with a gain of 1:

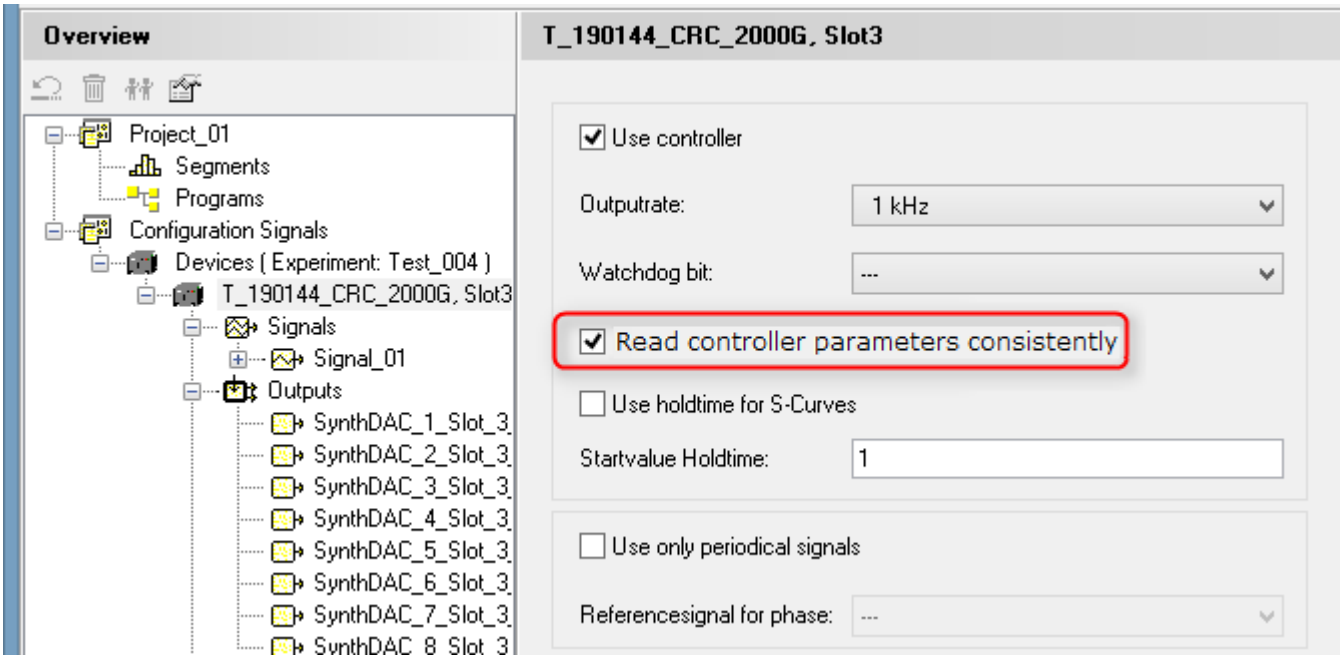


Controller parameter: Integration portion **Ki** = 10, Proportionality factor **Kp** = 0.5.

Result of the controller quantity when activating/ deactivating the integrator:



9.6 Read controller parameters consistently



consistent reading of the controller parameters from the process vector variables

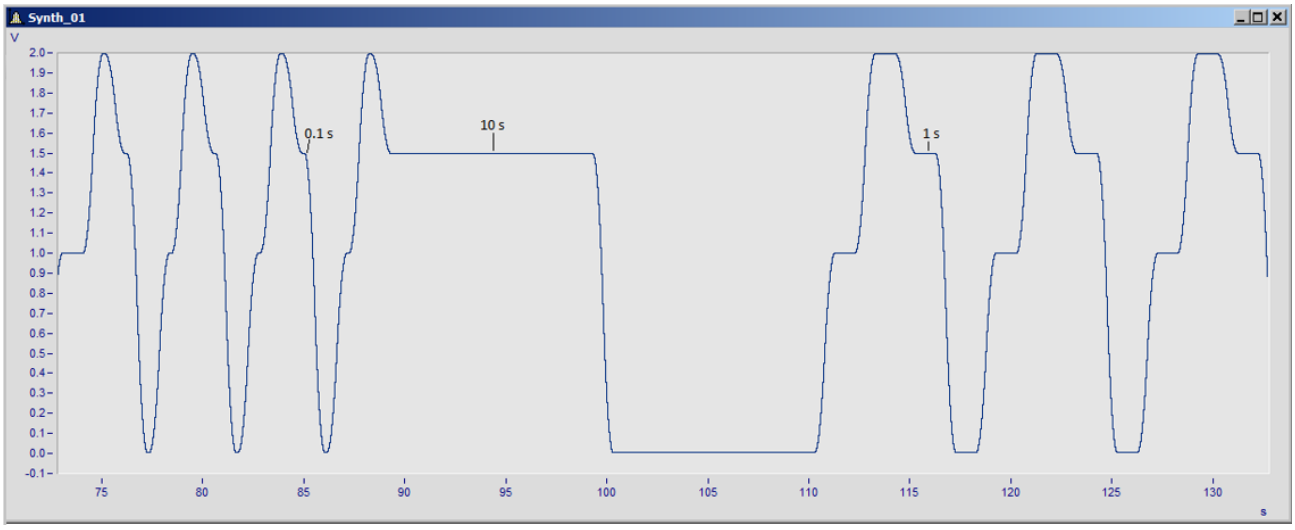
The controller parameters will not be readout one after another if you select the option "*Read controller parameters consistantly*". To commit the new controller parameters to the synthesizer you have to use a command interface.

9.7 Use dwell time for S-Curves

The outputted curve is calculated in such a way that the individual data points always begin and end with zero slope. By activating the controller mode and the option *"Use dwell for S-Curves"* and by specifying the framework of points in a certain way, it is possible to keep the curve at zero slope for a certain amount of time at this level. The time is governed globally for all of a device's S-curves by means of the process vector variable pv.<DeviceName>_HoldTime_SCurve during the output.

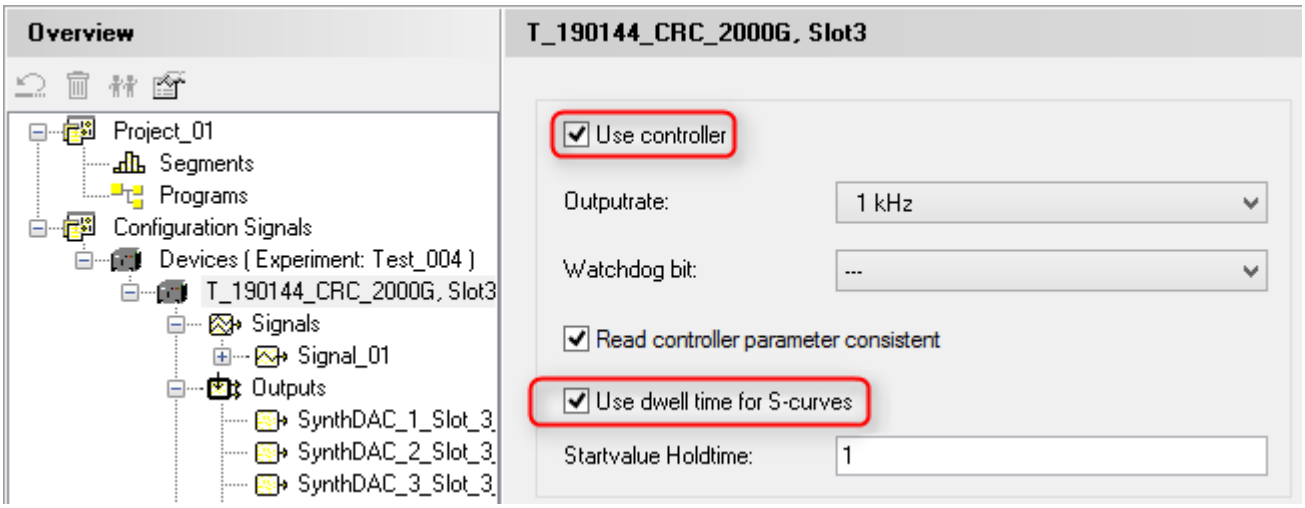
Note

For the dwell time, a value greater than zero must always be used.



Activating the option *"Use dwell for S-Curves"*

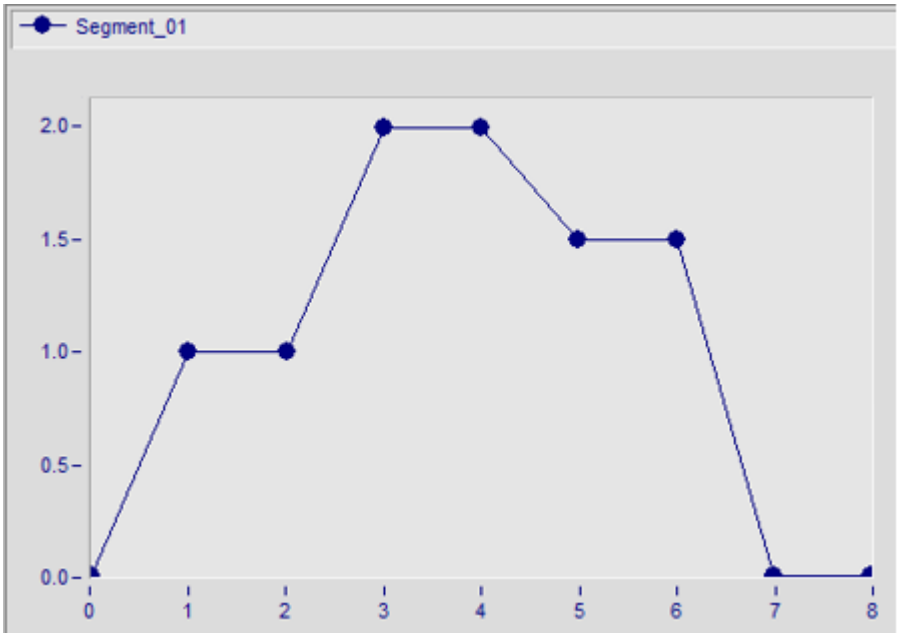
To do this, the respective Synthesizer–slot must be selected and in the configuration dialog at right the options ***"Use controller"*** and ***"Use dwell for S-Curves"*** must be activated.



Using holdoff times for S-curves

Specifying the framework of points:

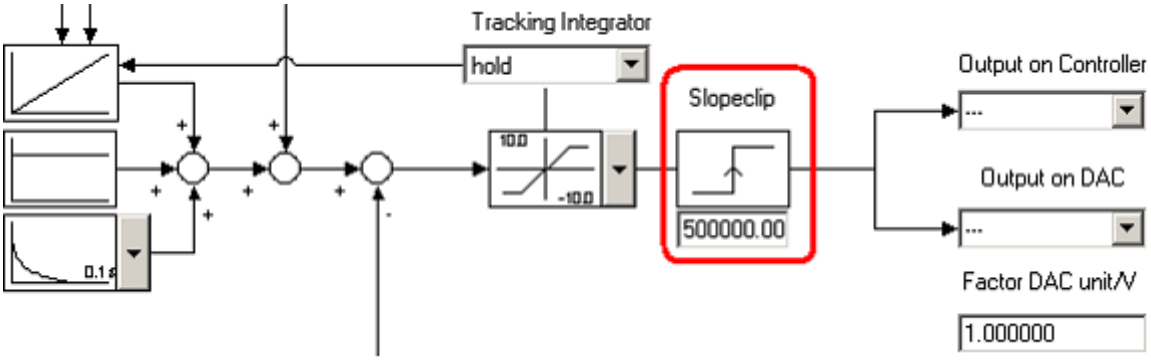
The framework of points must be specified in such a way that all points besides the first point are "duplicated", as shown in this example:



Y: 0 1 1 2 2 1.5 1.5 0 0
X: 0 1 2 3 4 5 6 7 8

9.8 Limitation of the steepness at the controller output

You can configure the maximum slew rate of the output of the controller.



In this example the integrator part will be handled just like an overrun of the upper and lower border to avoid a "windup".

10 Use of the Synthesizer as a frequency generator

For the purpose of outputting high-speed signals, the Synthesizer can be used in its Frequency Generator mode. The special preconditions for use of this mode are:

1. A signal may only contain one segment.
2. The associated segment is played back indefinitely.
3. The segment may contain no more than 1024 sample points.
4. All sample points must must be at equidistant time intervals.
5. For the output, only the DACs 1 and 2 or 5 and 6 may be used.

Changing the amplitude and offset during measurement:

The amplitude and offset of each individual signal can be changed by means of the associated process vector variables. The respective names are suffixed with _Factor and _Offset. This is preceded by the signal name, the label "Slot" and the Synthesizer board's slot number.

Changing the frequency:

By specifying a stretching factor, it is also possible to change a signal's frequency by means of the associated process vector variable. The name ends with "_Period". This is preceded by the signal name, the label "Slot" and the Synthesizer board's slot number.
If a segment having 1000 points over 1 s is defined, then with a value of 0.1, the frequency can be increased from 1 Hz to 10 Hz.

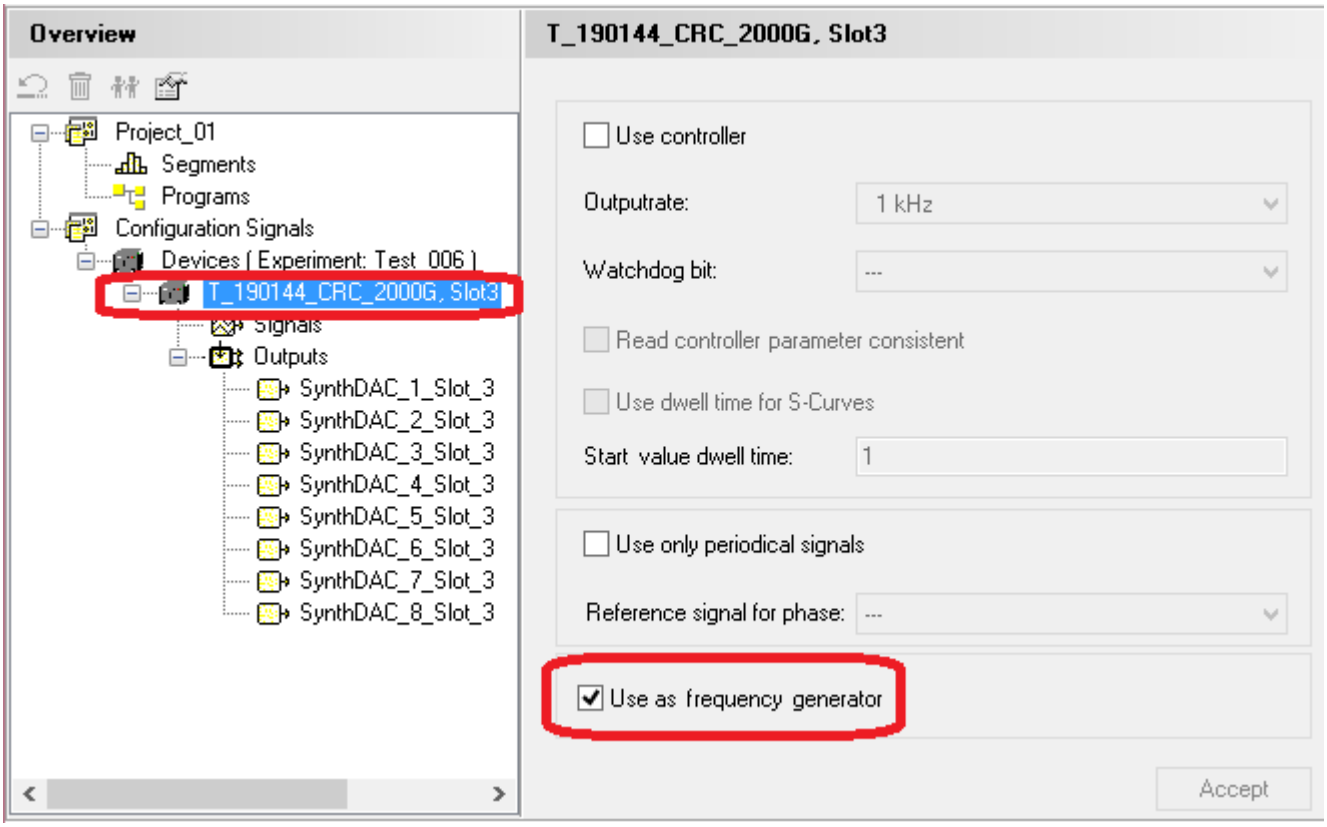
Manipulation of the frequency is subject to the following relationships:
Signals which are linked with DAC1 and DAC2 are interdependent in terms of frequency.
Changes affecting DAC1 are also exactly reflected in DAC2 and vice versa if they are respectively activated.
The same applies to DAC5 and 6. DAC1 and DAC5, however, can be set independently of each other.
If the distance between sample points falls below a time of 2 µs, then sample points are omitted until this distance is restored.

Example

1000 sample points distributed over 1 s => 1 ms per sample point. As the factor, 0.0001 is to be selected, in order to achieve an output frequency of 10 kHz. The resulting point distance would be 1 ms * 0.0001 = 100 ns. In order to avoid falling below 2 µs, only every 20th point (20 * 100 ns = 2 µs) is accepted, i.e. only 50 instead of 1000 sample points are used to generate the curve.

10.1 Activating Frequency Generator Mode

In order to be able to use the Frequency Generator mode, it is necessary to select the associated Synthesizer board from the tree diagram on the left side in the Assistant. On the right side, a settings dialog appears. Here, the option "Use as frequency generator" must be activated:



Note

In the simplest case, it makes sense to define the signal to be outputted with 1000 points distributed over 1 second. Then by means of the appropriate factor, the desired frequency can be set.

E.g. the factor 0.1 generates a signal with a 10 Hz fundamental frequency; a factor of 0.01 produces 100 Hz.

10.2 Example

A sine signal having 1000 points distributed over 1 s is to be outputted.

1. Generating a segment:
   Select the element "Segments" from the tree diagram at left.
   Click on the button "new segment"  .
   In the tree diagram at left, select the newly created segment.
   In the Segment Editor, click on the item "Standard signals"  .
   As the distance between data points, select the value 0.001 instead of 0.01.
   Click on the button "Insert", then click on the curve window and close the dialog.
2. Creating the signal:
   Move the segment to the desired Synthesizer output by means of Drag and Drop.
   In the dialog which subsequently opens, select "undefined" for the count of repetitions.
3. Activate the Frequency Generator mode.
4. Exit the Synthesizer Assistant
5. In imc Online FAMOS, link the associated process vector variable with a Display variable by means of the following command line: `"pv.Signal_01_Slot1_Period=DisplayVar_01"`. If it has been ensured that the Display variable does not become zero, then alternatively the reciprocal can also be selected in order to be able to set the frequency directly: `"pv.Signal_01_Slot1_Period=1.0/DisplayVar_01"`.

11 From the simulation to actual practice

Now that all the individual steps and options have been presented, here is a brief outline of how to proceed in practice:

1. **Create segments**: Define the course of one or more output signals.
2. **Create a program**: Your program must determine the sequence of the segments. Decide how many times to repeat a segment and according to what conditions the program proceeds.
3. **Save** segments and a program in a project. In this way, they will be available to open for new signals later on.
4. Drag the program's entry in the tree to the entries for the signals desired; this creates an **output signal**.
5. Provide names for the **conditions** and/or repetitions.
6. Assign an **output channel**.
7. Set the output window and the conditions simulator window to be displayed.
8. **Check** the conditions. Does the signal appear as you expected in response to the conditions? If so, save your project and exit the signal generator.
9. **Connect** your measurement device. Make certain that the Synthesizer output has been connected correctly.
10. When the measurement is **prepared**, the Synthesizer begins to run its program. If output is to begin upon the start of the measurement, a condition built into the Synthesizer program must arrange it, for example, by setting a virtual bit upon the measurement's start.

It is recommended for your first attempts to measure the Synthesizer output along with the channel, and to have the output signal displayed in a curve window.

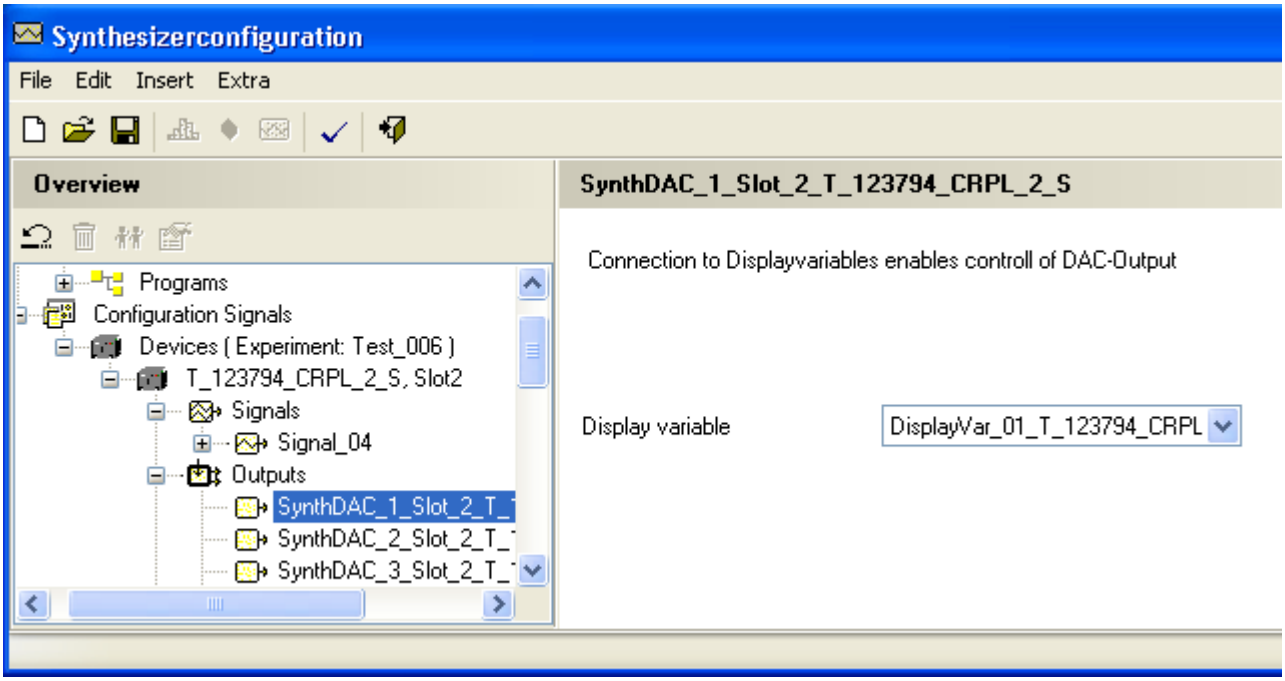
12 Assigning a display variable to a synthesizer output

Not only signals can be assigned to synthesizer outputs; outputting display variables is also possible. By these means, the output can be addressed in the same way as a normal device-DAC via Online FAMOS or the DIODAC-dialog.

For this purpose, there are various ways to intervene, depending on the mode (purely synthesizer-mode, controller mode).

Synthesizer mode

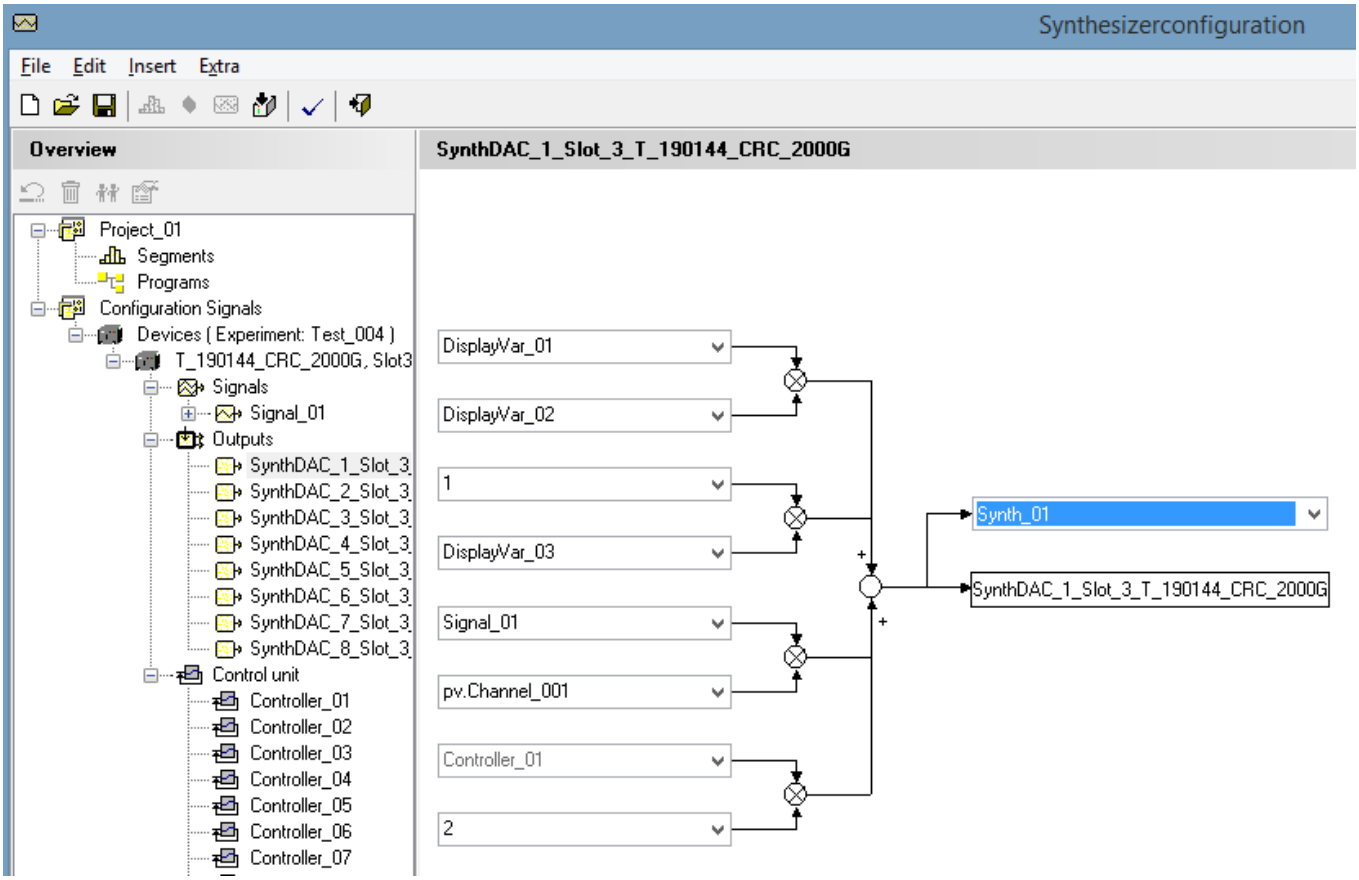
When you select a synthesizer output in the overview, a configurator in which the display variable can be selected appears on the right side of the dialog:



Initially, no Display variable is selected in the selection list, which is indicated by the readout "---". Pop the list down and select the variable desired.

Controller mode:

When a Synthesizer output is selected in the Overview, a configurator appears on the right side:



Pairs of quantities can be multiplied with each other and added to the output.

Possible quantities include signal- and controller outputs, Display- and process vector variables, as well as freely declarable constants.

The symbol "---" denotes where the mixer input is not occupied.

Both factors must always be occupied, but summands can remain unoccupied. In the dialog shown as an example, all summands are occupied.

The mixer's output can be applied to an output channel on the right side, for debugging purposes.

Index

G

H

I

L

M

N

O

P

Q

R

S

T

U

W