SIEMENS

SIPLUS CMS

SIPLUS CMS4000 X-Tools

User Manual - 06 - Analyzing System

English

Release 2011-09

Safety Guidelines

This document contains notices which you should observe to ensure your own personal safety as well as to avoid property damage. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol

Danger

Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

Warning

Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

Caution

Used with the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

Caution

Used without safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

Notice

Used without the safety alert symbol indicates a potential situation which, if not avoided, may result in an undesirable result or state.

When several danger levels apply, the notices of the highest level (lower number) are always displayed. If a notice refers to personal damages with the safety alert symbol, then another notice may be added warning of property damage.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this documentation. Only qualified personnel should be allowed to install and work on the equipment. Qualified persons are defined as persons who are authorized to commission, to earth, and to tag circuits, equipment and systems in accordance with established safety practices and standards.

Intended Use

Please note the following:

Warning

This device and its components may only be used for the applications described in the catalog or technical description, and only in connection with devices or components from other manufacturers approved or recommended by Siemens. This product can only function correctly and safely if it is transported, stored, set up and installed correctly, and operated and maintained as recommended.

Trademarks

All designations marked with ® are registered trademarks of Siemens AG. Other designations in this documentation might be trademarks which, if used by third parties for their purposes, might infringe upon the rights of the proprietors.

Disclaimer of Liability

We have checked the contents of this document for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in the manual are reviewed regularly, and any necessary corrections will be included in subsequent editions. Suggestions for improvement are welcomed.

Siemens AG
Industry Sector
P.O. Box 4848
90327 Nuremberg
Germany

Siemens AG 2011
Technical data subject to change

English
Release 2011-09

Table of Contents

Table of Figures

1 Preface

1.1 Purpose of this Document

This document provides detailed information about the functionalities and usage of the software

- SIPLUS CMS4000 X-Tools

of the SIPLUS CMS product line.

In addition to the detailed information about each dialog and functionality of the **Analyzing System** which is found within this document, also the following documentation is available:

- SIPLUS CMS4000 X-Tools - User Manual - 01 - Introduction
  - provides an introduction into the basic functionalities of SIPLUS CMS4000 ***X-Tools***
- SIPLUS CMS4000 X-Tools - User Manual - 02 - Master Data System
  - provides detailed information about the functionality which is provided by the **Master Data System**
- SIPLUS CMS4000 X-Tools - User Manual - 03 - Main Management System
  - provides detailed information about the functionality which is provided by the **Main Management System**
- SIPLUS CMS4000 X-Tools - User Manual - 04 - Device Management System
  - provides detailed information about the functionality which is provided by the **Device Management System**
- SIPLUS CMS4000 X-Tools - User Manual - 05 - Monitoring System
  - provides detailed information about the functionality which is provided by the **Monitoring System**
- SIPLUS CMS4000 X-Tools - User Manual - 07 - Storage System
  - provides detailed information about the functionality which is provided by the **Storage System**
- SIPLUS CMS4000 X-Tools - Release Notes
  - provides additional information about the released version of SIPLUS CMS4000 ***X-Tools***
- SIPLUS CMS4000 X-Tools - Change Log
  - provides an overview about the changes which have been introduced with the current version of SIPLUS CMS4000 ***X-Tools***

1.2 Validity of this Document

This document is valid for the following software:

- SIPLUS CMS4000 X-Tools Demo V 03.04
- SIPLUS CMS4000 X-Tools Standard V 03.04
- SIPLUS CMS4000 X-Tools Professional V 03.04

During the following pages, these software packages will be referred to by the term ***X-Tools***.

1.3 Audience

This document is intended for personnel involved in the commissioning and using of the software:

- ***X-Tools***

1.4 Notations

The following notations are used within this document:

- ***bold, italic*** text is being used for the main executables of ***X-Tools***
  - examples: ***X-Tools Client***, ***X-Tools Server***
- **bold** text is being used for the software modules of ***X-Tools***
  - examples: **Main Management System**, **Device Profile Editor**, **IPE Socket T001**
- green text is being used for controls like tables and trees
  - examples: Main Profile Settings table, Device Profile Data table
- orange text is being used for simple controls like a menu button, a single row/column/cell of a table or a branch of a tree
  - examples: Open… menu button, IP Address column, Target Device Name cell, Interfaces Branch
- dark yellow text is being used for the entries of context menus
  - examples: Advanced Append…, Edit
- Camel Notation is being used for major terms of ***X-Tools***
  - examples: Main Profile, Interface Profile, User Accounts File, Analyzing Function
- < and > brackets are being used for keyboard keys
  - examples: <Ctrl>, <Alt>, <Shift>, <Del>
- [and] brackets are being used for mouse operations
  - examples: [left mouse button down], [mouse move]

2 Analyzing System

2.1 Analyzing System Explorer

2.1.1 Overview

The **Analyzing System Explorer** (in the following, the **ANS Explorer**) is used in order to visualize and maintain all of the files and modules which are relevant for the **Analyzing System**. It is displayed as a tree which contains all of the relevant and available **Analyzing System** items. Via Drag&Drop the user is able to move items within the **ANS Explorer** and from the **ANS Explorer** to other dialogs of the **Analyzing System**. The following lines provide a short overview about the information that is available from the **ANS Explorer**.

The following screenshot shows an example of an **ANS Explorer**:

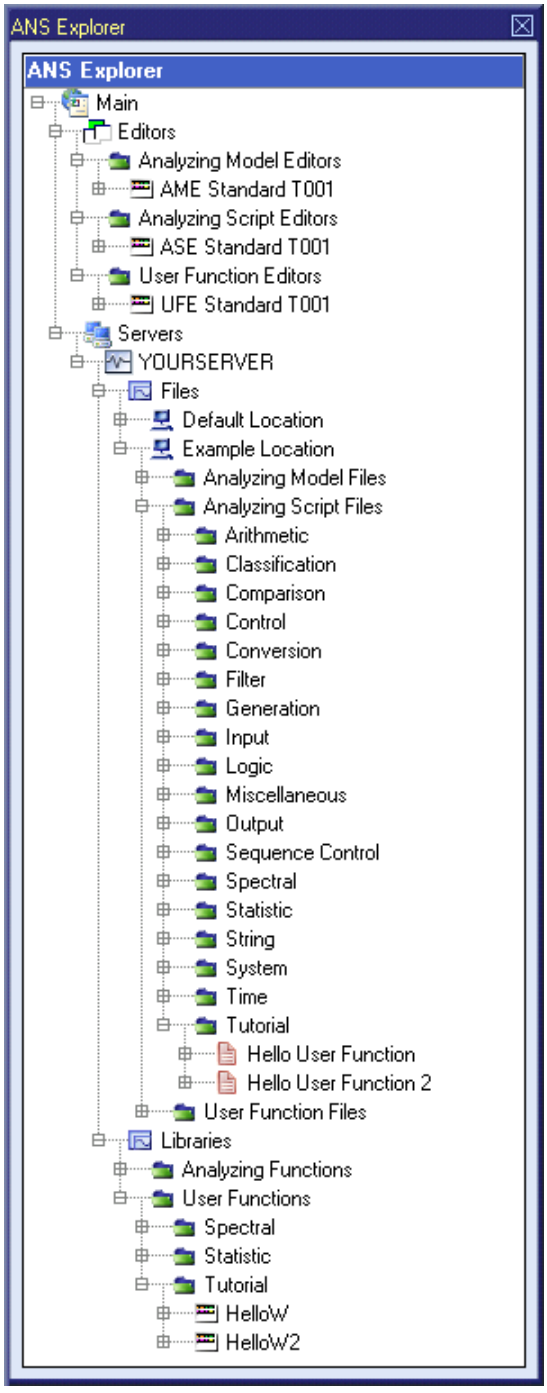


*Figure 1: Example of an **ANS Explorer***

Each branch of the **ANS Explorer** has a defined task and provides certain functionalities. The following major branches are provided by the **ANS Explorer**:

- Main Branch
- Editors Branch
- Analyzing Model Editors Branch
- Analyzing Model Editor Branch
- Analyzing Script Editors Branch
- Analyzing Script Editor Branch
- User Function Editors Branch
- User Function Editor Branch
- Servers Branch
- Server Branch
- Files Branch
- File Location Branch
- Analyzing Model Files Branch
- Analyzing Script Files Branch
- User Function Files Branch
- Libraries Branch
- Analyzing Functions Branch
- Analyzing Function Branch
- User Functions Branch
- User Function Branch

2.1.2 Main Branch

The one and only Main Branch provides all of the other items of the **ANS Explorer**. It can be expanded and collapsed in order to show or hide its sub-items.

2.1.3 Editors Branch

The one and only Editors Branch provides all of the editors which are available within the **Analyzing System**.

2.1.4 Analyzing Model Editors Branch

The one and only Analyzing Model Editors Branch provides all of the **Analyzing Model Editors** which are available within the **Analyzing System**. The tree of shown **Analyzing Model Editors** is updated automatically whenever an editor file at the local disk is added/removed/modified.

2.1.5 Analyzing Model Editor Branch

Each Analyzing Model Editor Branch represents one available **Analyzing Model Editor**. Dragging of an Analyzing Model Editor Branch into the **Analyzing System** workspace opens an empty editor of the dragged type. After the editor has been opened, any of the present files of correct type can be dragged into it for visualization and/or editing.

The following specific context menu items are provided:

Context Menu Item	Description
Open	opens an empty **Analyzing Model Editor** of the chosen type within the **Analyzing System** workspace
New Analyzing Model > …	opens an **Analyzing Model Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Model and initializes it with default values

2.1.6 Analyzing Script Editors Branch

The one and only Analyzing Script Editors Branch provides all of the **Analyzing Script Editors** which are available within the **Analyzing System**. The tree of shown **Analyzing Script Editors** is updated automatically whenever an editor file at the local disk is added/removed/modified.

2.1.7 Analyzing Script Editor Branch

Each Analyzing Script Editor Branch represents one available **Analyzing Script Editor**. Dragging of an Analyzing Script Editor Branch into the **Analyzing System** workspace opens an empty editor of the dragged type. After the editor has been opened, any of the present files of correct type can be dragged into it for visualization and/or editing.

The following specific context menu items are provided:

Context Menu Item	Description
Open	opens an empty **Analyzing Script Editor** of the chosen type within the **Analyzing System** workspace
New Analyzing Script > …	opens an **Analyzing Script Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Script and initializes it with default values

2.1.8 User Function Editors Branch

The one and only User Function Editors Branch provides all of the **User Function Editors** which are available within the **Analyzing System**. The tree of shown **User Function Editors** is updated automatically whenever an editor file at the local disk is added/removed/modified.

2.1.9 User Function Editor Branch

Each User Function Editor Branch represents one available **User Function Editor**. Dragging of a User Function Editor Branch into the **Analyzing System** workspace opens an empty editor of the dragged type. After the editor has been opened, any of the present files of correct type can be dragged into it for visualization and/or editing.

The following specific context menu items are provided:

Context Menu Item	Description
Open	opens an empty **User Function Editor** of the chosen type within the **Analyzing System** workspace
New User Function > …	opens a **User Function Editor** of the chosen type within the **Analyzing System** workspace, creates a new User Function and initializes it with default values

2.1.10 Servers Branch

The one and only Servers Branch provides all of the ***X-Tools Servers*** which are connected at the moment. The tree of shown ***X-Tools Servers*** is updated automatically whenever an ***X-Tools Server*** is attached or detached.

2.1.11 Server Branch

Each Server Branch represents one of the currently connected ***X-Tools Servers***.

2.1.12 Files Branch

The Files Branch of each connected ***X-Tools Server*** provides all of the files which are available within the **Analyzing System**.

The following specific context menu item is provided:

Context Menu Item	Description
Add Location…	opens the **Add Location** dialog and adds a new Configuration File location afterwards

2.1.13 File Location Branch

Each File Location Branch provides all of the **Analyzing System** specific files which are available from the directory to which the Configuration File location points.

The following specific context menu items are provided:

Context Menu Item	Description
Delete Location	deletes the chosen file location from the hard disk
Remove Location	removes the chosen file location from ***X-Tools*** but keeps it at the hard disk

2.1.14 Analyzing Model Files Branch

Each Analyzing Model Files Branch provides all of the Analyzing Models which are available from the directory to which its Configuration File location points. The tree of shown Analyzing Models is updated automatically whenever an Analyzing Model at the disk is added/deleted/modified.

Drag&Drop can be used in order to copy/move Analyzing Model directories and Analyzing Model files. The default Drag&Drop operation within an ***X-Tools Server*** is “move”, but when the <Ctrl> key is pressed a “copy” operation is performed. The default Drag&Drop operation from one ***X-Tools Server*** to another is “copy”, but when the <Shift> key is pressed a “move” operation is performed.

Analyzing Model directories and Analyzing Model files can be copied/moved within Analyzing Model Files Branches (either within one ***X-Tools Server*** or over different ***X-Tools Servers***) but they can not be copied/moved to another files branch. As the unique name of each Analyzing Model also includes its storage directory, multiple Analyzing Models with matching file names can be stored in different Analyzing Model directories.

Dragging of an Analyzing Model into the **Analyzing System** workspace opens the **Analyzing Model Editor** for the dragged Analyzing Model.

The following specific context menu items are provided:

Context Menu Item	Description
New Analyzing Model > …	opens an **Analyzing Model Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Model and initializes it with default values
Edit	opens a new **Analyzing Model Editor** for the chosen file within the **Analyzing System** workspace
Cut	cuts the currently selected items
Copy	copies the currently selected items
Paste	pastes currently copied/cut items
Delete	deletes the selected items from the disk
Rename	allows to rename the selected item directly within the **ANS Explorer**
New Directory…	opens the **Add Directory** dialog and creates a new directory afterwards
Start	starts the currently selected items
Pause	pauses the currently selected items
Continue	continues the currently selected items
Stop	stops the currently selected items

The following icons are provided and indicate the current status of each Analyzing Model:

Icon	Description
(stopped)	The Analyzing Model is stopped and does not perform any operations.
(started)	The Analyzing Model is started and performs the configured calculations. The resulting data is put into the ring buffers and can be accessed via the **MDS Explorer**.
(paused)	When the Analyzing Model is paused it freezes the current contents of the ring buffers of its data and it stops to perform its calculations. The data which is produced by the Analyzing Model is not removed from the **MDS Explorer** and keeps available.
(suspended)	The Analyzing Model is suspended because it has detected an error condition (e.g. because the input data became unavailable or because it was unable to perform its calculations in time, …). The Analyzing Model will resume after the error condition has been solved and all of the data which is produced by the Analyzing Model is being removed from the **MDS Explorer** until the Analyzing Model can resume. In case the Analyzing Model is suspended and its according Analyzing Model File at the disk is being updated, the updated Analyzing Model File is being loaded and the Analyzing Model uses the updated configuration for its further processing.

2.1.15 Analyzing Script Files Branch

Each Analyzing Script Files Branch provides all of the Analyzing Scripts which are available from the directory to which its Configuration File location points. The tree of shown Analyzing Scripts is updated automatically whenever an Analyzing Script at the disk is added/deleted/modified.

Drag&Drop can be used in order to copy/move Analyzing Script directories and Analyzing Script files. The default Drag&Drop operation within an ***X-Tools Server*** is “move”, but when the <Ctrl> key is pressed a “copy” operation is performed. The default Drag&Drop operation from one ***X-Tools Server*** to another is “copy”, but when the <Shift> key is pressed a “move” operation is performed.

Analyzing Script directories and Analyzing Script files can be copied/moved within Analyzing Script Files Branches (either within one ***X-Tools Server*** or over different ***X-Tools Servers***) but they can not be copied/moved to another files branch. As the unique name of each Analyzing Script also includes its storage directory, multiple Analyzing Scripts with matching file names can be stored in different Analyzing Script directories.

Dragging of an Analyzing Script into the **Analyzing System** workspace opens the **Analyzing Script Editor** for the dragged Analyzing Script.

The following specific context menu items are provided:

Context Menu Item	Description
New Analyzing Script > …	opens an **Analyzing Script Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Script and initializes it with default values
Edit	opens a new **Analyzing Script Editor** for the chosen file within the **Analyzing System** workspace
Cut	cuts the currently selected items
Copy	copies the currently selected items
Paste	pastes currently copied/cut items
Delete	deletes the selected items from the disk
Rename	allows to rename the selected item directly within the **ANS Explorer**
New Directory…	opens the **Add Directory** dialog and creates a new directory afterwards
Start	starts the currently selected items
Pause	pauses the currently selected items
Continue	continues the currently selected items
Stop	stops the currently selected items

The following icons are provided and indicate the current status of each Analyzing Script:

Icon	Description
(stopped)	The Analyzing Script is stopped and does not perform any operations.
(started)	The Analyzing Script is started and performs the configured calculations. The resulting data is put into the ring buffers and can be accessed via the **MDS Explorer**.
(paused)	When the Analyzing Script is paused it freezes the current contents of the ring buffers of its data and it stops to perform its calculations. The data which is produced by the Analyzing Script is not removed from the **MDS Explorer** and keeps available.
(suspended)	The Analyzing Script is suspended because it has detected an error condition (e.g. because the input data became unavailable or because it was unable to perform its calculations in time, …). The Analyzing Script will resume after the error condition has been solved and all of the data which is produced by the Analyzing Script is being removed from the **MDS Explorer** until the Analyzing Script can resume. In case the Analyzing Script is suspended and its according Analyzing Script File at the disk is being updated, the updated Analyzing Script File is being loaded and the Analyzing Script uses the updated configuration for its further processing.

2.1.16 User Function Files Branch

Each User Function Files Branch provides all of the User Functions which are available from the directory to which its Configuration File location points. The tree of shown User Functions is updated automatically whenever a User Function at the disk is added/deleted/modified.

Drag&Drop can be used in order to copy/move User Function directories and User Function files. The default Drag&Drop operation within an ***X-Tools Server*** is “move”, but when the <Ctrl> key is pressed a “copy” operation is performed. The default Drag&Drop operation from one ***X-Tools Server*** to another is “copy”, but when the <Shift> key is pressed a “move” operation is performed.

User Function directories and User Function files can be copied/moved within User Function Files Branches (either within one ***X-Tools Server*** or over different ***X-Tools Servers***) but they can not be copied/moved to another files branch. As the unique name of each User Function also includes its storage directory, multiple User Functions with matching file names can be stored in different User Function directories.

Note

After copying of a User Function within one ***X-Tools Server***, there will be two User Function Files with identical information inside. As the short name and the full name of each User Function must be unique, a conflict will be detected after the copy and both User Functions won’t be usable by Analyzing Models and Analyzing Scripts.

In order to resolve the conflict, open one of the User Functions within a **User Function Editor** and change its full name and short name there.

Dragging of a User Function into the **Analyzing System** workspace opens the **User Function Editor** for the dragged User Function.

The following specific context menu items are provided:

Context Menu Item	Description
New User Function > …	opens a **User Function Editor** of the chosen type within the **Analyzing System** workspace, creates a new User Function and initializes it with default values
Edit	opens a new **User Function Editor** for the chosen file within the **Analyzing System** workspace
Cut	cuts the currently selected items
Copy	copies the currently selected items
Paste	pastes currently copied/cut items
Delete	deletes the selected items from the disk
Rename	allows to rename the selected item directly within the **ANS Explorer**
New Directory…	opens the **Add Directory** dialog and creates a new directory afterwards

2.1.17 Libraries Branch

The Libraries Branch of each connected ***X-Tools Server*** provides all of the libraries which are available for the **Analyzing System**.

2.1.18 Analyzing Functions Branch

The Analyzing Functions Branch of each connected ***X-Tools Server*** provides all of the Analyzing Functions which are available for the **Analyzing System**. The tree of shown Analyzing Functions is updated automatically whenever an Analyzing Function at the disk is added/removed/modified.

The categories below the Analyzing Functions Branch are determined automatically according to the categories which are reported by the present Analyzing Functions and each present Analyzing Function is listed below its reported category.

2.1.19 Analyzing Function Branch

Each Analyzing Function Branch represents one available Analyzing Function.

The following specific context menu items are provided:

Context Menu Item	Description
New Analyzing Model > …	opens an **Analyzing Model Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Model, inserts the Analyzing Function from which the context menu has been called and initializes it with default values
New Analyzing Script > …	opens an **Analyzing Script Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Script, inserts the Analyzing Function from which the context menu has been called and initializes it with default values
New User Function > …	opens a **User Function Editor** of the chosen type within the **Analyzing System** workspace, creates a new User Function, inserts the Analyzing Function from which the context menu has been called and initializes it with default values

2.1.20 User Functions Branch

The User Functions Branch of each connected ***X-Tools Server*** provides all of the User Functions which are available for the **Analyzing System**. The tree of shown User Functions is updated automatically whenever a User Function at the disk is added/removed/modified.

The categories below the User Functions Branch are determined automatically according to the categories which are reported by the present User Functions and each present User Function is listed below its reported category.

2.1.21 User Function Branch

Each User Function Branch represents one available User Function.

The following specific context menu items are provided:

Context Menu Item	Description
New Analyzing Model > …	opens an **Analyzing Model Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Model, inserts the User Function from which the context menu has been called and initializes it with default values
New Analyzing Script > …	opens an **Analyzing Script Editor** of the chosen type within the **Analyzing System** workspace, creates a new Analyzing Script, inserts the User Function from which the context menu has been called and initializes it with default values
New User Function > …	opens a **User Function Editor** of the chosen type within the **Analyzing System** workspace, creates a new User Function, inserts the User Function from which the context menu has been called and initializes it with default values

2.2 Analyzing Model Editors

2.2.1 Overview

Analyzing Model Editors are used in order to edit Analyzing Models. An Analyzing Model contains all of the information which is needed in order to start a model-based (online or offline) data calculation. Multiple Analyzing Models can be maintained and/or executed at each ***X-Tools Server*** simultaneously.

Analyzing Model Editors are known and accessed exclusively by the ***X-Tools Client***, the ***X-Tools Server*** has no knowledge about **Analyzing Model Editors** at all. However, the ***X-Tools Server*** is responsible to maintain Analyzing Model Files and therefore all Analyzing Model reading and writing operations are performed via the ***X-Tools Server***.

2.2.2 Common Controls

2.2.2.1 Overview

As all **Analyzing Model Editors** are built up in a similar way, they share some common controls (e.g. tables and the menu bar) which are the same in all **Analyzing Model Editors**.

Each control of an **Analyzing Model Editor** has a defined task and provides certain functionalities. The following major controls are provided by the **Analyzing Model Editors**. Depending to the **Analyzing Model Editor**, one or more parts may not be supported (because they are not needed) and one or more parts may be present in addition to the following ones (because they are necessary):

- Analyzing Model Settings Table
- Analyzing Model Parameters Table
- Offline Time Configuration Table
- Treatment of Status Codes Table
- Log Entries Table
- Menu Bar

2.2.2.2 Analyzing Model Settings Table

The Analyzing Model Settings table contains all of the model-dependent settings which can be configured within an Analyzing Model:

Parameter	Description
Target Name	contains the name of the target to which the Analyzing Model is stored
Storage Path	contains the path to which the Analyzing Model is stored (absolute or symbolic path)
Creation Date	contains the creation date of the Analyzing Model
Modification Date	contains the last modification date of the Analyzing Model
Model Description	contains the description of the Analyzing Model
Company Name	contains the company name
Author Name	contains the author name
Data Name Prefix for Input Data	contains the data name prefix for the input data
Data Name Prefix for Output Data	contains the data name prefix for the output data

Target Name

All of the information about the Target Name and Storage Path is set up within the **Save As** dialog.

Data Name Prefix for Input Data

The Data Name Prefix for Input Data is part of the final data name of all input data which is being used by the Analyzing Model. It is being inserted at the beginning of the name of each input data before the execution of the Analyzing Model.

Data Name Prefix for Output Data

The Data Name Prefix for Output Data is part of the final data name of all output data which is being used by the Analyzing Model. It is being inserted at the beginning of the name of each output data before the execution of the Analyzing Model.

2.2.2.3 Analyzing Model Parameters Table

The Analyzing Model Parameters table contains all of the model-dependent parameters which can be configured within an Analyzing Model:

Parameter	Description
Operating Mode	allows to switch between the available operating modes
Calculation Mode	allows to switch between the available calculation modes
Action Mode	allows to switch between the available action modes
Cycle Time/Rate	contains the cycle time/rate of the Analyzing Model
Default Precision	allows to switch between the available precisions
Output Data Class	allows to switch between the available data classes
Output Time Domain	allows to switch between the available time domains
Default Input Data Interpolation Mode	allows to switch between the available interpolation modes
Default Output Data Record Mode	allows to switch between the available record modes
Default Output Data Record Parameters	allows to enter the record parameters

Operating Mode

Operating Mode	Description

Time Slice based	In case of time-slice based operating modes, the complete Analyzing Model is being executed time slice after time slice:
	• at each time slice, all (Analyzing and User) functions within the Analyzing Model are being called one after the other an in the order which is specified via their sequence ID
	  ○ each function performs exactly one calculation out of the exactly one value which is provided at each of its input arguments
	  ○ each function outputs exactly one value to each of its output arguments
	• the time of the current time slice is being called the “heart beat time”
	  ○ the heart beat time is totally independent to the system time, it can be an absolute or relative time and it can point to the past, to the present or to the future
	  ○ the heart beat time is being determined according to the parameters of the Analyzing Model
	• advanced functions may have to collect the values of multiple cycles (e.g. the FFT Analyzing Function, …) and perform their actual calculation when they have collected all needed values

Calculation Mode

Calculation Mode	Description
Equidistant	In this calculation mode, the specified calculation instructions are being executed exactly each defined time interval and with the interval which is specified via the cycle time/rate parameter.
Action based and cyclic	In this calculation mode, the specified calculation instructions are being executed exactly each defined time interval and with the interval which is specified via the cycle time/rate parameter. In opposite to the calculation mode “Equidistant”, the execution of calculations is not done in case the specified action (see Action Mode) did not take place within the last calculation interval. The calculations are done again as soon as the specified action is detected again. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.
Action based and once	In this calculation mode, the specified calculation instructions are being executed exactly at the times where the specified action (see Action Mode) takes place. Compared to the “Action based and cyclic” calculation mode, this calculation mode does not create an equidistant output data even in case the specified action is detected constantly. Instead, the resulting output data contains exactly one calculation for each time at which the specified action has been detected and therefore it can handle also high speed input data when it is being received irregularly. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.

Action Mode

Action Mode	Description
Run one Calculation when at least one Input provides a new Value	In this action mode, the specified calculation instructions are being executed whenever at least one of the inputs provides a new value. This action mode can be used especially for applications where the input data is collected irregularly (e.g. because a kind of data reduction is applied to the input data or because the source provides data irregularly by design, …) and the amount of calculated data shall be optimized (because a calculation is performed only in case when there are new values available from at least one of the inputs). For cases where the input data is received irregularly and with high speed, this action mode allows much higher calculation intervals than the calculation mode “Equidistant”. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.
Run one Calculation when all Inputs provide a new Value	In this action mode, the specified calculation instructions are being executed whenever all of the inputs provide a new value. This action mode can be used especially for applications where the input data is collected irregularly (e.g. because a kind of data reduction is applied to the input data or because the source provides data irregularly by design, …) and the amount of calculated data shall be optimized (because a calculation is performed only in case when there are new values available from all of the inputs). For cases where the input data is received irregularly and with high speed, this action mode allows much higher calculation intervals than the calculation mode “Equidistant”. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.

Cycle Time/Rate

The specified Cycle Time/Rate is being used for the determination of the successive heart beat times of the to-be executed calculations.

Default Precision

The default precision is used in order to specify the precision which is used internally by the (Analyzing and User) functions of the Analyzing Model. At the moment, only “Decimal64” is supported as default precision.

Output Data Class

Each data is uniquely identified via its data name and its data class (which can be online or offline). In order to be able to determine whether the current Analyzing Model shall produce online or offline data, the output data class can be specified and the Analyzing Model operates on base of the specified data class (thus, all input data of unspecified class is taken from the data pool which matches the specified output data class). For special applications the user is able to override the specified output data class and to specify for each of the used input data independently whether it shall be taken from the online or from the offline data pool.

Output Data Class	Description
Online Data	An executed Analyzing Model produces online data in case its “Output Data Class” parameter is set to “Online Data”. This statement is true even in case one, multiple or all of the input data is offline data. Possible cases are: • an Analyzing Model which bases only on online input data ○ all input Analyzing Functions use directly the time stamps from the (ring) buffers of their data for the determination of the values and time stamps which shall be used • an Analyzing Model which bases only on offline input data ○ when an input Analyzing Function detects that it shall read data of an offline data although the parent Analyzing Model is configured for online output data, it determines the offset in between the start time of the parent Analyzing Model and the first available time stamp of its offline data ■ at each following call of the input Analyzing Function, it reads out the value of the offline data shifted by the determined time offset ■ as the time offset is calculated for each offline data and by each input Analyzing Function independently, offline data from different periods of time are shifted to the heart beat time of the online data calculation automatically ■ in case of action-based calculation modes, all input Analyzing Functions which base on offline data return exactly the start time of the parent Analyzing Model as first possible heart beat time ○ in case of action-based calculation modes, the input Analyzing Functions can provide the heart beat times because they apply the calculated time offset to their offline input data in order to find out about the according online data time ○ when the end of the offline data has been reached, the input Analyzing Function starts to read the offline data from its beginning again • an Analyzing Model which bases on online and offline input data ○ for input Analyzing Functions it is not known whether other input Analyzing Functions within the same Analyzing Model base on online or on offline data, thus they behave like described above
Offline Data	An executed Analyzing Model produces offline data in case its “Output Data Class” parameter is set to “Offline Data”. This statement is true even in case one, multiple or all of the input data is online data. Produced offline data is returned to the system when the stop condition is reached. Possible cases are: • an Analyzing Model which bases only on offline data ○ all input Analyzing Functions use directly the time stamps from the buffers of their data for the determination of the values and time stamps which shall be used • an Analyzing Model which bases only on online data ○ when an input Analyzing Function detects that it shall read data of an online data although the parent Analyzing Model is configured for offline output data, it determines the offset in between the start time (in system time) of the parent Analyzing Model and the configured minimum time ■ at each following call of the input Analyzing Function, it reads out the value of the online data shifted by the determined time offset ■ in case of action-based calculation modes, all input Analyzing Functions which base on online data return exactly the start time of the parent Analyzing Model as first possible heart beat time ○ in case of action-based calculation modes, the input Analyzing Functions can provide the heart beat times because they apply the calculated time offset to their online input data in order to find out about the according offline data time • an Analyzing Model which is based on online and offline data ○ for input Analyzing Functions it is not known whether other input Analyzing Functions within the same Analyzing Model base on online or on offline data, thus they behave like described above

Output Time Domain

The output time domain specifies whether the timestamps of the created data have to be interpreted as absolute time in ns since 1970-01-01 00:00:00.000.000.000 (GMT) or as relative time in ns (since 0).

Output Time Domain	Description
Absolute Times	An executed Analyzing Model produces absolute time stamps in case its “Output Time Domain” parameter is set to “Absolute Times”. This statement is true even in case one, multiple or all of the input data is of relative time. Possible cases are: • an Analyzing Model which bases on data with absolute time stamps • an Analyzing Model which bases on data with relative time stamps ○ the relative time of all input data which is of relative time is converted to absolute time by the input Analyzing Functions ■ the time offset is calculated for each data which is of relative time and by each input Analyzing Function independently, so relative data from different periods of time are shifted to the absolute heart beat time of the Analyzing Model • an Analyzing Model which bases on data with relative and absolute time stamps ○ for input Analyzing Functions it is not known whether other input Analyzing Functions within the same Analyzing Model base on absolute or relative time stamps, thus they behave like described above
Relative Times	An executed Analyzing Model produces relative time stamps in case its “Output Time Domain” parameter is set to “Relative Times”. This statement is true even in case one, multiple or all of the input data is of absolute time. Possible cases are: • an Analyzing Model which bases on data with relative time stamps • an Analyzing Model which bases on data with absolute time stamps ○ the absolute time of all input data which is of absolute time is converted to relative time by the input Analyzing Functions ■ the time offset is calculated for each data which is of absolute time and by each input Analyzing Function independently, so absolute data from different periods of time are shifted to the relative heart beat time of the Analyzing Model • an Analyzing Model which bases on data with absolute and relative time stamp ○ for input Analyzing Functions it is not known whether other input Analyzing Functions within the same Analyzing Model base on absolute or relative time stamps, thus they behave like described above

Default Input Data Interpolation Mode

The chosen Default Input Data Interpolation Mode can be overwritten by each input Analyzing Function.

Default Output Data Record Mode

The chosen Default Output Data Record Mode can be overwritten by each output Analyzing Function.

Default Output Data Record Parameters

The chosen Default Output Data Record Parameters can be overwritten by each output Analyzing Function.

2.2.2.4 Offline Time Configuration Table

The Offline Time Configuration table contains all of the offline data calculation related parameters which can be configured within an Analyzing Model:

Parameter	Description
Time Input Mode	allows to switch between the available time input modes
Minimum Time Format	allows to switch between the available input formats for the minimum time
Minimum Time Value	allows to enter the minimum time of the scaling
Time Interval Format	allows to switch between the available input formats of the time interval
Time Interval Value	allows to enter the time interval of the scaling
Maximum Time Format	allows to switch between the available input formats for the maximum time
Maximum Time Value	allows to enter the maximum time of the scaling

Time Input Mode

Time Input Mode	Description
Minimum Time and Time Interval	In this time input mode, the Analyzing Model is executed from the specified minimum time and for the specified time interval. The according maximum time is calculated automatically.
Minimum Time and Maximum Time	In this time input mode, the Analyzing Model is executed from the specified minimum time to the specified maximum time.
Time Interval and Maximum Time	In this time input mode, the Analyzing Model is executed for the specified time interval and until the specified maximum time. The according minimum time is calculated automatically.

2.2.2.5 Treatment of Status Codes Table

The Treatment of Status Codes table contains all of the status-treatment dependent parameters which can be configured within an Analyzing Model:

Parameter	Description
Treatment of Status “Overflow”	contains the treatment of the value status “Overflow”
Treatment of Status “Underflow”	contains the treatment of the value status “Underflow”
Treatment of Status “Loss of Accuracy”	contains the treatment of the value status “Loss of Accuracy”
Treatment of Status “Input out of Range”	contains the treatment of the value status “Input out of Range”
Treatment of Status “Input not available”	contains the treatment of the value status “Input not available”
Treatment of Status “Disabled”	contains the treatment of the value status “Disabled”
Treatment of Status “Division by Zero”	contains the treatment of the value status “Division by Zero”
Treatment of Status “Infinite Result”	contains the treatment of the value status “Infinite Result”
Treatment of Status “Undefined Result”	contains the treatment of the value status “Undefined Result”
Treatment of Status “Function specific Error”	contains the treatment of the value status “Function specific Error”

Treatment of Status “xxx”

Treatment of Status “xxx”	Description
Ignore Status	In case a (Analyzing or User) function detects a status which shall be ignored at any of its input arguments, it continues with its internal operation and uses the value which is provided by the input argument (internally, the status of the input argument is treated as “data is valid” in this case). In case the value of the provided input argument can not be used by the calculation (e.g. out of range, …), the calculation of the function creates according status information for its output arguments.
Treat Status as Error	In case a function detects a status which shall be treated as error at any of its input arguments, it creates according status information (“input not available”) for its output arguments and does not perform its actual calculation.

2.2.2.6 Log Entries Table

The Log Entries table contains all of the events which apply to the currently opened Analyzing Model:

Parameter	Description
No.	contains the row number
Date and Time	contains the date and time of the entry
Entry	contains the text of the entry
Source Module	contains the source module of the entry
Source Machine	contains the source machine of the entry
Source User	contains the source user of the entry
Priority Class	contains the priority class of the entry
Execution ID	contains the execution ID of the entry

2.2.2.7 Menu Bar

Menu Button	Description
New	This button creates a new, empty Analyzing Model and initializes the Analyzing Model Settings table with the default values for new Analyzing Models. In case there is an Analyzing Model opened already, it is closed automatically before the new one is being created.
Open…	This button opens the **Open** dialog where the user is able to select the file which shall be opened.
Save	This button saves the currently opened Analyzing Model to the currently known storage location. In case the storage location has not been defined yet, the **Save As** dialog is popping up automatically and the user is able to select the desired storage location.
Save As…	This button opens the **Save As** dialog where the user is able to select the desired storage location.
Convert	This button converts a currently opened Analyzing Model into a User Function and initializes it with the default values within a new **User Function Editor**.
Close	This button closes the editor. In case the currently opened Analyzing Model is not saved, the editor asks the user whether the Analyzing Model shall be saved before it is closed.

2.2.3 AME Standard T001

2.2.3.1 Overview

The **AME Standard T001** is used in order to visualize, create and edit Analyzing Models of type “Standard T001”. Multiple editors of this type can be opened and used simultaneously.

The following screenshot shows an example of an **AME Standard T001**:

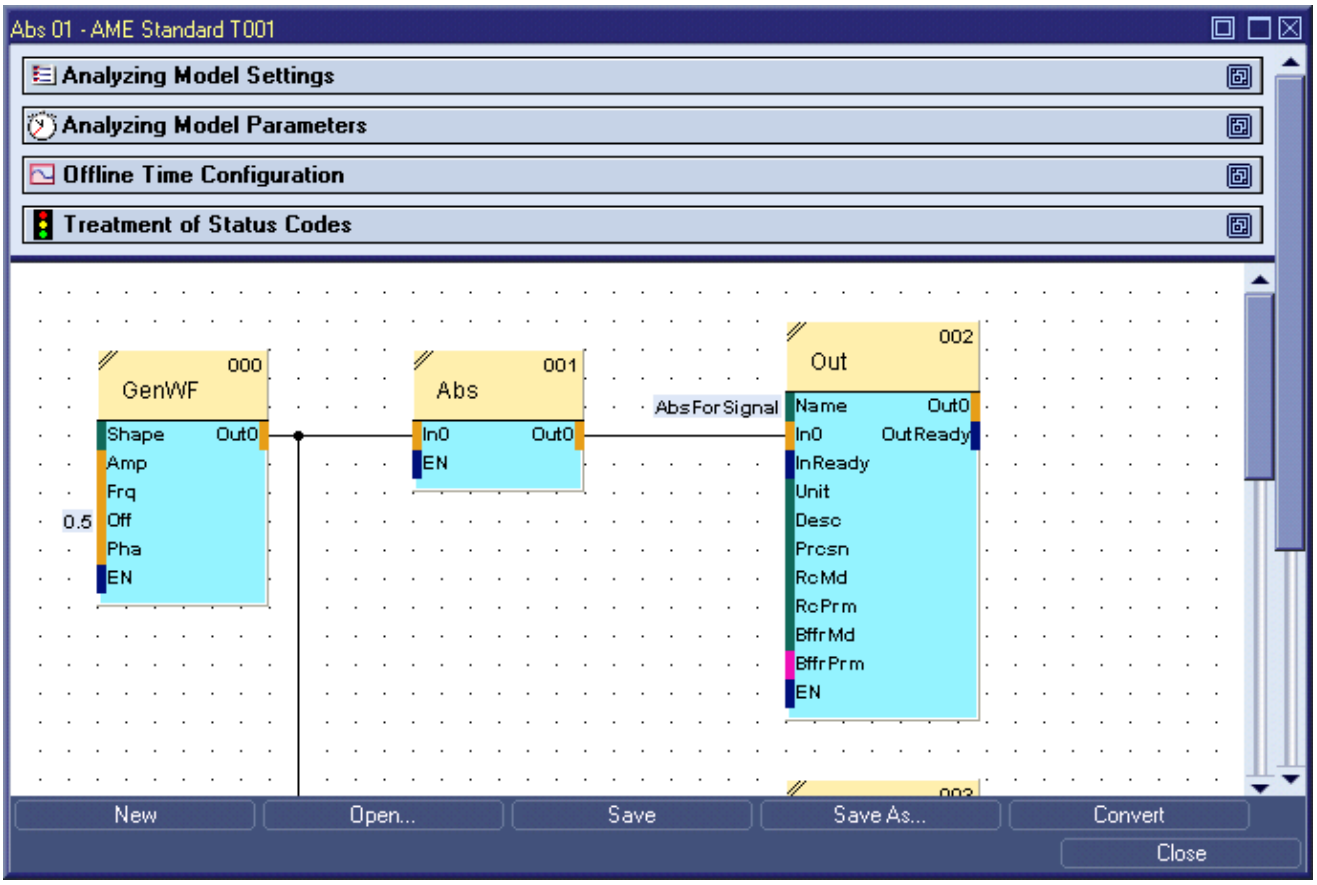


*Figure 2: Example of an **AME Standard T001***

Each control of the **AME Standard T001** has a defined task and provides certain functionalities. The following major controls are provided by the **AME Standard T001**:

- Analyzing Model Settings Table
- Analyzing Model Parameters Table
- Offline Time Configuration Table
- Treatment of Status Codes Table
- Action Area
- Log Entries Area
- Menu Bar
- Dropping of Items
- Analyzing Function Properties Dialog
- User Function Properties Dialog

2.2.3.2 Analyzing Model Settings Table

The standard Analyzing Model Settings table is being used by the **AME Standard T001** (see point 2.2.2.2).

2.2.3.3 Analyzing Model Parameters Table

The standard Analyzing Model Parameters table is being used by the **AME Standard T001** (see point 2.2.2.3).

2.2.3.4 Offline Time Configuration Table

The standard Offline Time Configuration table is being used by the **AME Standard T001** (see point 2.2.2.4).

2.2.3.5 Treatment of Status Codes Table

The standard Treatment of Status Codes table is being used by the **AME Standard T001** (see point 2.2.2.5).

2.2.3.6 Action Area

The Action Area contains all of the functions and links of the current Analyzing Model. Via Drag&Drop, the currently present functions and links can be edited and additional functions and links can be added to the Analyzing Model.

Symbol of Functions

Each present function is represented by a graphical symbol which provides the following information:

- name of the function
- sequence ID
- names of all input arguments
- names of all output arguments

Keyboard Operations

The following operation can be performed via the keyboard:

Operation	Description
<F2> in case an input argument of a function has the input focus	<F2> allows editing the value of the input argument directly within the Action Area of the **Analyzing Model Editor**. Depending to the type of the input argument, an edit box or a combo box is being displayed during the edit.
<Cursor Up> in case an input argument of a function has the input focus	moves the input focus one argument up
<Cursor Down> in case an input argument of a function has the input focus	moves the input focus one argument down

Operations via the Left Mouse Button

The following operations can be performed via the left mouse button:

Operation	Description
single click with releasing the button	A single click of the left mouse button with releasing the button above an item (function or link) selects or deselects the item below the current mouse position: • <Ctrl> can be used in order to select/deselect multiple (Analyzing and User) functions one after the other
single click with keeping the button above a selected item	A single click of the left mouse button above a selected item with keeping the button starts a move or copy operation. The actual moving or copying is performed when the mouse is moved: • [left mouse button down] + [mouse move] moves all currently selected items within the Action Area into the direction of the mouse move • <Esc> cancels the current operation and moves all items back to the positions which they had before the move operation had been started
single click with keeping the button not above a selected item	A single click of the left mouse button not above a selected item with keeping the button starts a select operation. The actual selecting is performed when the left mouse button is released: • [left mouse button down] + [mouse move] highlights all items which are within the area in between the mouse position at the start of the select operation and the current mouse position o during the mouse move and before the left mouse button is being released, a rectangular frame indicates the area of selection o all items which are completely enclosed by the rectangular frame are highlighted in order to show which items will be selected when the left mouse button is being released • <Esc> cancels the current operation and sets the selection of all items back to the state which they had before the selecting operation had been started
double click onto an Analyzing Function	A double click of the left mouse button onto any Analyzing Function opens the **Analyzing Function Properties** dialog for the Analyzing Function below the current mouse position.
double click onto an Analyzing Function argument	A double click of the left mouse button onto any argument of an Analyzing Function opens the **Analyzing Function Properties** dialog for the Analyzing Function below the current mouse position and sets the input focus to the value field of the clicked argument.
double click onto a User Function	A double click of the left mouse button onto any User Function opens the **User Function Properties** dialog for the User Function below the current mouse position.
double click onto a User Function argument	A double click of the left mouse button onto any argument of a User Function opens the **User Function Properties** dialog for the User Function below the current mouse position and sets the input focus to the value field of the clicked argument.

Operations via the Right Mouse Button

The following operation can be performed via the right mouse button:

Operation	Description
single click with releasing the button	A single click of the right mouse button with releasing the button somewhere within the Action Area opens the context menu for the below item.

Drag&Drop of a (Analyzing or User) Function

When a function is dragged into the Action Area, the size and position of the new function is visualized to the user during the Drag&Drop operation. After the drop of the dragged function, the dragged function is inserted at the specified position within the Action Area.

In case multiple functions are being selected simultaneously and dragged into the Action Area, the Action Area processes all selected functions during the Drag&Drop operation and all functions are being inserted into the Action Area simultaneously.

Drag&Drop of Data

The following operations can be performed via Drag&Drop:

Operation	Description
dropping of data into empty space	When data is dropped into empty space of the Action Area, a proper input Analyzing Function, which fits the data type of the dragged data, is searched automatically. In case a proper input Analyzing Function is available, the Drag&Drop operation behaves as if the determined input Analyzing Function would be dropped. After the drop of the dragged data, the determined input Analyzing Function is inserted and the name of the dropped data is added as input for the new Analyzing Function automatically.
dropping of data into an unclassified Analyzing Model	When data is dropped into an unclassified Analyzing Model (= an Analyzing Model whose default data class has not been set to online or offline yet), the default data class (online or offline) of the unclassified Analyzing Model is set to the class of the dragged data (online or offline) automatically. In addition, the output time domain (absolute or relative times) of the unclassified Analyzing Model is set to the time domain of the dragged data automatically.
dropping of data of different classes	Analyzing Models allow to use online and offline data simultaneously (e.g., a neural cloud (Analyzing or User) function may perform its confidence calculation for online data based on some key values that are available as offline data). Via the standard, textual online/offline identifiers of the system, the user can override the default data class of the Analyzing Model when a textual online/offline identifier is being used for the specification of an input data name.

Creation of Links between (Analyzing or User) Functions

In order to create a link from the output argument of one function to the input argument of another function, click onto one of the to-be-connected arguments first and onto the second one afterwards. After the first to-be-connected argument has being clicked, all of the available target arguments of all functions within the **Analyzing Model Editor** are being highlighted automatically.

Sequence IDs

Each (Analyzing or User) function receives a unique sequence ID which determines when it is being called during the processing of one cycle:

- the function with the lowest sequence number is being called first
- the function with the highest sequence number is being called last

The sequence ID of each function can be assigned via the **Analyzing Function Properties** and **User Function Properties** dialogs.

Context menu of a single Analyzing Function

Context Menu Item	Description
Cut	cuts the currently selected Analyzing Function
Copy	copies the currently selected Analyzing Function
Remove	removes the currently selected Analyzing Function
Expand	expands the Analyzing Function (displays the full names of all arguments)
Collapse	collapses the Analyzing Function (displays the short names of all arguments)
Show Values of Inputs > …	sets whether the values of the input arguments of the currently selected Analyzing Functions shall be displayed within the Action Area
Analyzing Function Properties…	opens the **Analyzing Function Properties** dialog

Context menu of a single User Function

Context Menu Item	Description
Cut	cuts the currently selected User Function
Copy	copies the currently selected User Function
Remove	removes the currently selected User Function
Expand	expands the User Function (displays the full names of all arguments)
Collapse	collapses the User Function (displays the short names of all arguments)
Show Values of Inputs > …	sets whether the values of the input arguments of the currently selected User Functions shall be displayed within the Action Area
User Function Properties…	opens the **User Function Properties** dialog
Open…	opens a new **User Funciton Editor** and opens the User Function within this **User Function Editor**

Context Menu of a single Link

Context Menu Item	Description
Remove	removes the currently selected link

Context Menu of multiple selected Items

Context Menu Item	Description
Cut	cuts the currently selected items
Copy	copies the currently selected items
Remove	removes the currently selected items
Expand	expands the (Analyzing and User) functions (displays the full names of the functions and of all of their arguments)
Collapse	collapses the (Analyzing and User) functions (displays the short names of the functions and of all of their arguments)
Show Values of Inputs > …	sets whether the values of the input arguments of the currently selected functions shall be displayed within the Action Area
Analyzing Function Properties…	opens the **Analyzing Function Properties** dialog of the **Analyzing Function** from which the context menu has been called
User Function Properties…	opens the **User Function Properties** dialog of the **User Function** from which the context menu has been called

Context Menu of empty Space

Context Menu Item	Description
Select all Items	selects all items (functions and links)
Paste	pastes the currently copied items
Remove all Items	removes all items (functions and links) from the Analyzing Model
Reassign Sequence IDs automatically	automatically assigning all sequence IDs so that the sequence IDs are rising from the input Analyzing Functions to the output Analyzing Functions

2.2.3.7 Log Entries Table

The standard Log Entries table is being used by the **AME Standard T001** (see point 2.2.2.6).

2.2.3.8 Menu Bar

The standard Menu Bar is being used by the **AME Standard T001** (see point 2.2.2.7).

2.2.3.9 Dropping of Items

Dropped Item	Description
Analyzing Model File Branches of type “AMF Standard T001”	opens the dropped Analyzing Model
Analyzing Function Branches in case the **AME Standard T001** is empty	creates a new Analyzing Model, initializes the target server to the parent server of the dragged Analyzing Function and drops the Analyzing Function as seen above
User Function Branches in case the **AME Standard T001** is empty	creates a new Analyzing Model, initializes the target server to the parent server of the dragged User Function and drops the User Function as seen above
Offline Data Branches in case the **AME Standard T001** is empty	creates a new Analyzing Model, initializes the target server to the parent server of the dragged offline data and afterwards it drops the offline data as seen above
Online Data Branches in case the **AME Standard T001** is empty	creates a new Analyzing Model, initializes the target server to the parent server of the dragged online data and afterwards it drops the online data as seen above
Analyzing Function Branches in case the **AME Standard T001** contains an Analyzing Model	drops the Analyzing Function as seen above
Analyzing Function Branches in case the **AME Standard T001** contains an Analyzing Model	drops the User Function as seen above
Offline Data Branches in case the **AME Standard T001** contains an Analyzing Model	drops the offline data as seen above
Online Data Branches in case the **AME Standard T001** contains an Analyzing Model	drops the online data as seen above

2.2.3.10 Analyzing Function Properties Dialog

2.2.3.10.1 Overview

The following screenshot shows an example of an **Analyzing Function Properties** dialog:

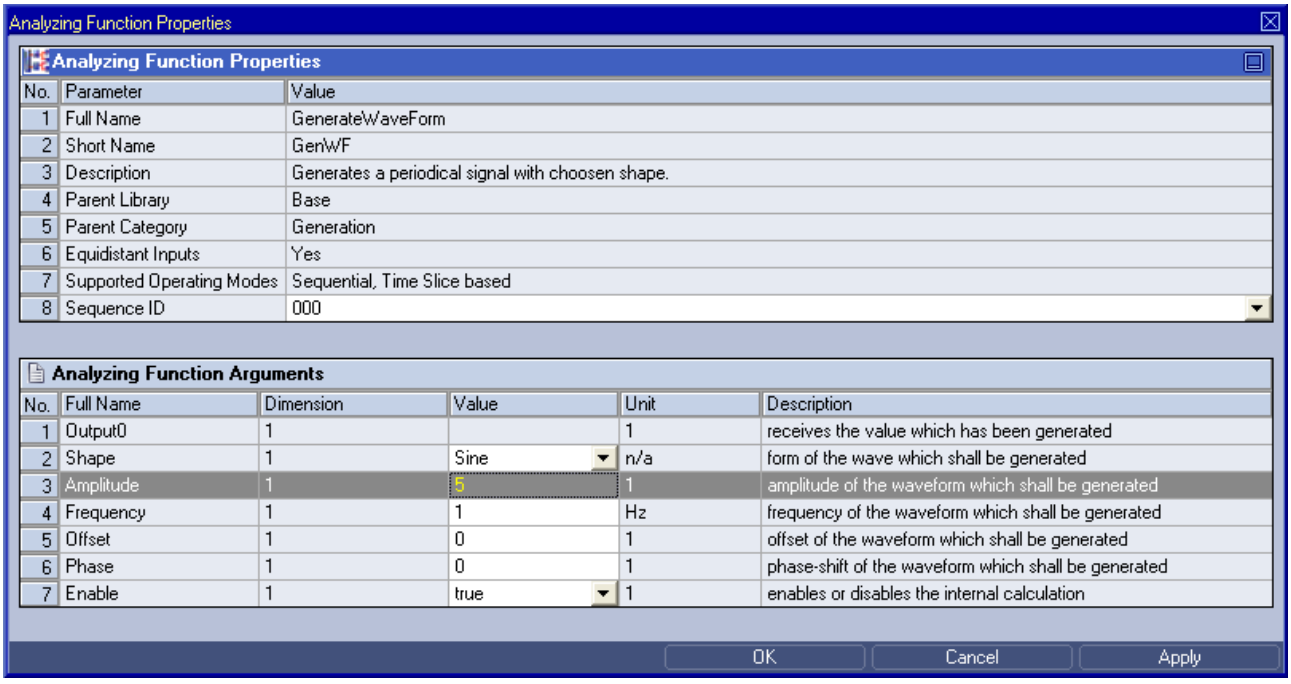


*Figure 3: Example of an **Analyzing Function Properties** Dialog*

2.2.3.10.2 Analyzing Function Properties Table

The Analyzing Function Properties table contains an overview about the properties of the Analyzing Function:

Parameter	Description
Full Name	contains the full name of the Analyzing Function
Short Name	contains the short name of the Analyzing Function
Description	contains the description of the Analyzing Function
Parent Library	contains the parent library of the Analyzing Function
Parent Category	contains the parent category of the Analyzing Function
Equidistant Inputs	indicates whether the Analyzing Function requires equidistant inputs
Supported Operating Modes	contains the supported operating modes of the Analyzing Function
Sequence ID	contains the sequence ID of the Analyzing Function

2.2.3.10.3 Analyzing Function Arguments Table

The Analyzing Function Arguments table contains an overview about the arguments of the Analyzing Function:

Parameter	Description
No.	contains the row number
Full Name	contains the full name of the argument
Short Name	contains the short name of the argument
Dimension	contains “1” when the argument is not configurable or the available number of sub-arguments in case of configurable arguments
Value	contains the value of the argument
Unit	contains the unit of the argument
Description	contains the description of the argument
Class	contains the class of the argument
Direction	contains the direction of the argument
Supported Data Types	contains the supported data types of the argument
Default Value	contains the default value of the argument

It is not possible to add or remove arguments to/from the Analyzing Function Properties table because each Analyzing Function provides detailed information about all of its arguments by itself.

Dimension

In case of configurable arguments, the Dimension cell contains a combo box with the available dimensions.

In case of arguments which are not configurable, the Dimension cell contains the dimension as static text.

Value

The following specific context menu item is provided:

Context Menu Item	Description
Set to Default	sets the value of the cell to its default value

2.2.3.11 User Function Properties Dialog

The appearing and functionality of the **User Function Properties** dialog is identical to the appearing and functionality of the **Analyzing Function Properties** dialog (see point 2.2.3.10).

2.3 Analyzing Script Editors

2.3.1 Overview

Analyzing Script Editors are used in order to edit Analyzing Scripts. An Analyzing Script contains all of the information which is needed in order to start a script-based (online or offline) data calculation. Multiple Analyzing Scripts can be maintained and/or executed at each ***X-Tools Server*** simultaneously.

Analyzing Script Editors are known and accessed exclusively by the ***X-Tools Client***, the ***X-Tools Server*** has no knowledge about **Analyzing Script Editors** at all. However, the ***X-Tools Server*** is responsible to maintain Analyzing Script Files and therefore all Analyzing Script reading and writing operations are performed via the ***X-Tools Server***.

2.3.2 Common Controls

2.3.2.1 Overview

As all **Analyzing Script Editors** are built up in a similar way, they share some common controls (e.g. tables and the menu bar) which are the same in all **Analyzing Script Editors**.

Each control of an **Analyzing Script Editor** has a defined task and provides certain functionalities. The following major controls are provided by the **Analyzing Script Editors**. Depending to the **Analyzing Script Editor**, one or more parts may not be supported (because they are not needed) and one or more parts may be present in addition to the following ones (because they are necessary):

- Analyzing Script Settings Table
- Analyzing Script Parameters Table
- Offline Time Configuration Table
- Treatment of Status Codes Table
- Log Entries Table
- Menu Bar

2.3.2.2 Analyzing Script Settings Table

The Analyzing Script Settings table contains all of the script-dependent settings which can be configured within an Analyzing Script:

Parameter	Description
Target Name	contains the name of the target to which the Analyzing Script is stored
Storage Path	contains the path to which the Analyzing Script is stored (absolute or symbolic path)
Creation Date	contains the creation date of the Analyzing Script
Modification Date	contains the last modification date of the Analyzing Script
Script Description	contains the description of the Analyzing Script
Company Name	contains the company name
Author Name	contains the author name
Data Name Prefix for Input Data	contains the data name prefix for the input data
Data Name Prefix for Output Data	contains the data name prefix for the output data

Target Name

All of the information about the Target Name and Storage Path is set up within the **Save As** dialog.

Data Name Prefix for Input Data

The Data Name Prefix for Input Data is part of the final data name of all input data which is being used by the Analyzing Model. It is being inserted at the beginning of the name of each input data before the execution of the Analyzing Model.

Data Name Prefix for Output Data

The Data Name Prefix for Output Data is part of the final data name of all output data which is being used by the Analyzing Model. It is being inserted at the beginning of the name of each output data before the execution of the Analyzing Model.

2.3.2.3 Analyzing Script Parameters Table

The Analyzing Script Parameters table contains all of the script-dependent parameters which can be configured within an Analyzing Script:

Parameter	Description
Operating Mode	allows to switch between the available operating modes
Calculation Mode	allows to switch between the available calculation modes
Action Mode	allows to switch between the available action modes
Cycle Time/Rate	contains the cycle time/rate of the Analyzing Script
Default Precision	allows to switch between the available precisions
Output Data Class	allows to switch between the available data classes
Output Time Domain	allows to switch between the available time domains
Default Input Data Interpolation Mode	allows to switch between the available interpolation modes
Default Output Data Record Mode	allows to switch between the available record modes
Default Output Data Record Parameters	allows to enter the record parameters

Operating Mode

Operating Mode	Description
Sequential	In case of sequential operating mode, the specified Analyzing Script is being executed instruction after instruction: • each instruction is applied to the complete available input data and to all of its values at once • when the last instruction of the Analyzing Script has been processed, the Analyzing Script is stopped automatically • Analyzing Scripts in sequential mode support advanced grammar, including loops, conditional statements, variables definitions and user defined functions
Time Slice based	In case of time-slice based operating modes, the complete Analyzing Model is being executed time slice after time slice: • at each time slice, all **Analyzing Functions** within the Analyzing Model are being called one after the other an in the order which is specified via their sequence ID ○ each **Analyzing Function** performs exactly one calculation out of the exactly one value which is provided at each of its input arguments ○ each **Analyzing Function** outputs exactly one value to each of its output arguments • the time of the current time slice is being called the “heart beat time” ○ the heart beat time is totally independent to the system time, it can be an absolute or relative time and it can point to the past, to the present or to the future ○ the heart beat time is being determined according to the parameters of the Analyzing Model • advanced **Analyzing Functions** may have to collect the values of multiple cycles (e.g. the FFT **Analyzing Function**, …) and perform their actual calculation when they have collected all needed values

Calculation Mode

Calculation Mode	Description
Equidistant	In this calculation mode, the specified calculation instructions are being executed exactly each defined time interval and with the interval which is specified via the cycle time/rate parameter.
Action based and cyclic	In this calculation mode, the specified calculation instructions are being executed exactly each defined time interval and with the interval which is specified via the cycle time/rate parameter. In opposite to the calculation mode “Equidistant”, the execution of calculations is not done in case the specified action (see Action Mode) did not take place within the last calculation interval. The calculations are done again as soon as the specified action is detected again. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.
Action based and once	In this calculation mode, the specified calculation instructions are being executed exactly at the times where the specified action (see Action Mode) takes place. Compared to the “Action based and cyclic” calculation mode, this calculation mode does not create an equidistant output data even in case the specified action is detected constantly. Instead, the resulting output data contains exactly one calculation for each time at which the specified action has been detected and therefore it can handle also high speed input data when it is being received irregularly. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.

Action Mode

Action Mode	Description
Run one Calculation when at least one Input provides a new Value	In this action mode, the specified calculation instructions are being executed whenever at least one of the inputs provides a new value. This action mode can be used especially for applications where the input data is collected irregularly (e.g. because a kind of data reduction is applied to the input data or because the source provides data irregularly by design, …) and the amount of calculated data shall be optimized (because a calculation is performed only in case when there are new values available from at least one of the inputs). For cases where the input data is received irregularly and with high speed, this action mode allows much higher calculation intervals than the calculation mode “Equidistant”. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.
Run one Calculation when all Inputs provide a new Value	In this action mode, the specified calculation instructions are being executed whenever all of the inputs provide a new value. This action mode can be used especially for applications where the input data is collected irregularly (e.g. because a kind of data reduction is applied to the input data or because the source provides data irregularly by design, …) and the amount of calculated data shall be optimized (because a calculation is performed only in case when there are new values available from all of the inputs). For cases where the input data is received irregularly and with high speed, this action mode allows much higher calculation intervals than the calculation mode “Equidistant”. As the calculation is not necessarily performed equidistantly, calculations which require equidistance (e.g. “FFT”, …) may produce warnings in combination with this calculation mode and may not be able to perform their calculations.

Cycle Time/Rate

The specified Cycle Time/Rate is being used for the determination of the successive heart beat times of the to-be executed calculations.

Default Precision

The default precision is used in order to specify the precision which is used internally by the **Analyzing Functions** of the Analyzing Script. At the moment, only “Decimal64” is supported as default precision.

Output Data Class

Each data is uniquely identified via its data name and its data class (which can be online or offline). In order to be able to determine whether the current Analyzing Script shall produce online or offline data, the output data class can be specified and the Analyzing Script operates on base of the specified data class (thus, all input data of unspecified class is taken from the data pool which matches the specified output data class). For special applications the user is able to override the specified output data class and to specify for each of the used input data independently whether it shall be taken from the online or from the offline data pool.

Output Data Class	Description
Online Data	An executed Analyzing Script produces online data in case its “Output Data Class” parameter is set to “Online Data”. This statement is true even in case one, multiple or all of the input data is offline data. Possible cases are: • an Analyzing Script which bases only on online input data ○ all input **Analyzing Functions** use directly the time stamps from the (ring) buffers of their data for the determination of the values and time stamps which shall be used • an Analyzing Script which bases only on offline input data ○ when an input **Analyzing Function** detects that it shall read data of an offline data although the parent Analyzing Script is configured for online output data, it determines the offset in between the start time of the parent Analyzing Script and the first available time stamp of its offline data ■ at each following call of the input **Analyzing Function**, it reads out the value of the offline data shifted by the determined time offset ■ as the time offset is calculated for each offline data and by each input **Analyzing Function** independently, offline data from different periods of time are shifted to the heart beat time of the online data calculation automatically ■ in case of action-based calculation modes, all input **Analyzing Functions** which base on offline data return exactly the start time of the parent Analyzing Script as first possible heart beat time ○ in case of action-based calculation modes, the input **Analyzing Functions** can provide the heart beat times because they apply the calculated time offset to their offline input data in order to find out about the according online data time ○ when the end of the offline data has been reached, the input **Analyzing Function** starts to read the offline data from its beginning again • an Analyzing Script which bases on online and offline input data ○ for input **Analyzing Functions** it is not known whether other input **Analyzing Functions** within the same Analyzing Script base on online or on offline data, thus they behave like described above
Offline Data	An executed Analyzing Script produces offline data in case its “Output Data Class” parameter is set to “Offline Data”. This statement is true even in case one, multiple or all of the input data is online data. Produced offline data is returned to the system when the stop condition is reached. Possible cases are: • an Analyzing Script which bases only on offline data ○ all input **Analyzing Functions** use directly the time stamps from the buffers of their data for the determination of the values and time stamps which shall be used • an Analyzing Script which bases only on online data ○ when an input **Analyzing Function** detects that it shall read data of an online data although the parent Analyzing Script is configured for offline output data, it determines the offset in between the start time (in system time) of the parent Analyzing Script and the configured minimum time ■ at each following call of the input **Analyzing Function**, it reads out the value of the online data shifted by the determined time offset ■ in case of action-based calculation modes, all input **Analyzing Functions** which base on online data return exactly the start time of the parent Analyzing Script as first possible heart beat time ○ in case of action-based calculation modes, the input **Analyzing Functions** can provide the heart beat times because they apply the calculated time offset to their online input data in order to find out about the according offline data time • an Analyzing Script which is based on online and offline data ○ for input **Analyzing Functions** it is not known whether other input **Analyzing Functions** within the same Analyzing Script base on online or on offline data, thus they behave like described above

Output Time Domain

The output time domain specifies whether the timestamps of the created data have to be interpreted as absolute time in ns since 1970-01-01 00:00:00.000.000.000 (GMT) or as relative time in ns (since 0).

Output Time Domain	Description
Absolute Times	An executed Analyzing Script produces absolute time stamps in case its “Output Time Domain” parameter is set to “Absolute Times”. This statement is true even in case one, multiple or all of the input data is of relative time. Possible cases are: • an Analyzing Script which bases on data with absolute time stamps • an Analyzing Script which bases on data with relative time stamps ○ the relative time of all input data which is of relative time is converted to absolute time by the input **Analyzing Functions** ■ the time offset is calculated for each data which is of relative time and by each input **Analyzing Function** independently, so relative data from different periods of time are shifted to the absolute heart beat time of the Analyzing Script • an Analyzing Script which bases on data with relative and absolute time stamps ○ for input **Analyzing Functions** it is not known whether other input **Analyzing Functions** within the same Analyzing Script base on absolute or relative time stamps, thus they behave like described above
Relative Times	An executed Analyzing Script produces relative time stamps in case its “Output Time Domain” parameter is set to “Relative Times”. This statement is true even in case one, multiple or all of the input data is of absolute time. Possible cases are: • an Analyzing Script which bases on data with relative time stamps • an Analyzing Script which bases on data with absolute time stamps ○ the absolute time of all input data which is of absolute time is converted to relative time by the input **Analyzing Functions** ■ the time offset is calculated for each data which is of absolute time and by each input **Analyzing Function** independently, so absolute data from different periods of time are shifted to the relative heart beat time of the Analyzing Script • an Analyzing Script which bases on data with absolute and relative time stamp ○ for input **Analyzing Functions** it is not known whether other input **Analyzing Functions** within the same Analyzing Script base on absolute or relative time stamps, thus they behave like described above

Default Input Data Interpolation Mode

The chosen Default Input Data Interpolation Mode can be overwritten by each input **Analyzing Function**.

Default Output Data Record Mode

The chosen Default Output Data Record Mode can be overwritten by each output **Analyzing Function**.

Default Output Data Record Parameters

The chosen Default Output Data Record Parameters can be overwritten by each output **Analyzing Function**.

2.3.2.4 Offline Time Configuration Table

The Offline Time Configuration table contains all of the offline data calculation related parameters which can be configured within an Analyzing Script:

Parameter	Description
Time Input Mode	allows to switch between the available time input modes
Minimum Time Format	allows to switch between the available input formats for the minimum time
Minimum Time Value	allows to enter the minimum time of the scaling
Time Interval Format	allows to switch between the available input formats of the time interval
Time Interval Value	allows to enter the time interval of the scaling
Maximum Time Format	allows to switch between the available input formats for the maximum time
Maximum Time Value	allows to enter the maximum time of the scaling

Time Input Mode

Time Input Mode	Description
Minimum Time and Time Interval	In this time input mode, the Analyzing Script is executed from the specified minimum time and for the specified time interval. The according maximum time is calculated automatically.
Minimum Time and Maximum Time	In this time input mode, the Analyzing Script is executed from the specified minimum time to the specified maximum time.
Time Interval and Maximum Time	In this time input mode, the Analyzing Script is executed for the specified time interval and until the specified maximum time. The according minimum time is calculated automatically.

2.3.2.5 Treatment of Status Codes Table

The Treatment of Status Codes table contains all of the status-treatment dependent parameters which can be configured within an Analyzing Script:

Parameter	Description
Treatment of Status “Overflow”	contains the treatment of the value status “Overflow”
Treatment of Status “Underflow”	contains the treatment of the value status “Underflow”
Treatment of Status “Loss of Accuracy”	contains the treatment of the value status “Loss of Accuracy”
Treatment of Status “Input out of Range”	contains the treatment of the value status “Input out of Range”
Treatment of Status “Input not available”	contains the treatment of the value status “Input not available”
Treatment of Status “Disabled”	contains the treatment of the value status “Disabled”
Treatment of Status “Division by Zero”	contains the treatment of the value status “Division by Zero”
Treatment of Status “Infinite Result”	contains the treatment of the value status “Infinite Result”
Treatment of Status “Undefined Result”	contains the treatment of the value status “Undefined Result”
Treatment of Status “Function specific Error”	contains the treatment of the value status “Function specific Error”

Treatment of Status “xxx”

Treatment of Status “xxx”	Description
Ignore Status	In case an **Analyzing Function** detects a status which shall be ignored at any of its input arguments, it continues with its internal operation and uses the value which is provided by the input argument (internally, the status of the input argument is treated as “data is valid” in this case). In case the value of the provided input argument can not be used by the calculation (e.g. out of range, …), the calculation of the **Analyzing Function** creates according status information for its output arguments.
Treat Status as Error	In case an **Analyzing Function** detects a status which shall be treated as error at any of its input arguments, it creates according status information (“input not available”) for its output arguments and does not perform its actual calculation.

2.3.2.6 Log Entries Table

The Log Entries table contains all of the events which apply to the currently opened Analyzing Script:

Parameter	Description
No.	contains the row number
Date and Time	contains the date and time of the entry
Entry	contains the text of the entry
Source Module	contains the source module of the entry
Source Machine	contains the source machine of the entry
Source User	contains the source user of the entry
Priority Class	contains the priority class of the entry
Execution ID	contains the execution ID of the entry

2.3.2.7 Menu Bar

Menu Button	Description
New	This button creates a new, empty Analyzing Script and initializes the Analyzing Script Settings table with the default values for new Analyzing Scripts. In case there is an Analyzing Script opened already, it is closed automatically before the new one is being created.
Open…	This button opens the **Open** dialog where the user is able to select the file which shall be opened.
Save	This button saves the currently opened Analyzing Script to the currently known storage location. In case the storage location has not been defined yet, the **Save As** dialog is popping up automatically and the user is able to select the desired storage location.
Save As…	This button opens the **Save As** dialog where the user is able to select the desired storage location.
Close	This button closes the editor. In case the currently opened Analyzing Script is not saved, the editor asks the user whether the Analyzing Script shall be saved before it is closed.

2.3.3 ASE Standard T001

2.3.3.1 Overview

The **ASE Standard T001** is used in order to visualize, create and edit Analyzing Scripts of type “Standard T001”. Multiple editors of this type can be opened and used simultaneously.

The following screenshot shows an example of an **ASE Standard T001**:

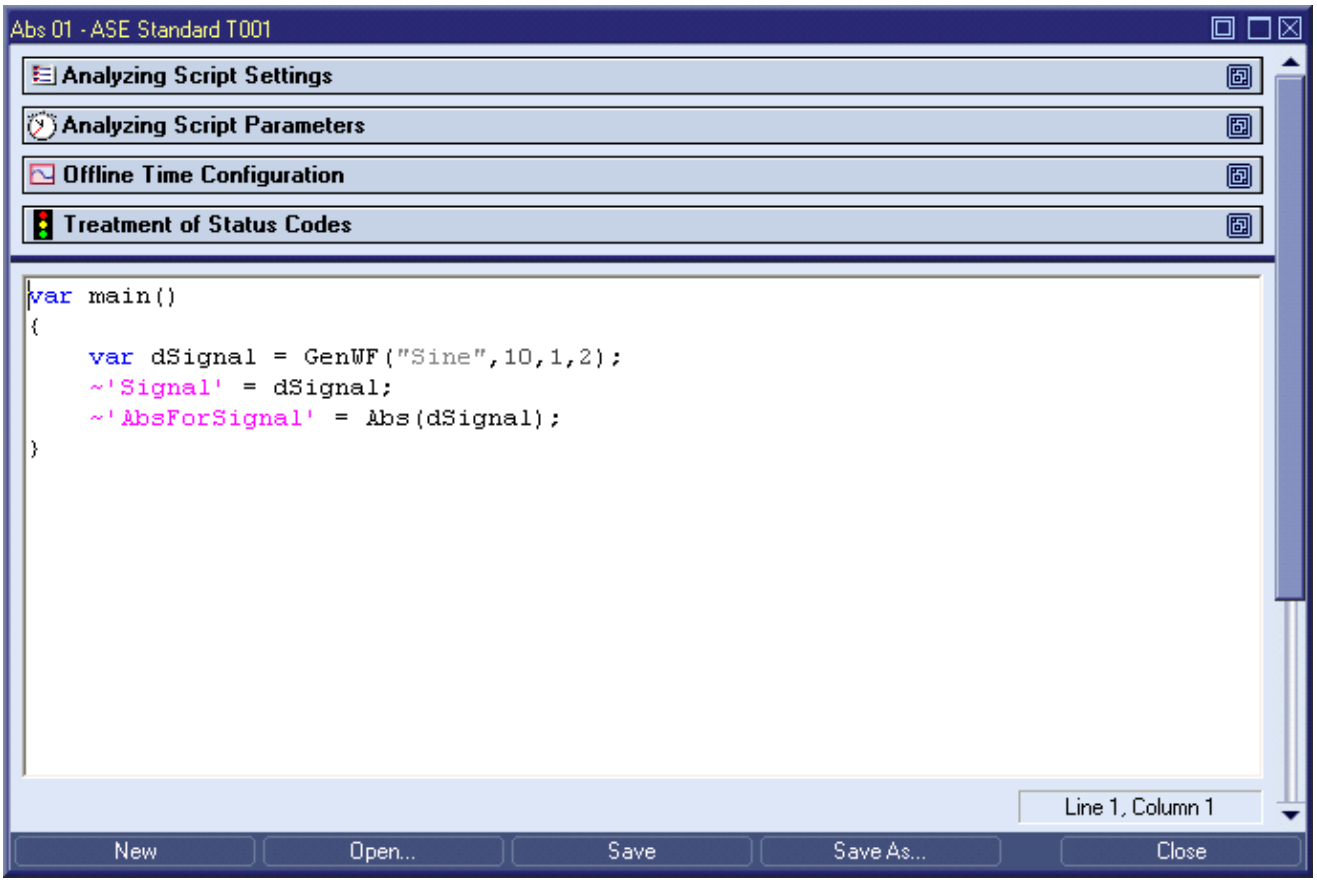


*Figure 4: Example of an **ASE Standard T001***

Each control of the **ASE Standard T001** has a defined task and provides certain functionalities. The following major controls are provided by the **ASE Standard T001**:

- Analyzing Script Settings Table
- Analyzing Script Parameters Table
- Offline Time Configuration Table
- Treatment of Status Codes Table
- Action Area
- Log Entries Area
- Menu Bar
- Dropping of Items

2.3.3.2 Analyzing Script Settings Table

The standard Analyzing Script Settings table is being used by the **ASE Standard T001** (see point 2.3.2.2).

2.3.3.3 Analyzing Script Parameters Table

The standard Analyzing Script Parameters table is being used by the **ASE Standard T001** (see point 0).

2.3.3.4 Offline Time Configuration Table

The standard Offline Time Configuration table is being used by the **ASE Standard T001** (see point 2.3.2.4).

2.3.3.5 Treatment of Status Codes Table

The standard Treatment of Status Codes table is being used by the **ASE Standard T001** (see point 2.3.2.5).

2.3.3.6 Action Area

The Action Area contains the current Analyzing Script. The currently present Analyzing Script can be edited here and Drag&Drop can be used in order to add additional **Analyzing Functions** to the Analyzing Script.

The current row and column of the text cursor position within the current Analyzing Script is displayed at the bottom of the Action Area.

Operations via the Left Mouse Button

The following operations can be performed via the left mouse button:

Operation	Description
single click with releasing the button	A single click of the left mouse button with releasing the button moves the current text cursor to the clicked position.
single click with keeping the button above a selected text	A single click of the left mouse button above a selected item with keeping the button starts a move operation. The actual moving is performed when the mouse is released: • [left mouse button down] + [mouse move] moves the currently selected text within the Action Area to the position which is indicated by the moved mouse cursor • <Esc> cancels the current operation and moves all texts back to the positions which they had before the move operation had been started
single click with keeping the button not above a selected text	A single click of the left mouse button not above a selected item with keeping the button starts a select operation. The actual selecting is performed when the left mouse button is released: • [left mouse button down] + [mouse move] highlights all text which are within the area in between the mouse position at the start of the select operation and the current mouse position

Drag&Drop of an Analyzing Function

When an **Analyzing Function** is dropped into the Action Area, it is being inserted at the current text cursor position.

In case multiple **Analyzing Functions** are being selected simultaneously and dragged into the Action Area, the Action Area processes only the first selected **Analyzing Function** during all Drag&Drop operations. Drag&Drop of multiple **Analyzing Functions** at once is not being supported.

Drag&Drop of Data

The following operations can be performed via Drag&Drop:

Operation	Description
dropping of data into the Action Area	When data is dragged into the Action Area, its name is being inserted at the current keyboard cursor position. In case multiple data are being selected simultaneously and dragged into the Action Area, the Action Area processes the selected data during all Drag&Drop operations
dropping of data into an unclassified Analyzing Script	When data is dropped into an unclassified Analyzing Script (= an Analyzing Script whose default data class has not been set to online or offline yet), the default data class (online or offline) of the unclassified Analyzing Script is set to the class of the dragged data (online or offline) automatically. In addition, the output time domain (absolute or relative times) of the unclassified Analyzing Script is set to the time domain of the dragged data automatically.
dropping of data of different classes	Analyzing Scripts allow to use online and offline data simultaneously (e.g., a neural cloud **Analyzing Function** may perform its confidence calculation for online data based on some key values that are available as offline data). Via the standard, textual online/offline identifiers of the system, the user can override the default data class of the Analyzing Script when a textual online/offline identifier is being used for the specification of an input data name.

2.3.3.7 Log Entries Table

The standard Log Entries table is being used by the **ASE Standard T001** (see point 2.3.2.6).

2.3.3.8 Menu Bar

The standard Menu Bar is being used by the **ASE Standard T001** (see point 2.3.2.7).

2.3.3.9 Dropping of Items

Dropped Item	Description
Analyzing Script File Branches of type “ASF Standard T001”	opens the dropped Analyzing Script
Analyzing Function Branches in case the **ASE Standard T001** is empty	creates a new Analyzing Script, initializes the target server to the parent server of the dragged Analyzing Function and drops the Analyzing Function as seen above
User Function Branches in case the **ASE Standard T001** is empty	creates a new Analyzing Script, initializes the target server to the parent server of the dragged User Function and drops the User Function as seen above
Offline Data Branches in case the **ASE Standard T001** is empty	creates a new Analyzing Script, initializes the target server to the parent server of the dragged offline data and afterwards it drops the offline data as seen above
Online Data Branches in case the **ASE Standard T001** is empty	creates a new Analyzing Script, initializes the target server to the parent server of the dragged online data and afterwards it drops the online data as seen above
Analyzing Function Branches in case the **ASE Standard T001** contains an Analyzing Script	drops the Analyzing Function as seen above
User Function Branches in case the **ASE Standard T001** contains an Analyzing Script	drops the User Function as seen above
Offline Data Branches in case the **ASE Standard T001** contains an Analyzing Script	drops the offline data as seen above
Online Data Branches in case the **ASE Standard T001** contains an Analyzing Script	drops the online data as seen above

2.3.3.10 Syntax of Analyzing Scripts

2.3.3.10.1 Overview

The syntax of Analyzing Scripts is similar to C. It provides the following features:

- Basics
- Variables
- Functions
- Conditional Statements
- Loops

2.3.3.10.2 Basics

2.3.3.10.2.1 main()

Each Analyzing Script starts and ends within the main() function:

```
var main()
{

}
```

The processing starts with the first instruction within main() and ends with the last instruction within main(). In case of sequential Analyzing Scripts the Analyzing Script is stopped automatically after the last instruction has been processed. Time-slice based Analyzing Scripts continue to execute the first instruction again after the last instruction has finished.

2.3.3.10.2.2 Comments

Comments can be used in order to add remarks to the Analyzing Script which shall be ignored by the calculation. A sequence of “//” comments out the rest of the current line, where as a pair of “/*” and “*/” comments out all characters in between:

```
var main()
{
    // this is a comment which covers one line
    var bNQ, bQ;

    bQ = RSFF($'Binary Data 1',$'Binary Data 2',true,bNQ);

    /* the following calculations are commented out for now
    Out("Q",bQ,true,"","","Boolean");
    Out("QN",bNQ,true,"","","Boolean");
    */
}
```

2.3.3.10.3 Variables

2.3.3.10.3.1 Overview

There are three kinds of variables available:

- Generic Variables
- Variables which represent online Data
- Variables which represent offline Data
- Constants

2.3.3.10.3.2 Generic Variables

Generic variables can be used in order to store any kind of information, e.g. a single value, a string or a complete (online or offline) data. Examples:

```
var main()
{
    var Value  = 10.123;
    var String = "Hello World";
    var Data   = GenWF("Sine",10,1,2);
}
```

2.3.3.10.3.3 Variables which represent online Data

In case an online data shall be read from the system, the following syntax must be used:

```
var main()
{
    var Data = ~'My online Data';
}
```

The same syntax also can be used in order to write online data back to the system:

```
var main()
{
    var Data = ~'My online Data';
    ~'Output Data' = Data;
}
```

Within time slice based operating modes, ~'My online Data' and ~'Output Data' effectively always contain only one value (the one which belongs to the current time slice).

2.3.3.10.3.4 Variables which represent offline Data

Variables for offline data can be used in the same way as variables for online data. In a simple example the Analyzing Script outputs the specified input data directly as output data:

```
var main()
{
    $'Output Data' = $'My offline Data';
}
```

Within time slice based operating modes, $'My offline Data' and $'Output Data' effectively always contain only one value (the one which belongs to the current time slice).

2.3.3.10.3.5 Constants

The "const" keyword must be used for variables whose value is unchangeable. An example for the usage of "const" are variables for input arguments of Analyzing Functions which are of type "Parameter" - these Analyzing Functions do not accept non-constant values as input of their "Parameter" arguments:

```
var main()
{
    const MyConst = 1024;

    Bffr(3,MyConst);
}
```

2.3.3.10.4 Functions

2.3.3.10.4.1 Overview

There are five kinds of functions available:

- Internal Operators
- Internal Functions
- Built-in Analyzing Functions
- User-defined User Functions
- Local Functions

2.3.3.10.4.2 Internal Operators

2.3.3.10.4.2.1 Overview

Internal operators can be used in order to access frequently used operations in a fast way. The following internal operators are available:

- Arithmetic Operators
- Compound Assignment Operators
- Logical Operators
- Comparison Operators
- Other Operators

2.3.3.10.4.2.2 Arithmetic Operators

The following arithmetic operators are available:

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)
- ++ (as prefix increment and as postfix increment)
- -- (as prefix decrement and as postfix decrement)
- + (unary plus)
- - (unary minus)

Example:

```
var main()
{
    var Value = 0;

    $'Add' = $'My offline Data' + 10.0;               // addition
    $'Sub' = $'My offline Data' - 3;                  // subtraction
    $'Mul' = $'My offline Data' * $'Sub';             // multiplication
    $'Div' = $'My offline Data' / $'My offline Data'; // division
    $'Mod' = $'My offline Data' % 50;                 // modulo

    Value++;           // postfix increment
    ++Value;           // prefix increment
    Value--;           // postfix decrement
    --Value;           // prefix decrement

    Value = +1;        // unary plus
    Value = -1;        // unary minus
}
```

2.3.3.10.4.2.3 Compound Assignment Operators

The following compound assignment operators are available:

- += (addition assignment)
- -= (subtraction assignment)
- *= (multiplication assignment)
- /= (division assignment)

Example:

```
var main()
{
    var Value = $'My offline Data';

    Value += 10.0;      // addition
    Value -= 3;         // subtraction
    Value *= 3;         // multiplication
    Value /= 7;         // division

    $'My offline Data 66' = Value;
}
```

2.3.3.10.4.2.4 Logical Operators

The following logical operators are available:

- ! (negation)

Example:

```
var main()
{
    var Value = 0;

    Value = !Value;     // negation
}
```

2.3.3.10.4.2.5 Comparison Operators

The following comparison operators are available:

- == (equal to)
- != (not equal to)
- > (greater than)
- >= (greater than or equal to)
- < (smaller than)
- <= (smaller than or equal to)

Example:

```
var main()
{
    var Value = $'My offline Data';

    Print(FrmtStr("Value is: {0}",Value));

    if(Value==5)
    {
        Print("Value is equal to 5");
    }

    if(Value!=5)
    {
        Print("Value is not equal to 5");
    }

    if(Value>5)
    {
        Print("Value is greater than 5");
    }

    if(Value>=5)
    {
        Print("Value is greater than or equal to 5");
    }

    if(Value<5)
    {
        Print("Value is smaller than 5");
    }

    if(Value<=5)
    {
        Print("Value is smaller than or equal to 5");
    }
}
```

2.3.3.10.4.2.6 Other Operators

The following other operators are available:

- [] (array subscript)
- | (concatenation)
- |= (concatenation assignment)

Example:

```
var main()
{
    var Array[4] = {0,2,4,6};          // declaration of an array
    var String = "Hello " | "World";   // concatenation
    var i;

    for(i=0;i<4;i++)
    {
        Print(FrmtStr("Index: {0}, Value: {1}",i,Array[i]));   // array subscript
    }

    String |= "!!!";   // concatenation assignment
    Print(String);     // outputs "Hello World!!!"
 }
```

2.3.3.10.4.3 Internal Functions

2.3.3.10.4.3.1 Overview

The following internal functions are available:

- Print()
- Implicit Inputs
- Implicit Outputs

2.3.3.10.4.3.2 Print()

Example:

```
var main()
{
    if(IsDataPrsnt($'My offline Data'))
    {
        Print("The following data is present: 'My offline Data'");
    }
    else
    {
        Print("The following data is not present: 'My offline Data'");
    }
}
```

In combination with the Analyzing Function FrmtStr(), Print() also can output current data values to the system log:

```
var main()
{
    var LastValue;
    var MyPI = 3.1415927;

    if( IsDataPrsnt($'My offline Data') )
    {
        Print("The following data is present: 'My offline Data'");

        LastValue = GetVFrmSD($'My offline Data');
        Print(FrmtStr("Last value: {0}, PI: {1}",LastValue,MyPI));
    }
}
```

2.3.3.10.4.3.3 Implicit Inputs

Whenever the name of an online or offline data in form `~'My online Data'` or `$'My offline Data'` is being used as input directly, an implicit input is being called. Internally, the according Analyzing Function (e.g. In(), InB1D(), InB2D(), …) is being used with default parameters in order to read the values of the data from the system.

The following Analyzing Script uses implicit inputs:

```
var main()
{
    var Data = ~'My online Data';
}
```

The above Analyzing Script is equivalent to the following one, which calls the Analyzing Function In() explicitly:

```
var main()
{
    var Data = In("My online data");
}
```

Using of implicit inputs is very useful in case reading of data values from the system shall be performed with the default settings of the input Analyzing Function. Explicit calling of the input Analyzing Function allows to modify the default settings of the input Analyzing Function and also allows to elegantly switch the Analyzing Script between operating on online or offline data.

2.3.3.10.4.3.4 Implicit Outputs

Whenever the name of an online or offline data in form `~'My online Data'` or `$'My offline Data'` is being used as output directly, an implicit output is being called. Internally, the according Analyzing Function (e.g. Out(), OutB1D(), OutB2D(), …) is being used with default parameters in order to write the values of the data to the system.

The following Analyzing Script uses implicit inputs:

```
var main()
{
    var Data = GenWF();
    ~'My online Data' = Data;
}
```

The above Analyzing Script is equivalent to the following one, which calls the Analyzing Function Out() explicitly:

```
var main()
{
    var Data = GenWF();
    Out("My online Data",Data);
}
```

Using of implicit outputs is very useful in case writing of data values to the system shall be performed with the default settings of the output Analyzing Function. Explicit calling of the output Analyzing Function allows to modify the default settings of the output Analyzing Function and also allows to elegantly switch the Analyzing Script between operating on online or offline data.

2.3.3.10.4.4 Built-in Analyzing Functions

The built-in Analyzing Functions which are available at each ***X-Tools Server*** are found within the **ANS Explorer** in the following path:

- Main -> Servers -> *YourServerName* -> Libraries -> Analyzing Functions

The following example shows how to call a built-in Analyzing Function:

```
var main()
{
    $'Output Data' = Abs($'My offline Data');
}
```

The "," character separates the single arguments which are passed to an Analyzing Function. Some Analzying Functions (e.g. AND(), …) accept a variable number of input arguments. At the end of the last value of a variable input argument, the ";" character must be placed instead of the standard "," in order to tell the Analyzing Script that the following value will belong to the following argument (instead of being another value of the current, variable argument).

In case of Analyzing Functions which return more than one output, the variables for all except the first output must be provided as additional parameters:

```
var main()
{
    var bNQ, bQ;

    bQ = RSFF($'Binary Data 1',$'Binary Data 2',true,bNQ);

    // output the values of bQ and bNQ
    Out("Q",bQ,true,"","","Boolean");
    Out("QN",bNQ,true,"","","Boolean");
}
```

Arguments at the end of a called Analyzing Function can be omitted when they are not needed. The default value is being used for each argument which has not been passed explicitly.

2.3.3.10.4.5 User-defined User Functions

The user-defined User Functions which are available at each ***X-Tools Server*** are found within the **ANS Explorer** in the following path:

- Main -> Servers -> *YourServerName* -> Libraries -> User Functions

The syntax for using of User Functions and the rest of the behavior of User Functions is identical to the usage of Analyzing Functions:

```
var main()
{
    MyFFT("NameOfMyInputData");
}
```

2.3.3.10.4.6 Local Functions

Local functions can be used in order to maintain functionalities which are used by several places within the Analyzing Script:

```
var Normalize(var InputData, var OutputData, var M, var B)
{
    var NormalizedData;

    // get the input data from the system and normalize it
    NormalizedData = In(InputData) * M + B;

    // output the calculated data to the system
    Out(OutputData,NormalizedData);
}


var main()
{
    Normalize("NameOfMyInputData1","NameOfMyOutputData1",10.0,5.0);
    Normalize("NameOfMyInputData2","NameOfMyOutputData2",20.0,0);
}
```

Local functions also can return a value or data to their caller. The example from above can be realized also like this:

```
var Normalize(var InputData, var M, var B)
{
    // get the input data from the system, normalize it and return the result
    return In(InputData) * M + B;
}


var main()
{
    $'NameOfMyOutputData1' = Normalize("NameOfMyInputData1",10.0,5.0);
    $'NameOfMyOutputData2' = Normalize("NameOfMyInputData2",20.0,0);
}
```

2.3.3.10.5 Conditional Statements

2.3.3.10.5.1 Overview

The following conditional statements are available:

- if() - else
- switch()

2.3.3.10.5.2 if() - else

Example:

```
var main()
{
    if(IsDataPrsnt($'My offline Data'))
    {
        // this branch is entered in case the expression within the brackets of if() is not 0 (true)
        Print("The following data is present: 'My offline Data'");
    }
    else
    {
        // this branch is entered in case the expression within the brackets of if() is 0 (false)
        Print("The following data is not present: 'My offline Data'");
    }
}
```

The “else“ branch is not necessary and can be omitted, like in the following example:

```
var main()
{
    if(IsDataPrsnt($'My offline Data'))
    {
        Print("The following data is present: 'My offline Data'");
    }
}
```

The behavior of the “if()” statement can be different between the time slice based and sequential operating modes. For example, the following Analyzing Script can not run in time-slice based operating mode because in theory the condition of “if()” (= true or false) could change with each time slice, in which case there would be two different output Analyzing Functions which would want to write to the same data of the system:

```
var main()
{
    var Rand = Random(0,10);
    var Data1 = $'My offline Data 1';
    var Data2 = $'My offline Data 2';

    if(Rand<5)
    {
        Out("My offline Data",Data1);
    }
    else
    {
        Out("My offline Data",Data2);
    }
}
```

Contrary, the same Analyzing Script runs perfectly in sequential operating mode, because the sequential Analyzing Script is running only once from the beginning to its end, thus it processes either only the “if()” or only the “else” branch and only one of them needs to output data to the system.

In order to run the above Analyzing Script in time slice based operating mode, the two outputs must write to different data. Example:

```
var main()
{
    var Rand = Random(0,10);
    var Data1 = $'My offline Data 1';
    var Data2 = $'My offline Data 2';

    if(Rand<5)
    {
        Out("My offline Data 1",Data1);
    }
    else
    {
        Out("My offline Data 2",Data2);
    }
}
```

2.3.3.10.5.3 switch()

The “switch()” statement can be used in order to execute different execution paths depending to the current value of the provided condition. Example:

```
var main()
{
    var Value = Round(Random(1,10));

    switch(Value)
    {
        case 1:
        {
            // we end up here in case "Value" contains "1"
            Print("Current Value: 1");
            break;
        }
        case 2:
        {
            // we end up here in case "Value" contains "2"
            Print("Current Value: 2");
            break;
        }
        default:
        {
            // we end up here in case "Value" contains anything else
            Print(FrmtStr("Current Value: {0}",Value));
            break;
        }
    }
}
```

At the end of each “case” branch, typically the “break” condition must be placed. In case “break” is missing, the execution continues automatically with the following “case” branch. The “default” branch can be used in order to handle all values which are not handled by any of the previous “case” branches.

2.3.3.10.6 Loops

2.3.3.10.6.1 Overview

The following loops are available:

- for()
- while()
- do - while()

Like conditional statements, also loops can show different behavior depending to whether they are used in time slice based operating modes or in the sequential operating mode. Generally speaking, most applications which include loops are based on sequential operating modes, because for time slice based Analyzing Scripts the usage of loops is more limited. Examples:

- Reading from or writing to buffers. As the size of the internal buffers can not change, time slice based Analyzing Scripts may run into invalid states. Contrary, sequential Analyzing Scripts do the reading/writing only once, thus they can't run into this issue.
- Variable data names. In case the name of the output data is created by some Analyzing Function within the Analyzing Script, each time slice could create another data name. Within sequential Analyzing Scripts this is less possible because the whole Analyzing Script is being processed only once.

2.3.3.10.6.2 for()

The "for()" loop can be used in order to run a certain amount of interations. Within the brackets, three parameters are passed to "for()" - the semicolon (";") is being used in order to separate these three parameters:

- the first parameter contains the initial value of the to-be-examined variable
- the second parameter contains the break condition, how long the loop shall run (the loop runs as long as this condition returns not 0 (not 0 = true))
- the third parameter can be used in order change the to-be-examined variable automatically at the end of each loop

The following example shows how to use a "for()" loop in order to automatically process several input which contain a counter as part of their name (the names of the present input data must be "My offline Data 1", "My offline Data 2", "My offline Data 3" and "My offline Data 4" for running of the following example - the Analyzing Script automatically generates the data names for each iteration):

```
var main()
{
    var i;
    var InputData;
    var OutputData;

    for(i=1;i<=4;i++)
    {
        // prepare the names of the input and output data for this interation of the loop
        InputData  = FrmtStr("My offline Data {0}",i);
        OutputData = FrmtStr("My result {0}",i);

        // do something with the determined input and output data
        Out(OutputData,In(InputData));
    }
}
```

Instead of a predefined, hard-coded value, also variables can be used for the parameters of the “for()” loop:

```
var main()
{
    var i;
    var Start = Round(Random(1,2));
    var Max   = Round(Random(3,4));
    var InputData;
    var OutputData;

    for(i=Start;i<=Max;i++)
    {
        // prepare the names of the input and output data for this interation of the loop
        InputData  = FrmtStr("My offline Data {0}",i);
        OutputData = FrmtStr("My result {0}",i);

        // do something with the determined input and output data
        Out(OutputData,In(InputData));
    }
}
```

There also can be cases where the condition to abort the loop is not known at the line when the “for()” is configured. For such cases, the “break” instruction can be used anywhere within the “for()” loop in order to exit the loop immediately:

```
var main()
{
    var i;
    var Start = Round(Random(1,2));
    var Max   = Round(Random(6,8));
    var InputData;
    var OutputData;

    for(i=Start;i<=Max;i++)
    {
        if(i>4)
        {
            // this instruction jumps out of the loop
            break;
        }

        // prepare the names of the input and output data for this interation of the loop
        InputData  = FrmtStr("My offline Data {0}",i);
        OutputData = FrmtStr("My result {0}",i);

        // do something with the determined input and output data
        Out(OutputData,In(InputData));
    }
}
```

In addition to “break”, also “continue” can be called within a “for()” loop. Instead of jumping out from the “for()” loop, “continue” jumps back to the beginning of the loop:

```
var main()
{
    var i;
    var Start = Round(Random(1,2));
    var Max   = Round(Random(6,8));
    var InputData;
    var OutputData;

    for(i=Start;i<=Max;i++)
    {
        if(i%2)
        {
            // only each second data will be processed by this loop
            continue;
        }

        // prepare the names of the input and output data for this interation of the loop
        InputData  = FrmtStr("My offline Data {0}",i);
        OutputData = FrmtStr("My result {0}",i);

        // do something with the determined input and output data
        Out(OutputData,In(InputData));
    }
}
```

The break condition (second parameter) of the “for()” loop also can be omitted. In this case, the “for()” loop can be left either via a “break” condition or the Analyzing Script suspends at some moment, like in the following case:

```
var main()
{
    var i;
    var BufferMember;
    var Value;

    for(i=0;;i++)
    {
        // we will suspend when i becomes bigger than the highest index of the buffer
        BufferMember = GetBffrM($'H1D 02 H1DForSignal',true,i);
        Value = GetVFrmSD(BufferMember);

        Print(FrmtStr("Index: {0}, Value: {1}",i,Value));
    }
}
```

2.3.3.10.6.3 while()

The “while()” loop can be used in order to run until a certain condition is met.

The following example shows how to use a “while()” loop in combination with a simple counter variable:

```
var main()
{
    var Array[4] = {0,2,4,6};
    var i=0;

    while(i<4)
    {
        Print(FrmtStr("Index: {0}, Value: {1}",i,Array[i]));
        i++;
    }
 }
```

A more typical case for the usage of “while()” are checks of a simple status:

```
var main()
{
    var Value = GetVFrmSD($'My offline Data');

    while(Value <= 100)
    {
        // this will loop until the last value of the input data is above 100
        $'My offline Data' = $'My offline Data' +10;
        Value = GetVFrmSD($'My offline Data');

        Print(FrmtStr("Current Value: {0}",Value));
    }

    Print("We are done");
 }
```

The “break” and “continue” statements are available also for the “while()” loop. They work in the same way as for the “for()” loop:

- “break” jumps out of the current loop
- “continue” jumps back to the beginning of the current loop - depending to whether the break condition is reached or not, the loop is executed again (or not)

2.3.3.10.6.4 do - while()

The “do - while()” loop is very similar to the “while()” loop and also can be used in order to run until a certain condition is met. The main difference is that “while()” checks the break condition BEFORE the first run of the internal loop and “do - while()” checks the break condition AFTER the first run of the internal loop.

The following example shows how to use a “do - while()” loop in combination with a check of a simple status:

```
var main()
{
    var Value = GetVFrmSD($'My offline Data');

    do
    {
        // this will loop until the last value of the input data is above 100
        $'My offline Data' = $'My offline Data' +10;
        Value = GetVFrmSD($'My offline Data');

        Print(FrmtStr("Current Value: {0}",Value));
    }while(Value <= 100)

    Print("We are done");
}
```

The “break” and “continue” statements are available also for the “do - while()” loop. They work in the same way as for the “for()” and “while()” loops:

- “break” jumps out of the current loop
- “continue” jumps back to the beginning of the current loop - note that there is no break condition to check at this point, thus the execution of the inner loop will start for sure after each call of “continue”

2.4 User Function Editors

2.4.1 Overview

User Function Editors are used in order to edit User Functions. User Functions can be used in order to combine multiple (Analyzing and User) functions to a new entity. Multiple User Functions can be maintained and/or executed within Analyzing Models and Analyzing Scripts simultaneously.

User Function Editors are known and accessed exclusively by the ***X-Tools Client***, the ***X-Tools Server*** has no knowledge about **User Function Editors** at all. However, the ***X-Tools Server*** is responsible to maintain User Function Files and therefore all User Function reading and writing operations are performed via the ***X-Tools Server***.

2.4.2 Common Controls

2.4.2.1 Overview

As all **User Function Editors** are built up in a similar way, they share some common controls (e.g. tables and the menu bar) which are the same in all **User Function Editors**.

Each control of a **User Function Editor** has a defined task and provides certain functionalities. The following major controls are provided by the **User Function Editors**. Depending to the **User Function Editor**, one or more parts may not be supported (because they are not needed) and one or more parts may be present in addition to the following ones (because they are necessary):

- User Function Settings Table
- User Function Parameters Table
- User Function Properties Table
- Provided Input Interfaces Table
- Provided Output Interfaces Table
- Menu Bar

2.4.2.2 User Function Settings Table

The User Function Settings table contains all of the function-dependent settings which can be configured within a User Function:

Parameter	Description
Target Name	contains the name of the target to which the User Function is stored
Storage Path	contains the path to which the User Function is stored (absolute or symbolic path)
Creation Date	contains the creation date of the User Function
Modification Date	contains the last modification date of the User Function
Function Description	contains the description of the User Function
Company Name	contains the company name
Author Name	contains the author name

Target Name

All of the information about the Target Name and Storage Path is set up within the **Save As** dialog.

2.4.2.3 User Function Parameters Table

The User Function Parameters table contains all of the function-dependent parameters which can be configured within a User Function:

Parameter	Description
Enable Log Messages	allows to enable or disable log the messages which are output by the (Analyzing and User) functions within the parent User Function
Password	contains the password of the User Function which must be specified in order to open the User Function within a **User Function Editor**
Password (confirm)	contains the password of the User Function (for confirmation) which must be specified in order to open the User Function within a **User Function Editor**

Password

In case a password is specified for a User Function, the password must be provided before the contents of the User Function File are transmitted to the ***X-Tools Client***. Contrary, the ***X-Tools Server*** always can access the User Function File, thus also a password-protected User Function can be used and runs within its parent Analyzing Model or Analyzing Script without any need to provide a password.

2.4.2.4 User Function Properties Table

The User Function Properties table contains all of the properties which can be configured within a User Function:

Parameter	Description
Full Name	allows to enter the full name of the User Function
Short Name	allows to enter the short name of the User Function
Description	allows to enter the description of the User Function
Parent Category	allows to enter the parent category of the User Function
Parent Library	contains the parent library of the User Function
Equidistant Inputs	contains information about whether the User Function requires equidistant inputs
Supported Operating Modes	contains information about the operating modes which are supported by the User Function

Full Name

The Full Name of the User Function must be unique among all User Functions which are present at the ***X-Tools Server***. In addition, the Full Name must not match the Full Name of any Analyzing Function which is present at the ***X-Tools Server***.

Short Name

The Short Name of the User Function must be unique among all User Functions which are present at the ***X-Tools Server***. In addition, the Short Name must not match the Short Name of any Analyzing Function which is present at the ***X-Tools Server***.

Parent Category

The Parent Category is being used in order to group all User Functions of identical Parent Category together below the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch of the **ANS Explorer**.

2.4.2.5 Provided Input Interfaces Table

The Provided Input Interfaces Table table contains all of the input interfaces which have been configured within a User Function:

Parameter	Description
Full Name	allows to enter the full name of the input interface
Short Name	allows to enter the short name of the input interface
Unit	allows to enter the unit of the input interface
Description	allows to enter the description of the input interface
Default Value	allows to enter the default value of the input interface
Parent Function	contains the parent Function of the input interface
Parent Argument	contains the parent argument of the input interface
Sequence ID	contains the sequence ID of the parent Function

Full Name

The Full Name of each input interface must be unique among the input and output interfaces of the User Function.

Short Name

The Short Name of each input interface must be unique among the input and output interfaces of the User Function.

2.4.2.6 Provided Output Interfaces Table

The Provided Output Interfaces Table table contains all of the output interfaces which have been configured within a User Function:

Parameter	Description
Full Name	allows to enter the full name of the output interface
Short Name	allows to enter the short name of the output interface
Unit	allows to enter the unit of the output interface
Description	allows to enter the description of the output interface
Default Value	allows to enter the default value of the output interface
Parent Function	contains the parent Function of the output interface
Parent Argument	contains the parent argument of the output interface
Sequence ID	contains the sequence ID of the parent Function

Full Name

The Full Name of each output interface must be unique among the input and output interfaces of the User Function.

Short Name

The Short Name of each output interface must be unique among the input and output interfaces of the User Function.

2.4.2.7 Menu Bar

Menu Button	Description
New	This button creates a new, empty User Function and initializes the User Function Settings table with the default values for new User Functions. In case there is a User Function opened already, it is closed automatically before the new one is being created.
Open…	This button opens the **Open** dialog where the user is able to select the file which shall be opened.
Save	This button saves the currently opened User Function to the currently known storage location. In case the storage location has not been defined yet, the **Save As** dialog is popping up automatically and the user is able to select the desired storage location.
Save As…	This button opens the **Save As** dialog where the user is able to select the desired storage location.
Convert	This button converts a currently opened User Function into an Analyzing Model and initializes it with the default values within a new **Analyzing Model Editor**.
Close	This button closes the editor. In case the currently opened User Function is not saved, the editor asks the user whether the User Function shall be saved before it is closed.

2.4.3 UFE Standard T001

2.4.3.1 Overview

The **UFE Standard T001** is used in order to visualize, create and edit User Functions of type “Standard T001”. Multiple editors of this type can be opened and used simultaneously.

The following screenshot shows an example of an **UFE Standard T001**:

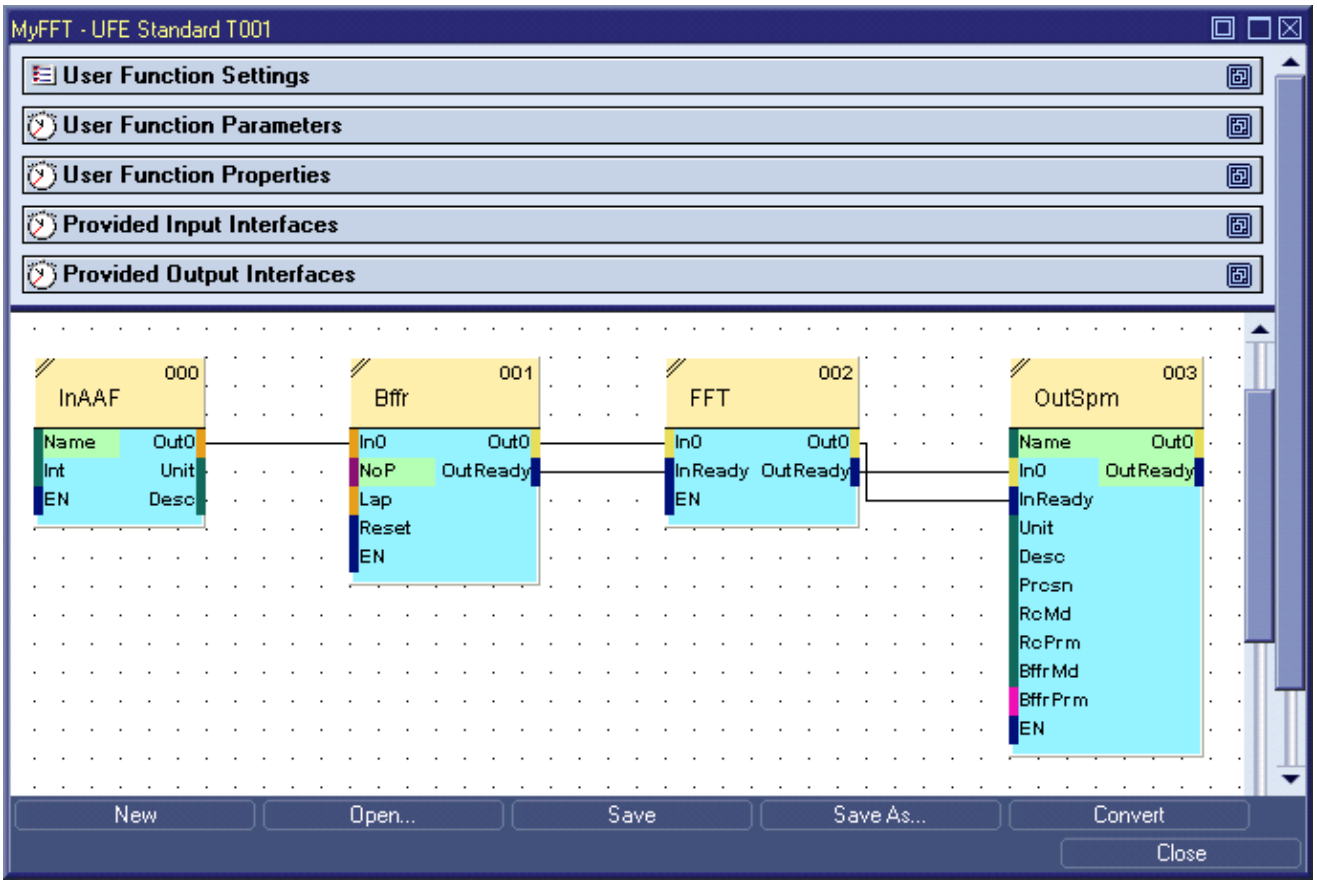


*Figure 5: Example of a **UFE Standard T001***

Each control of the **UFE Standard T001** has a defined task and provides certain functionalities. The following major controls are provided by the **UFE Standard T001**:

- User Function Settings Table
- User Function Parameters Table
- User Function Properties Table
- Provided Input Interfaces Table
- Provided Output Interfaces Table
- Action Area
- Menu Bar
- Dropping of Items
- Analyzing Function Properties Dialog
- User Function Properties Dialog

2.4.3.2 User Function Settings Table

The standard User Function Settings table is being used by the **UFE Standard T001** (see point 2.4.2.2).

2.4.3.3 User Function Parameters Table

The standard User Function Parameters table is being used by the **UFE Standard T001** (see point 2.4.2.3).

2.4.3.4 User Function Properties Table

The standard User Function Properties table is being used by the **UFE Standard T001** (see point 2.4.2.4).

2.4.3.5 Provided Input Interfaces Table

The standard Provided Input Interfaces table is being used by the **UFE Standard T001** (see point 2.4.2.5).

2.4.3.6 Provided Output Interfaces Table

The standard Provided Output Interfaces table is being used by the **UFE Standard T001** (see point 2.4.2.6).

2.4.3.7 Action Area

The Action Area contains all of the (Analyzing and User) functions and links of the current User Function. Via Drag&Drop, the currently present functions and links can be edited and additional functions and links can be added to the User Function.

Symbol of Functions

Each present function is represented by a graphical symbol which provides the following information:

- name of the function
- sequence ID
- names of all input arguments
- names of all output arguments

Keyboard Operations

The following operation can be performed via the keyboard:

Operation	Description
<F2> in case an input argument of a function has the input focus	<F2> allows editing the value of the input argument directly within the Action Area of the **User Function Editor**. Depending to the type of the input argument, an edit box or a combo box is being displayed during the edit.
<Cursor Up> in case an input argument of a function has the input focus	moves the input focus one argument up
<Cursor Down> in case an input argument of a function has the input focus	moves the input focus one argument down

Operations via the Left Mouse Button

The following operations can be performed via the left mouse button:

Operation	Description
single click with releasing the button	A single click of the left mouse button with releasing the button above an item (function or link) selects or deselects the item below the current mouse position: • <Ctrl> can be used in order to select/deselect multiple (Analyzing and User) functions one after the other
single click with keeping the button above a selected item	A single click of the left mouse button above a selected item with keeping the button starts a move or copy operation. The actual moving or copying is performed when the mouse is moved: • [left mouse button down] + [mouse move] moves all currently selected items within the Action Area into the direction of the mouse move • <Esc> cancels the current operation and moves all items back to the positions which they had before the move operation had been started
single click with keeping the button not above a selected item	A single click of the left mouse button not above a selected item with keeping the button starts a select operation. The actual selecting is performed when the left mouse button is released: • [left mouse button down] + [mouse move] highlights all items which are within the area in between the mouse position at the start of the select operation and the current mouse position o during the mouse move and before the left mouse button is being released, a rectangular frame indicates the area of selection o all items which are completely enclosed by the rectangular frame are highlighted in order to show which items will be selected when the left mouse button is being released • <Esc> cancels the current operation and sets the selection of all items back to the state which they had before the selecting operation had been started
double click onto an Analyzing Function	A double click of the left mouse button onto any Analyzing Function opens the **Analyzing Function Properties** dialog for the Analyzing Function below the current mouse position.
double click onto an Analyzing Function argument	A double click of the left mouse button onto any argument of an Analyzing Function opens the **Analyzing Function Properties** dialog for the Analyzing Function below the current mouse position and sets the input focus to the value field of the clicked argument.
double click onto a User Function	A double click of the left mouse button onto any User Function opens the **User Function Properties** dialog for the User Function below the current mouse position.
double click onto a User Function argument	A double click of the left mouse button onto any argument of a User Function opens the **User Function Properties** dialog for the User Function below the current mouse position and sets the input focus to the value field of the clicked argument.

Operations via the Right Mouse Button

The following operation can be performed via the right mouse button:

Operation	Description
single click with releasing the button	A single click of the right mouse button with releasing the button somewhere within the Action Area opens the context menu for the below item.

Drag&Drop of a (Analyzing or User) Function

When a function is dragged into the Action Area, the size and position of the new function is visualized to the user during the Drag&Drop operation. After the drop of the dragged function, the dragged function is inserted at the specified position within the Action Area.

In case multiple functions are being selected simultaneously and dragged into the Action Area, the Action Area processes all selected functions during the Drag&Drop operation and all functions are being inserted into the Action Area simultaneously.

Drag&Drop of Data

The following operations can be performed via Drag&Drop:

Operation	Description
dropping of data into empty space	When data is dropped into empty space of the Action Area, a proper input Analyzing Function, which fits the data type of the dragged data, is searched automatically. In case a proper input Analyzing Function is available, the Drag&Drop operation behaves as if the determined input Analyzing Function would be dropped. After the drop of the dragged data, the determined input Analyzing Function is inserted and the name of the dropped data is added as input for the new Analyzing Function automatically.
dropping of data into an unclassified User Function	When data is dropped into an unclassified User Function (= an User Function whose default data class has not been set to online or offline yet), the default data class (online or offline) of the unclassified User Function is set to the class of the dragged data (online or offline) automatically. In addition, the output time domain (absolute or relative times) of the unclassified User Function is set to the time domain of the dragged data automatically.
dropping of data of different classes	User Functions allow to use online and offline data simultaneously (e.g., a neural cloud (Analyzing or User) function may perform its confidence calculation for online data based on some key values that are available as offline data). Via the standard, textual online/offline identifiers of the system, the user can override the default data class of the User Function when a textual online/offline identifier is being used for the specification of an input data name.

Creation of Links between (Analyzing or User) Functions

In order to create a link from the output argument of one function to the input argument of another function, click onto one of the to-be-connected arguments first and onto the second one afterwards. After the first to-be-connected argument has being clicked, all of the available target arguments of all functions within the **User Functions Editor** are being highlighted automatically.

Sequence IDs

Each (Analyzing or User) function receives a unique sequence ID which determines when it is being called during the processing of one cycle:

- the function with the lowest sequence number is being called first
- the function with the highest sequence number is being called last

The sequence ID of each function can be assigned via the **Analyzing Function Properties** and **User Function Properties** dialogs.

Context menu of a single Analyzing Function

Context Menu Item	Description
Cut	cuts the currently selected Analyzing Function
Copy	copies the currently selected Analyzing Function
Remove	removes the currently selected Analyzing Function
Expand	expands the Analyzing Function (displays the full names of all arguments)
Collapse	collapses the Analyzing Function (displays the short names of all arguments)
Show Values of Inputs > …	sets whether the values of the input arguments of the currently selected Analyzing Functions shall be displayed within the Action Area
Analyzing Function Properties…	opens the **Analyzing Function Properties** dialog

Context menu of a single User Function

Context Menu Item	Description
Cut	cuts the currently selected User Function
Copy	copies the currently selected User Function
Remove	removes the currently selected User Function
Expand	expands the User Function (displays the full names of all arguments)
Collapse	collapses the User Function (displays the short names of all arguments)
Show Values of Inputs > …	sets whether the values of the input arguments of the currently selected User Functions shall be displayed within the Action Area
User Function Properties…	opens the **User Function Properties** dialog
Open…	opens a new **User Funciton Editor** and opens the User Function within this **User Function Editor**

Context Menu of a single Link

Context Menu Item	Description
Remove	removes the currently selected link

Context Menu of multiple selected Items

Context Menu Item	Description
Cut	cuts the currently selected items
Copy	copies the currently selected items
Remove	removes the currently selected items
Expand	expands the (Analyzing and User) functions (displays the full names of the functions and of all of their arguments)
Collapse	collapses the (Analyzing and User) functions (displays the short names of the functions and of all of their arguments)
Show Values of Inputs > …	sets whether the values of the input arguments of the currently selected functions shall be displayed within the Action Area
Analyzing Function Properties…	opens the **Analyzing Function Properties** dialog of the **Analyzing Function** from which the context menu has been called
User Function Properties…	opens the **User Function Properties** dialog of the **User Function** from which the context menu has been called

Context Menu of empty Space

Context Menu Item	Description
Select all Items	selects all items (functions and links)
Paste	pastes the currently copied items
Remove all Items	removes all items (functions and links) from the User Function
Reassign Sequence IDs automatically	automatically assigning all sequence IDs so that the sequence IDs are rising from the input Analyzing Functions to the output Analyzing Functions

2.4.3.8 Log Entries Table

The standard Log Entries table is being used by the **UFE Standard T001** (see point 2.2.2.6).

2.4.3.9 Menu Bar

The standard Menu Bar is being used by the **UFE Standard T001** (see point 2.2.2.7).

2.4.3.10 Dropping of Items

Dropped Item	Description
User Function File Branches of type “AMF Standard T001”	opens the dropped User Function
Analyzing Function Branches in case the **UFE Standard T001** is empty	creates a new User Function, initializes the target server to the parent server of the dragged Analyzing Function and drops the Analyzing Function as seen above
User Function Branches in case the **UFE Standard T001** is empty	creates a new User Function, initializes the target server to the parent server of the dragged User Function and drops the User Function as seen above
Offline Data Branches in case the **UFE Standard T001** is empty	creates a new User Function, initializes the target server to the parent server of the dragged offline data and afterwards it drops the offline data as seen above
Online Data Branches in case the **UFE Standard T001** is empty	creates a new User Function, initializes the target server to the parent server of the dragged online data and afterwards it drops the online data as seen above
Analyzing Function Branches in case the **UFE Standard T001** contains an User Function	drops the Analyzing Function as seen above
User Function Branches in case the **UFE Standard T001** contains an User Function	drops the User Function as seen above
Offline Data Branches in case the **UFE Standard T001** contains an User Function	drops the offline data as seen above
Online Data Branches in case the **UFE Standard T001** contains an User Function	drops the online data as seen above

2.4.3.11 Analyzing Function Properties Dialog

2.4.3.11.1 Overview

The following screenshot shows an example of an **Analyzing Function Properties** dialog:

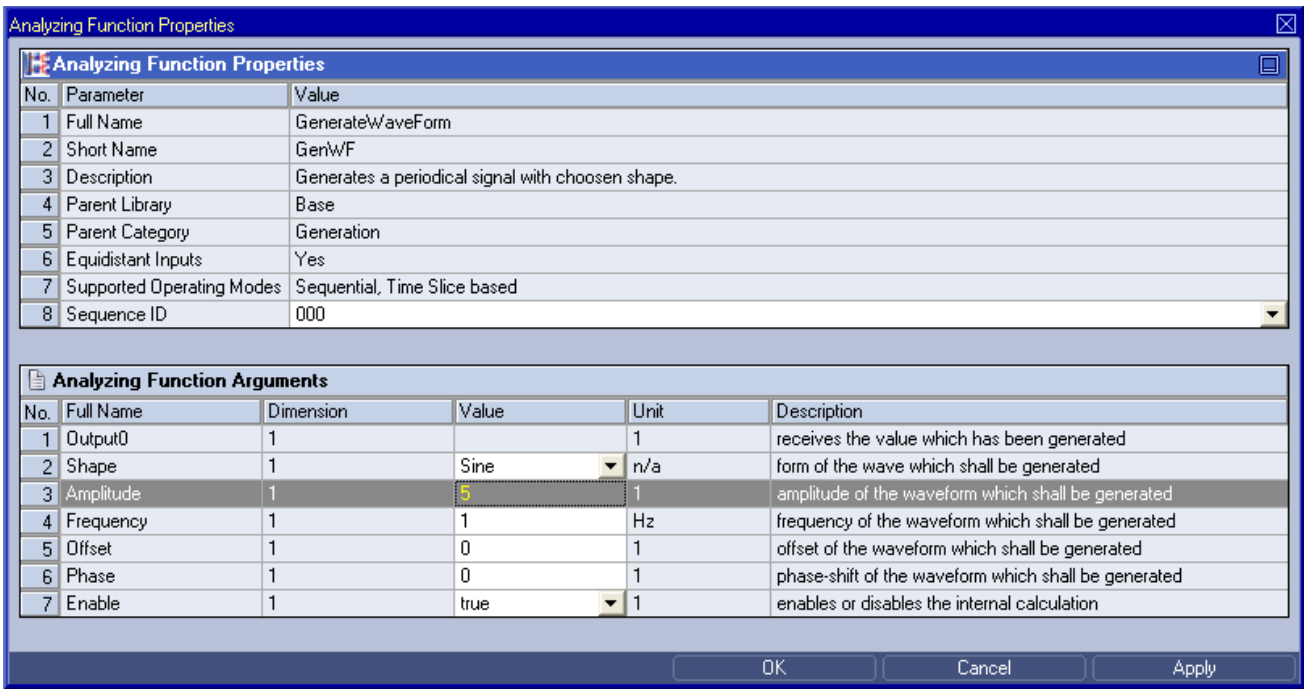


*Figure 6: Example of an **Analyzing Function Properties** Dialog*

2.4.3.11.2 Analyzing Function Properties Table

The Analyzing Function Properties table contains an overview about the properties of the Analyzing Function:

Parameter	Description
Full Name	contains the full name of the Analyzing Function
Short Name	contains the short name of the Analyzing Function
Description	contains the description of the Analyzing Function
Parent Library	contains the parent library of the Analyzing Function
Parent Category	contains the parent category of the Analyzing Function
Equidistant Inputs	indicates whether the Analyzing Function requires equidistant inputs
Supported Operating Modes	contains the supported operating modes of the Analyzing Function
Sequence ID	contains the sequence ID of the Analyzing Function

2.4.3.11.3 Analyzing Function Arguments Table

The Analyzing Function Arguments table contains an overview about the arguments of the Analyzing Function:

Parameter	Description
No.	contains the row number
Full Name	contains the full name of the argument
Short Name	contains the short name of the argument
Dimension	contains “1” when the argument is not configurable or the available number of sub-arguments in case of configurable arguments
Value	contains the value of the argument
Unit	contains the unit of the argument
Description	contains the description of the argument
Class	contains the class of the argument
Direction	contains the direction of the argument
Supported Data Types	contains the supported data types of the argument
Default Value	contains the default value of the argument

It is not possible to add or remove arguments to/from the Analyzing Function Properties table because each Analyzing Function provides detailed information about all of its arguments by itself.

Dimension

In case of configurable arguments, the Dimension cell contains a combo box with the available dimensions.

In case of arguments which are not configurable, the Dimension cell contains the dimension as static text.

Value

The following specific context menu item is provided:

Context Menu Item	Description
Set to Default	sets the value of the cell to its default value

2.4.3.12 User Function Properties Dialog

The appearing and functionality of the **User Function Properties** dialog is identical to the appearing and functionality of the **Analyzing Function Properties** dialog (see point 2.4.3.11).

2.5 Analyzing Functions

2.5.1 General

2.5.1.1 Categories of Analyzing Functions

Each Analyzing Function is assigned to a category in order to group Analyzing Functions of similar purpose together. These categories of Analyzing Functions are being used also by this manual in order to structure the provided Analyzing Functions.

2.5.1.2 Properties of Analyzing Functions

Each Analyzing Function provides the following properties:

Property	Description
Full Name	specifies the full name of the Analyzing Function, e.g. “AbsoluteValue”
Short Name	specifies the short name of the Analyzing Function, e.g. “Abs”
Description	specifies the description of the Analyzing Function, e.g. “Calculates the absolute of the input data or values.”
Parent Library	specifies the parent library of the Analyzing Function, e.g. “Base”
Parent Category	specifies the parent category of the Analyzing Function, e.g. “Arithmetic”
Equidistant Inputs	indicates whether the Analyzing Function can work on base of not-equidistant input data or not; for example, the AbsoluteValue() Analyzing Function can work on not-equidistant input data but the Histogram1D() Analyzing Function can’t
Supported Operating Modes	Indicates whether the Analyzing Function can work in time-slice based calculations and/or in sequential calculations: • time-slice based calculations o all calculation instructions (from the Analyzing Model or Analyzing Script) are executed each defined time slice, like it is known from standard automation systems o time-slice based calculations can be of output data class “online” or ”offline” o the input data for time-slice based calculations can be online and/or offline data, based on absolute and/or relative time stamps o the time-slice based execution is continued endless (in case online data is being output) or when the specified stop time has been reached (in case offline data is being output) • sequential calculations o each defined calculation instruction (from the Analyzing Model or Analyzing Script) is executed in the defined order and processes all input data at once o sequential calculations can be only of output data class “offline”, based on absolute or relative times o the sequential execution is finished when all defined calculation instructions have been executed once

2.5.1.3 Properties of Arguments of Analyzing Functions

Each argument of an Analyzing Function provides the following properties:

Property	Description
Full Name	specifies the full name of the argument, e.g. “Input0”
Short Name	specifies the short name of the argument, e.g. “In0”
Description	specifies the description of the argument, e.g. “data or values whose absolute value shall be calculated”
Direction	indicates whether the argument is an input argument (which provides data to the Analyzing Function) or an output argument (which receives the result of the execution of the Analyzing Function)
Class	Specifies the class of the argument, which can be one of the following: • Data o data arguments carry any kind of values, which can be either time series or single values o data arguments are used in order to transport the values which are to be used for the calculation o data arguments can be used as input and/or as output arguments of Analyzing Functions o each data argument can be assigned/connected to other data arguments (via links or variables) o each data argument can be assigned/connected to control arguments (via links or variables) • Parameter o parameter arguments carry any kind of single values (no time series) o the values are assigned by the user and the algorithm uses the specified values during the calculation o parameter arguments can be used as input arguments of Analyzing Functions o parameter arguments are used for providing of unchangeable values (like the number of points of the FFT(), …) o a value of a parameter argument which has been modified through the user can be taken over only in case its parent Analyzing Model/Analyzing Script is being stopped and restarted • Control o control arguments carry any kind of values, which can be either time series or single values o control arguments are to be used in order to reset, enable or disable the actual execution of the calculation o control arguments can be used as input and/or as output arguments of Analyzing Functions o each control argument can be assigned/connected to other control arguments (via links or variables) o each control argument can be assigned/connected to data arguments (via links or variables)
Default Value	specifies the default value of the argument
Unit	specifies the unit of the argument, if applicable
Configurable	indicates whether the argument is configurable; configurable arguments allow to use up to the specified number of sub-arguments; for example, the Addition() Analyzing Function can be configured to provide 2 to 8 input arguments which are added together during the execution then
Supported Data Types	specifies which data types can be attached to an input argument or which data type are provided to an output argument
Allowed manual Inputs	specifies the values or the value range which can be assigned to the argument manually

2.5.1.4 Examples

The following chapters contain examples for each Analyzing Function. All examples are found within the "..\X-Tools\Server\User\Example Location" directory, which must be added to your system in order to be able to access the examples (see tutorial, chapter "Configuration File Locations").

Some examples use online data from simulated devices as input for the calculation. In case of these examples, the example Interface Profile for the "Simulation T001" interface must be started and in addition also the according Device Profile must be started (see tutorial, chapter "Simulated Devices and Interface Profiles" and chapter "Device Configuration and Device Profiles").

Several examples base on certain offline data as input for the calculation. In case of these examples, the according input data must be loaded into the system before the example can run. All of the referenced offline data are found within the "..\X-Tools\Server\User\Example Offline Files\X-Tools Offline Data Files" directory - add this offline file location (see tutorial, chapter "Offline File Locations") and load the contained offline data via the **ODL Standard T001** (see tutorial, chapter "Loading of Data and Loading Profiles") in order to be able to use all examples which base on offline data.

2.5.2 Arithmetic

2.5.2.1 Abs() - AbsoluteValue()

2.5.2.1.1 General

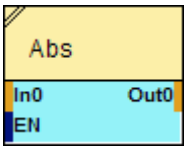
Properties

Property	Description
Full Name	AbsoluteValue
Short Name	Abs
Description	Calculates the absolute of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Abs
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose absolute value shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the absolute value calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
Abs 01	Online	Calculates the absolute value of a generated sine wave.
Abs 02	Offline	Calculates the absolute value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Abs 01	Online	Calculates the absolute value of a generated sine wave.
Abs 02	Offline	Calculates the absolute value of an offline signal.

2.5.2.2 Add() - Addition()

2.5.2.2.1 General

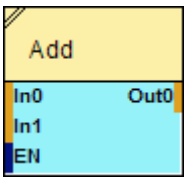
Properties

Property	Description
Full Name	Addition
Short Name	Add
Description	Calculates the arithmetical addition of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Add
(
    DataOrValue In0 ... In7 = 0;
    DataOrValue EN = true
);
```

Symbol



2.5.2.2.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	data or values which shall be added
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the addition
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
Add 01	Online	Calculates the addition of a generated triangle wave and 1.
Add 02	Offline	Calculates the addition of 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
Add 01	Online	Calculates the addition of 5 generated sine waves.
Add 02	Offline	Calculates the addition of 5 offline signals.

2.5.2.3 ArcCos() - ArcCosine()

2.5.2.3.1 General

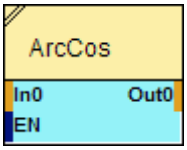
Properties

Property	Description
Full Name	ArcCosine
Short Name	ArcCos
Description	Calculates the arc cosine of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = ArcCos
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.3.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose arc cosine shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	-1 … +1
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the arc cosine calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
ArcCos 01	Online	Calculates the arc cosine of a generated triangle wave and than normalizes it.
ArcCos 02	Offline	Calculates the arc cosine of an offline signal and than normalizes it.

Analyzing Script

Name	Output Data Class	Description
ArcCos 01	Online	Calculates the arc cosine of 0.
ArcCos 02	Offline	Calculates the arc cosine of an offline signal and than normalizes it.

2.5.2.4 ArcCoTan() - ArcCoTangent()

2.5.2.4.1 General

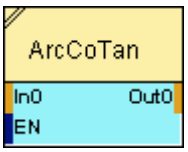
Properties

Property	Description
Full Name	ArcCoTangent
Short Name	ArcCoTan
Description	Calculates the arc cotangent of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = ArcCoTan
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.4.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose arc cotangent shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the arc cotangent calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
ArcCoTan 01	Online	Calculates the arc cotangent of a generated triangle wave and than normalizes it.
ArcCoTan 02	Offline	Calculates the arc cotangent of an offline signal and than normalizes it.

Analyzing Script

Name	Output Data Class	Description
ArcCoTan 01	Online	Calculates the arc cotangent of 0.
ArcCoTan 02	Offline	Calculates the arc cotangent of an offline signal and than normalizes it.

2.5.2.5 ArcSin() - ArcSine()

2.5.2.5.1 General

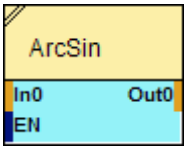
Properties

Property	Description
Full Name	ArcSine
Short Name	ArcSin
Description	Calculates the arc sine of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = ArcSin
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose arc sine shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	-1 … +1
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the arc sine calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
ArcSin 01	Online	Calculates the arc sine of a generated triangle wave and than normalizes it.
ArcSin 02	Offline	Calculates the arc sine of an offline signal and than normalizes it.

Analyzing Script

Name	Output Data Class	Description
ArcSin 01	Online	Calculates the arc sine of 0.8664.
ArcSin 02	Offline	Calculates the arc sine of an offline signal and than normalizes it.

2.5.2.6 ArcTan() - ArcTangent()

2.5.2.6.1 General

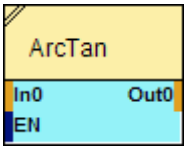
Properties

Property	Description
Full Name	ArcTangent
Short Name	ArcTan
Description	Calculates the arc tangent of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = ArcTan
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.6.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose arc tangent shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the arc tangent calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
ArcTan 01	Online	Calculates the arc tangent of a generated triangle wave.
ArcTan 02	Offline	Calculates the arc tangent of an offline signal.

Analyzing Script

Name	Output Data Class	Description
ArcTan 01	Online	Calculates the arc tangent of 13.14159265.
ArcTan 02	Offline	Calculates the arc tangent of an offline signal.

2.5.2.7 Ceil() - Ceil()

2.5.2.7.1 General

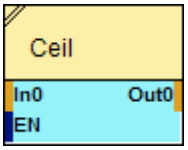
Properties

Property	Description
Full Name	Ceil
Short Name	Ceil
Description	Calculates the smallest integer number that is greater than or equal to the input data or value.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Ceil
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.7.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose ceil shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.7.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the ceil calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.7.4 Examples

Analyzing Model

Name	Output Data Class	Description
Ceil 01	Online	Calculates the ceil value of a generated sine wave.
Ceil 02	Offline	Calculates the ceil value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Ceil 01	Online	Calculates the ceil value of 0.1.
Ceil 02	Offline	Calculates the ceil value of an offline signal.

2.5.2.8 Cos() - Cosine()

2.5.2.8.1 General

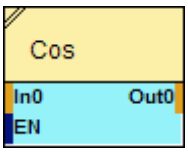
Properties

Property	Description
Full Name	Cosine
Short Name	Cos
Description	Calculates the cosine of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Cos
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.8.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose cosine shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.8.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the cosine calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.8.4 Examples

Analyzing Model

Name	Output Data Class	Description
Cos 01	Online	Calculates the cosine of a generated triangle wave normalized to PI.
Cos 02	Offline	Calculates the cosine of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Cos 01	Online	Calculates the cosine of PI.
Cos 02	Offline	Calculates the cosine of an offline signal.

2.5.2.9 Div() - Division()

2.5.2.9.1 General

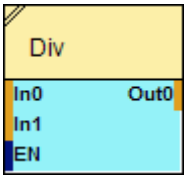
Properties

Property	Description
Full Name	Division
Short Name	Div
Description	Calculates the arithmetical division of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Div
(
    DataOrValue In0 … In7 = 1;
    DataOrValue EN = true
);
```

Symbol



2.5.2.9.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	data or values which shall be used as dividend (In0) and divisors (In1 … In7)
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.9.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the division
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.9.4 Examples

Analyzing Model

Name	Output Data Class	Description
Div 01	Online	Calculates the division of a generated cosine wave by 3.
Div 02	Offline	Calculates the division of 2 offline signal.

Analyzing Script

Name	Output Data Class	Description
Div 01	Online	Calculates the division of 8 defined values.
Div 02	Offline	Calculates the division of 5 offline signal.

2.5.2.10 Exp() - Exponential()

2.5.2.10.1 General

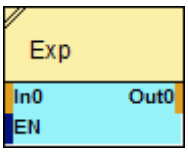
Properties

Property	Description
Full Name	Exponential
Short Name	Exp
Description	Calculates the exponential of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Exp
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.10.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose exponential shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	Input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.10.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the exponential calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.10.4 Examples

Analyzing Model

Name	Output Data Class	Description
Exp 01	Online	Calculates the exponential of 2.302585093.
Exp 02	Offline	Calculates the exponential of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Exp 01	Online	Calculates the exponential of a generated triangle wave.
Exp 02	Offline	Calculates the exponential of an offline signal.

2.5.2.11 Pow() - Exponentiation()

2.5.2.11.1 General

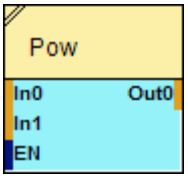
Properties

Property	Description
Full Name	Exponentiation
Short Name	Pow
Description	Calculates the exponentiation of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Pow
(
    DataOrValue In0 = 0,
    DataOrValue In1 = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.2.11.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values used as base in the exponentiation calculation
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In1

Property	Description
Full Name	Input1
Short Name	In1
Description	data or values used as exponent in the exponentiation calculation
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.11.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the exponentiation calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.11.4 Examples

Analyzing Model

Name	Output Data Class	Description
Pow 01	Online	Calculates the exponentiation of a generated triangle wave to the power of 1.234.
Pow 02	Offline	Calculates the exponentiation of one offline signal to the power of another offline signal.

Analyzing Script

Name	Output Data Class	Description
Pow 01	Online	Calculates the exponentiation of 1.356 to the power of 2.4545.
Pow 02	Offline	Calculates the exponentiation of an offline signal to the power of 1.73.

2.5.2.12 Floor() - Floor()

2.5.2.12.1 General

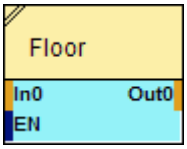
Properties

Property	Description
Full Name	Floor
Short Name	Floor
Description	Calculates the largest integer number that is less than or equal to the input data or value.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Floor
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.12.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose floor shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.12.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the floor calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.12.4 Examples

Analyzing Model

Name	Output Data Class	Description
Floor 01	Online	Calculates the floor value of a generated sine wave.
Floor 02	Offline	Calculates the floor value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Floor 01	Online	Calculates the floor value of 5.7.
Floor 02	Offline	Calculates the floor value of an offline signal.

2.5.2.13 Integral() - Integral()

2.5.2.13.1 General

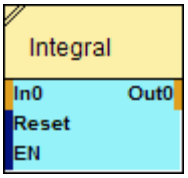
Properties

Property	Description
Full Name	Integral
Short Name	Integral
Description	Calculates the integral of the input data or values.
Parent Library	Basic
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Integral
(
    DataOrValue In0 = 0,
    DataOrValue Reset = false,
    DataOrValue EN = true
);
```

Symbol



2.5.2.13.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose integral shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the integration
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.13.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the integration
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.13.4 Examples

Analyzing Model

Name	Output Data Class	Description
Integral 01	Online	Calculates the integral of a generated sine wave.
Integral 02	Offline	Calculates the integral of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Integral 01	Online	Calculates the integral of a generated sine wave.
Integral 02	Offline	Calculates the integral of an offline signal.

2.5.2.14 Log() - Logarithm()

2.5.2.14.1 General

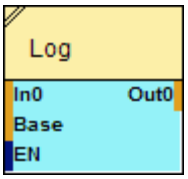
Properties

Property	Description
Full Name	Logarithm
Short Name	Log
Description	Calculates the logarithm to a defined base of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Log
(
    DataOrValue In0 = 1,
    DataOrValue Base = 10,
    DataOrValue EN = true
);
```

Symbol



2.5.2.14.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose logarithm shall be calculated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

Base

Property	Description
Full Name	Base
Short Name	Base
Description	data or values which is used as the base in the logarithm calculation
Direction	input
Class	data
Default Value	10
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … 1) & (1 … max of the Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.14.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the logarithm calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.14.4 Examples

Analyzing Model

Name	Output Data Class	Description
Log 01	Online	Calculates the logarithm of a generated triangle wave to the base 1.73.
Log 02	Offline	Calculates the logarithm of an offline signal to the base 2.3 and to the base 50.

Analyzing Script

Name	Output Data Class	Description
Log 01	Online	Calculates the logarithm of 1.356 to the base 2.4545.
Log 02	Offline	Calculates the logarithm of an offline signal to the base 2.23.

2.5.2.15 Log10() - Logarithm10()

2.5.2.15.1 General

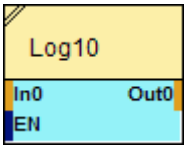
Properties

Property	Description
Full Name	Logarithm10
Short Name	Log10
Description	Calculates the base 10 logarithm of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Log10
(
    DataOrValue In0 = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.2.15.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose logarithm to the base 10 shall be calculated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.15.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the base 10 logarithm calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.15.4 Examples

Analyzing Model

Name	Output Data Class	Description
Log10 01	Online	Calculates the base 10 logarithm of a generated triangle wave.
Log10 02	Offline	Calculates the base 10 logarithm of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Log10 01	Online	Calculates the base 10 logarithm of 122.5454.
Log10 02	Offline	Calculates the base 10 logarithm of an offline signal.

2.5.2.16 Log2() - Logarithm2()

2.5.2.16.1 General

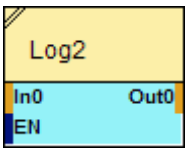
Properties

Property	Description
Full Name	Logarithm2
Short Name	Log2
Description	Calculates the base 2 logarithm of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Log2
(
    DataOrValue In0 = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.2.16.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose logarithm to the base 2 shall be calculated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.16.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the base 2 logarithm calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.16.4 Examples

Analyzing Model

Name	Output Data Class	Description
Log2 01	Online	Calculates the base 2 logarithm of a generated triangle wave.
Log2 02	Offline	Calculates the base 2 logarithm of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Log2 01	Online	Calculates the base 2 logarithm of 4.
Log2 02	Offline	Calculates the base 2 logarithm of an offline signal.

2.5.2.17 Mod() - Modulo()

2.5.2.17.1 General

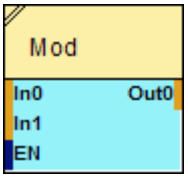
Properties

Property	Description
Full Name	Modulo
Short Name	Mod
Description	Calculates the remainder of the division of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Mod
(
    DataOrValue In0 = 0,
    DataOrValue In1 = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.2.17.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values used as the dividend in the modulo calculation
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In1

Property	Description
Full Name	Input1
Short Name	In1
Description	data or values used as the divisor in the modulo calculation
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min of Decimal64 … 0) & (0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.17.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the modulo calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.17.4 Examples

Analyzing Model

Name	Output Data Class	Description
Mod 01	Online	Calculates the modulo of a generated triangle wave with divisor 3.
Mod 02	Offline	Calculates the modulo of an offline signal with divisor 2.

Analyzing Script

Name	Output Data Class	Description
Mod 01	Online	Calculates the modulo of 7 with divisor 4.
Mod 02	Offline	Calculates the modulo of an offline signal with divisor 7.

2.5.2.18 Mul() - Multiplication()

2.5.2.18.1 General

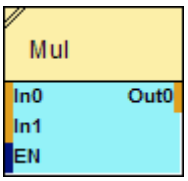
Properties

Property	Description
Full Name	Multiplication
Short Name	Mul
Description	Calculates the arithmetical multiplication of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Mul
(
    DataOrValue In0 … In7 = 0;
    DataOrValue EN = true
);
```

Symbol



2.5.2.18.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	data or values which shall be multiplied
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.18.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the multiplication
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.18.4 Examples

Analyzing Model

Name	Output Data Class	Description
Mul 01	Online	Calculates the multiplication of a generated cosine wave by -2.5.
Mul 02	Offline	Calculates the multiplication of 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
Mul 01	Online	Calculates the multiplication of 3 defined values.
Mul 02	Offline	Calculates the multiplication of 5 offline signals.

2.5.2.19 Ln() - NaturalLogarithm()

2.5.2.19.1 General

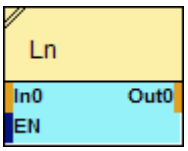
Properties

Property	Description
Full Name	NaturalLogarithm
Short Name	Ln
Description	Calculates the natural logarithm of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Ln
(
    DataOrValue In0 = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.2.19.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose natural logarithm shall be calculated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.19.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the natural logarithm calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.19.4 Examples

Analyzing Model

Name	Output Data Class	Description
Ln 01	Online	Calculates the natural logarithm of a generated triangle wave.
Ln 02	Offline	Calculates the natural logarithm of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Ln 01	Online	Calculates the natural logarithm of e.
Ln 02	Offline	Calculates the natural logarithm of an offline signal.

2.5.2.20 Neg() - Negation()

2.5.2.20.1 General

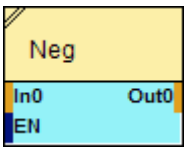
Properties

Property	Description
Full Name	Negation
Short Name	Neg
Description	Calculates the negative value of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Neg
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.20.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose negation shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.20.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the negation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.20.4 Examples

Analyzing Model

Name	Output Data Class	Description
Neg 01	Online	Calculates the negative value of a generated cosine wave.
Neg 02	Offline	Calculates the negative value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Neg 01	Online	Calculates the negative value of 5.7.
Neg 02	Offline	Calculates the negative value of an offline signal.

2.5.2.21 NthRt() - NthRoot()

2.5.2.21.1 General

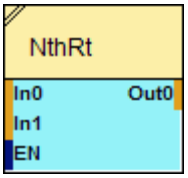
Properties

Property	Description
Full Name	NthRoot
Short Name	NthRt
Description	Calculates the n-th root of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = NthRt
(
    DataOrValue In0 = 0,
    DataOrValue In1 = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.2.21.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values used as radicand for the calculation of the n-th root
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

In1

Property	Description
Full Name	Input1
Short Name	In1
Description	data or values used as radical for the calculation of the n-th root
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	1 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.21.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the n-th root calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.21.4 Examples

Analyzing Model

Name	Output Data Class	Description
NthRt 01	Online	Calculates the 2.2-root of a generated triangle wave.
NthRt 02	Offline	Calculates the 3.333-root of an offline signal.

Analyzing Script

Name	Output Data Class	Description
NthRt 01	Online	Calculates the 12.45 -root of 671.356.
NthRt 02	Offline	Calculates the 5.27-root of an offline signal.

2.5.2.22 Round() - Round()

2.5.2.22.1 General

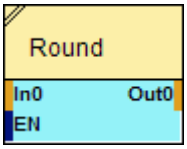
Properties

Property	Description
Full Name	Round
Short Name	Round
Description	Rounds the input data or values to the closest integer.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Round
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.22.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values to be rounded by the calculation
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.22.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the round calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.22.4 Examples

Analyzing Model

Name	Output Data Class	Description
Round 01	Online	Rounds the generated sine wave to a close integer.
Round 02	Offline	Rounds the offline signal to a close integer.

Analyzing Script

Name	Output Data Class	Description
Round 01	Online	Rounds 0.5 to a close integer.
Round 02	Offline	Rounds the offline signal to a close integer.

2.5.2.23 Sin() - Sine()

2.5.2.23.1 General

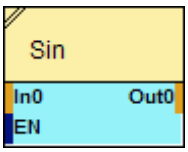
Properties

Property	Description
Full Name	Sine
Short Name	Sin
Description	Calculates the sine of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Sin
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.23.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose sine shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.23.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the sine calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.23.4 Examples

Analyzing Model

Name	Output Data Class	Description
Sin 01	Online	Calculates the sine of a generated triangle wave normalized to PI.
Sin 02	Offline	Calculates the sine of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Sin 01	Online	Calculates the sine of PI.
Sin 02	Offline	Calculates the sine of an offline signal.

2.5.2.24 SqRt() - SquareRoot()

2.5.2.24.1 General

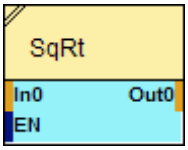
Properties

Property	Description
Full Name	SquareRoot
Short Name	SqRt
Description	Calculates the square root of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = SqRt
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.24.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values used as radicand for the calculation of the square root
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.24.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the square root calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.24.4 Examples

Analyzing Model

Name	Output Data Class	Description
SqRt 01	Online	Calculates the square root of a generated cosine wave.
SqRt 02	Offline	Calculates the square root of an offline signal.

Analyzing Script

Name	Output Data Class	Description
SqRt 01	Online	Calculates the square root of 2.25.
SqRt 02	Offline	Calculates the square root of an offline signal.

2.5.2.25 Sub() - Subtraction()

2.5.2.25.1 General

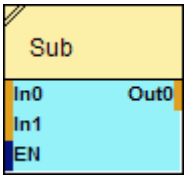
Properties

Property	Description
Full Name	Subtraction
Short Name	Sub
Description	Calculates the arithmetical subtraction of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Sub
(
    DataOrValue In0 … In7 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.25.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	data or values which shall be used as minuend (In0) and subtrahend (In1 … In7)
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.25.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the subtraction
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.25.4 Examples

Analyzing Model

Name	Output Data Class	Description
Sub 01	Online	Calculates the subtraction of 2 from a generated cosine wave.
Sub 02	Offline	Calculates the subtraction of 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
Sub 01	Online	Calculates the subtraction of 3 defined values.
Sub 02	Offline	Calculates the subtraction of 5 offline signals.

2.5.2.26 Tan() - Tangent()

2.5.2.26.1 General

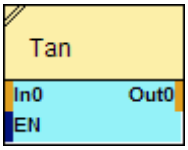
Properties

Property	Description
Full Name	Tangent
Short Name	Tan
Description	Calculates the tangent of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Tan
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.26.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose tangent shall be calculated
Direction	Input
Class	Data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64 / π/2+ n*π, n = 0,1,2,…
Remark	None

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	Input
Class	Control
Default Value	True
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	None

2.5.2.26.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the tangent calculation
Direction	Output
Class	Data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	None

2.5.2.26.4 Examples

Analyzing Model

Name	Output Data Class	Description
Tan 01	Online	Calculates the tangent of a generated triangle wave normalized to PI.
Tan 02	Offline	Calculates the tangent of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Tan 01	Online	Calculates the tangent of PI.
Tan 02	Offline	Calculates the tangent of an offline signal.

2.5.2.27 Trunc() - Truncate()

2.5.2.27.1 General

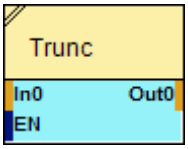
Properties

Property	Description
Full Name	Truncate
Short Name	Trunc
Description	Calculates the largest integer number that is less than or equal to the absolute of the input data or values.
Parent Library	Base
Parent Category	Arithmetic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Trunc
(
    DataOrValue In0 = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.2.27.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values to be truncated by the calculation
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.2.27.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the truncation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.2.27.4 Examples

Analyzing Model

Name	Output Data Class	Description
Trunc 01	Online	Calculates the trunk value of a generated sine wave.
Trunc 02	Offline	Calculates the trunk value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Trunc 01	Online	Calculates the truncated value of -5.8.
Trunc 02	Offline	Calculates the truncated value of an offline signal.

2.5.3 Classification

2.5.3.1 RFC() – RainFlowCounting()

2.5.3.1.1 General

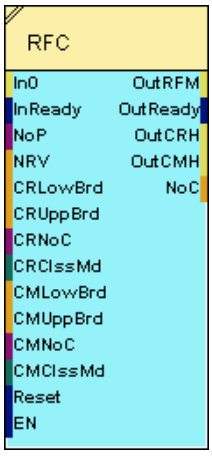
Properties

Property	Description
Full Name	RainFlowCounting
Short Name	RFC
Description	Applies the rainflow counting algorithm to the input buffer.
Parent Library	Advanced
Parent Category	Classification
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer2D OutRFM = RFC
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    Value        NoP = 1024,
    Value        NRV = 0,
    Value        CRLowBrd = 0,
    Value        CRUppBrd = 1,
    Value        CRNoC = 2,
    Value        CRClssMd = "Outside",
    Value        CMLowBrd = 0,
    Value        CMUppBrd = 1,
    Value        CMNoC = 2,
    Value        CMClssMd = "Outside",
    DataOrValue  Reset = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL,
    Buffer1D*    OutCRH = NULL,
    Buffer1D*    OutCMH = NULL,
    DataOrValue* NoC = NULL
);
```

Symbol



2.5.3.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the rainflow counting classification
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines the number of points which shall be used for the rainflow counting
Direction	input
Class	parameter
Default Value	1024
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	the NoP must be set equal to the number of points of the input buffer

NRV

Property	Description
Full Name	NoiseReductionValue
Short Name	NRV
Description	threshold value of the noise reduction filter which shall be used by the extremes finding
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

CRLowBrd

Property	Description
Full Name	CyclesRangeLowerBorder
Short Name	CRLowBrd
Description	determines the position of the lower borderline for the cycles range
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CRUppBrd

Property	Description
Full Name	CyclesRangeUpperBorder
Short Name	CRUppBrd
Description	determines the position of the upper borderline for the cycles range
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CRNoC

Property	Description
Full Name	CyclesRangeNumberOfClasses
Short Name	CRNoC
Description	determines the number of classes for the cycles range
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

CRClssMd

Property	Description
Full Name	CyclesRangeClassificationMode
Short Name	CRClssMd
Description	contains the acquisition mode of the cycles range for the rainflow counting
Direction	input
Class	parameter
Default Value	“Outside”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Inside”, “Outside”
Remark	none

CMLowBrd

Property	Description
Full Name	CyclesMeanLowerBorder
Short Name	CMLowBrd
Description	determines the position of the lower borderline for the cycles mean
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CMUppBrd

Property	Description
Full Name	CyclesMeanUpperBorder
Short Name	CMUppBrd
Description	determines the position of the upper borderline for the cycles mean
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CMNoC

Property	Description
Full Name	CyclesMeanNumberOfClasses
Short Name	CMNoC
Description	determines the number of classes for the cycles mean
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

CMClssMd

Property	Description
Full Name	CyclesMeanClassificationMode
Short Name	CMClssMd
Description	contains the acquisition mode of the cycles mean for the rainflow counting
Direction	input
Class	parameter
Default Value	“Outside”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Inside”, “Outside”
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the rainflow counting
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.3.1.3 Outputs

OutRFM

Property	Description
Full Name	OutputRainFlowMatrix
Short Name	OutRFM
Description	receives the 2-dimensional buffer data with the result of the rainflow counting
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer2D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the output data are valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

OutCRH

Property	Description
Full Name	OutputCyclesRangeHistogram
Short Name	OutCRH
Description	receives the 1-dimensional histogram of the cycles range values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutCMH

Property	Description
Full Name	OutputCyclesMeanHistogram
Short Name	OutCMH
Description	receives the 1-dimensional histogram of the cycles mean values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutNoC

Property	Description
Full Name	OutputNumberOfCycles
Short Name	OutNoC
Description	receives the detected number of rainflow cycles
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.3.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
RFC 01	Online	Calculates rainflow matrix for a generated online signal.
RFC 02	Offline	Calculates rainflow matrix for an offline signal.

Analyzing Script

Name	Output Data Class	Description
RFC 01	Online	Calculates rainflow matrix for a generated online signal.
RFC 02	Offline	Calculates rainflow matrix for an offline signal.

2.5.3.2 TLC() – TimeAtLevelCounting()

2.5.3.2.1 General

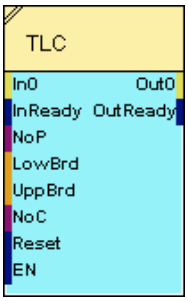
Properties

Property	Description
Full Name	TimeAtLevelCounting
Short Name	TLC
Description	Applies the time at level counting algorithm to the input buffer.
Parent Library	Base
Parent Category	Classification
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = TLC
(
    Buffer1D      In0 = 0,
    DataOrValue   InReady = false,
    Value         NoP = 1024,
    Value         LowBrd = 0,
    Value         UppBrd = 1,
    Value         NoC = 2,
    DataOrValue   Reset = false,
    DataOrValue   EN = true,
    DataOrValue*  OutReady = NULL
);
```

Symbol



2.5.3.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the time at level counting classification
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines the number of points which shall be used for the time at level counting
Direction	input
Class	parameter
Default Value	1024
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	the NoP must be set equal to the number of points of the input buffer

LowBrd

Property	Description
Full Name	LowerBorder
Short Name	LowBrd
Description	determines the position of the lower borderline for the classification
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

UppBrd

Property	Description
Full Name	UpperBorder
Short Name	UppBrd
Description	determines the position of the upper borderline for the classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoC

Property	Description
Full Name	NumberOfClasses
Short Name	NoC
Description	determines the number of classes for the classification
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the time at level counting
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.3.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer data with the result of the time at level counting
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.3.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
TLC 01	Online	Calculates time at level histogram for a generated online signal.
TLC 02	Offline	Calculates time at level histogram for an offline signal.

Analyzing Script

Name	Output Data Class	Description
TLC 01	Online	Calculates time at level histogram for a generated online signal.
TLC 02	Offline	Calculates time at level histogram for an offline signal.

2.5.4 Comparison

2.5.4.1 Cmp() - Compare()

2.5.4.1.1 General

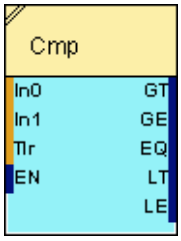
Properties

Property	Description
Full Name	Compare
Short Name	Cmp
Description	Compares two input values and returns the comparison result.
Parent Library	Base
Parent Category	Comparison
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue GT = Cmp
(
    DataOrValue  In0 = 0,
    DataOrValue  In1 = 0,
    Value        Tlr = 0,
    DataOrValue  EN = true,
    DataOrValue* GE = NULL,
    DataOrValue* EQ = NULL,
    DataOrValue* LT = NULL,
    DataOrValue* LE = NULL
);
```

Symbol



2.5.4.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	first input value to compare
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In1

Property	Description
Full Name	Input1
Short Name	In1
Description	second input value to compare
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Tlr

Property	Description
Full Name	Tolerance
Short Name	Tlr
Description	indicates the tolerance interval by the comparison
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.4.1.3 Outputs

GT

Property	Description
Full Name	GreaterThan
Short Name	GT
Description	indicates whether the actual value of Input0 is bigger than the correlating value of Input1
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	GT = true in case the value of (Input0 - Tolerance) is greater than the value of Input1, otherwise is false

GE

Property	Description
Full Name	GreaterEqual
Short Name	GE
Description	indicates whether the actual value of Input0 is bigger than or equal to the correlating value of Input1
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	GE = true in case the value of (Input0 - Tolerance) is greater than the value of Input1 or the absolute value of (Input0 – Input1) is lower than the value of Tolerance, otherwise is false

EQ

Property	Description
Full Name	Equal
Short Name	EQ
Description	indicates whether the actual value of Input0 is equal to the correlating value of Input1
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	EQ = true in case the absolute value of (Input0 – Input1) is lower than the value of Tolerance , otherwise is false

LT

Property	Description
Full Name	LowerThan
Short Name	LT
Description	indicates whether the actual value of Input0 is lower than the correlating value of Input1
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	LT = true in case the value of (Input0 + Tolerance) is lower than the value of Input1, otherwise is false

LE

Property	Description
Full Name	LowerEqual
Short Name	LE
Description	indicates whether the actual value of Input0 is lower than or equal to the correlating value of Input1
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	LE = true in case the value of (Input0 + Tolerance) is lower than the value of Input1 or the absolute value of (Input0 – Input1) is lower than the value of Tolerance, otherwise is false

2.5.4.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
Cmp 01	Online	Compares a generated sine wave with a generated triangle wave.
Cmp 02	Offline	Compares 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
Cmp 01	Online	Compares 2 defined values.
Cmp 02	Offline	Compares 2 offline signals.

2.5.5 Control

2.5.5.1 Continue() - Continue()

2.5.5.1.1 General

Properties

Property	Description
Full Name	Continue
Short Name	Continue
Description	Continues the execution of the specified Configuration File.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = Continue
(
    Value       FileName = "",
    Value       WaitMode = "Wait",
    Value       Timeout = 0,
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.1.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the Configuration File which shall be continued, in format "Location Name://[Directory/]Configuration File Name.Extension", e.g. "Example Location://Arithmetic/Abs 01.amf.xts"
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File ("*.ipf.xts"), Device Profile File ("*.dpf.xts"), Analyzing Model ("*.amf.xts"), Analyzing Script ("*.asf.xts") or Storage Profile File ("*.spf.xts")

WaitMode

Property	Description
Full Name	WaitMode
Short Name	WaitMode
Description	specifies whether the function shall wait until the stop of the Configuration File
Direction	input
Class	parameter
Default Value	“Wait”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Wait”, “DoNotWait”
Remark	in case “Wait” is being chosen, the function waits until the specified configuration enters the state “stopped” again; in case “DoNotWait” is being chosen, the function waits until the specified Configuration File enters the state “started”

Timeout

Property	Description
Full Name	Timeout
Short Name	Timeout
Description	specifies the maximum waiting time in case the Configuration File does not reach the expected state (“stopped” in case of the WaitMode “Wait” or “started” in case of the WaitMode “DoNotWait”)
Direction	input
Class	parameter
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	a value of 0 disables the timeout and Continue() waits for the expected state forever

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.1.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the function has exited because the expected state has been reached and Status = false in case the function has exited because of any other reason (e.g. because of the timeout)

2.5.5.1.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **Continue()**.

Analyzing Script

Name	Output Data Class	Description
Pause 01	Offline	Alternating pause and continue of two Configuration Files.

2.5.5.2 Exit() - Exit()

2.5.5.2.1 General

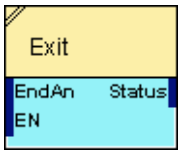
Properties

Property	Description
Full Name	Exit
Short Name	Exit
Description	Stops current analyzing program.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Status = Exit
(
    DataOrValue EndAn = true,
    DataOrValue EN = true
);
```

Symbol



2.5.5.2.2 Inputs

EndAn

Property	Description
Full Name	EndAnalysis
Short Name	EndAn
Description	controls condition for the exiting
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	in case EndAnalysis = true the current analyzing program will be stopped, otherwise it does nothing

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.2.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = false in case the function does not cease time-slice and Status = true if function causes time-sliced to cease

2.5.5.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
Exit 01	Online	Stops execution of the current analyzing model after reaching of proper condition by a generated signal.
Exit 02	Offline	Stops execution of the current analyzing model after reaching of proper condition by an offline signal.

Analyzing Script

Name	Output Data Class	Description
Exit 01	Online	Stops execution of the current analyzing script after reaching of proper condition by a generated signal.
Exit 02	Offline	Stops execution of the current analyzing script after reaching of proper condition by an offline signal.

2.5.5.3 IsDataPrsnt() - IsDataPresent()

2.5.5.3.1 General

Properties

Property	Description
Full Name	IsDataPresent
Short Name	IsDataPrsnt
Description	Detects whether the specified data is currently present.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = IsDataPrsnt
(
    Value       Name = "",
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.3.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the online or offline data whose presence shall be checked, e.g. "~'OnlineSignal_1'" for online data or "$'OfflineData_1'" for offline data
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the data
Remark	in case a data name prefix is being specified for output data of the parent Analyzing Model/Analyzing Script, the specified output data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.3.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the specified data is present at the moment even if it does contain not any data values, otherwise Status = false

2.5.5.3.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **IsDataPrsnt()**.

Analyzing Script

Name	Output Data Class	Description
IsDataPrsnt 01	Offline	Detects whether the offline data produced by the specified analyzing program is currently present.

2.5.5.4 Pause() - Pause()

2.5.5.4.1 General

Properties

Property	Description
Full Name	Pause
Short Name	Pause
Description	Pauses the execution of the specified Configuration File.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = Pause
(
    Value       FileName = "",
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.4.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the Configuration File which shall be paused, in format “Location Name://[Directory/]Configuration File Name.Extension”, e.g. “Example Location://Arithmetic/Abs 01.amf.xts”
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File (“*.ipf.xts”), Device Profile File (“*.dpf.xts”), Analyzing Model (“*.amf.xts”), Analyzing Script (“*.asf.xts”) or Storage Profile File (“*.spf.xts”)

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.4.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the function has exited because the state “paused” has been reached and Status = false in case the function has exited because of any other reason

2.5.5.4.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **Pause()**.

Analyzing Script

Name	Output Data Class	Description
Pause 01	Offline	Alternating pause and continue of two Configuration Files.

2.5.5.5 RemoveData() - RemoveAllDataFrom()

2.5.5.5.1 General

Properties

Property	Description
Full Name	RemoveAllDataFrom
Short Name	RemoveData
Description	Removes all data created by specified analyzing program.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = RemoveData
(
    Value       FileName = "",
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.5.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the analyzing program whose created data shall be removed, in format “Location Name://[Directory/]Analyzing Program Name.Extension”, e.g. “Example Location://Arithmetic/Abs 01.amf.xts”
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File (“*.ipf.xts”), Device Profile File (“*.dpf.xts”), Analyzing Model (“*.amf.xts”), Analyzing Script (“*.asf.xts”) or Storage Profile File (“*.spf.xts”)

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.5.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the specified offline data has been removed successfully, otherwise Status = false

2.5.5.5.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **RemoveData()**.

Analyzing Script

Name	Output Data Class	Description
RemoveData 01	Offline	Removes all offline data created by the specified analyzing program.

2.5.5.6 Start() - Start()

2.5.5.6.1 General

Properties

Property	Description
Full Name	Start
Short Name	Start
Description	Starts the execution of the specified Configuration File.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = Start
(
    Value       FileName = "",
    Value       WaitMode = "Wait",
    Value       Timeout = 0,
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.6.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the Configuration File which shall be started, in format "Location Name://[Directory/]Configuration File Name.Extension", e.g. "Example Location://Arithmetic/Abs 01.amf.xts"
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File ("*.ipf.xts"), Device Profile File ("*.dpf.xts"), Analyzing Model ("*.amf.xts"), Analyzing Script ("*.asf.xts") or Storage Profile File ("*.spf.xts")

WaitMode

Property	Description
Full Name	WaitMode
Short Name	WaitMode
Description	specifies whether the function shall wait until the stop of the Configuration File
Direction	input
Class	parameter
Default Value	“Wait”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Wait”, “DoNotWait”
Remark	in case “Wait” is being chosen, the function waits until the specified configuration enters the state “stopped” again; in case “DoNotWait” is being chosen, the function waits until the specified Configuration File enters the state “started”

Timeout

Property	Description
Full Name	Timeout
Short Name	Timeout
Description	specifies the maximum waiting time in case the Configuration File does not reach the expected state (“stopped” in case of the WaitMode “Wait” or “started” in case of the WaitMode “DoNotWait”)
Direction	input
Class	parameter
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	a value of 0 disables the timeout and Start() waits for the expected state forever

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.6.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the function has exited because the expected state has been reached and Status = false in case the function has exited because of any other reason (e.g. because of the timeout)

2.5.5.6.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **Start()**.

Analyzing Script

Name	Output Data Class	Description
Start 01	Offline	Starts the specified Configuration File after reaching of defined conditions.

2.5.5.7 Stop() - Stop()

2.5.5.7.1 General

Properties

Property	Description
Full Name	Stop
Short Name	Stop
Description	Stops the execution of the specified Configuration File.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = Stop
(
    Value       FileName = "",
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.7.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the Configuration File which shall be stopped, in format “Location Name://[Directory/]Configuration File Name.Extension”, e.g. “Example Location://Arithmetic/Abs 01.amf.xts”
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File (“*.ipf.xts”), Device Profile File (“*.dpf.xts”), Analyzing Model (“*.amf.xts”), Analyzing Script (“*.asf.xts”) or Storage Profile File (“*.spf.xts”)

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.7.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the function has exited because the state “stopped” has been reached and Status = false in case the function has exited because of any other reason

2.5.5.7.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **Stop()**.

Analyzing Script

Name	Output Data Class	Description
Stop 01	Offline	Stops the specified Configuration File after reaching of defined conditions.

2.5.5.8 WaitForCF() – WaitForConfigurationFile()

2.5.5.8.1 General

Properties

Property	Description
Full Name	WaitForConfigurationFile
Short Name	WaitForCF
Description	Blocks the execution of the current calculation until the specified state of the specified Configuration File is being reached.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	N/A
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = WaitForCF
(
    Value         FileName = "",
    Value         State0 ... State4 = "Stopped";
    Value         Timeout = 0,
    DataOrValue   EN = true,
    DataOrValue* Out0 = ""
);
```

Symbol

N/A

2.5.5.8.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the Configuration File for which the function shall wait, in format "Location Name://[Directory/]Configuration File Name.Extension", e.g. "Example Location://Arithmetic/Abs 01.amf.xts"
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File ("*.ipf.xts"), Device Profile File ("*.dpf.xts"), Analyzing Model ("*.amf.xts"), Analyzing Script ("*.asf.xts") or Storage Profile File ("*.spf.xts")

State0 … State4

Property	Description
Full Name	State0 … State4
Short Name	State0 … State4
Description	specifies the states of the Configuration File for which the function shall wait
Direction	input
Class	parameter
Default Value	“Stopped”
Unit	n/a
Configurable	5
Supported Data Types	String
Allowed manual Inputs	“Started”, “Stopped”, “Suspended”, “Paused”, “Triggering”
Remark	in case the Configuration File is in a specified state already at the first call to WaitForCF(), WaitForCF() returns immediately

In0_4

Property	Description
Full Name	Input0_4
Short Name	In0_4
Description	determines how many of the input arguments State0 … State4 are used by the calculation
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 5
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

Timeout

Property	Description
Full Name	Timeout
Short Name	Timeout
Description	specifies the maximum waiting time in case the Configuration File does not reach any of the specified states
Direction	input
Class	parameter
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	a value of 0 disables the timeout and WaitForCF() waits for the specified state forever

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.8.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the function has exited because one of the specified states has been reached and Status = false in case the function has exited because of any other reason (e.g. because of the timeout)

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the state which caused the function to exit
Direction	output
Class	data
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	Out0 = “” in case the Status is “false”

2.5.5.8.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **WaitForCF()**.

Analyzing Script

Name	Output Data Class	Description
WaitForCF 01	Offline	Waits for the specified state of the defined Configuration File and executes current analyzing program forward.

2.5.5.9 WaitForEDT() – WaitForEventByDateAndTime()

2.5.5.9.1 General

Properties

Property	Description
Full Name	WaitForEventByDateAndTime
Short Name	WaitForEDT
Description	Blocks the execution of the current calculation until the specified time event is being reached.
Parent Library	Advanced
Parent Category	Control
Equidistant Inputs	No
Supported Operating Modes	Sequential

Function Prototype

```
DataOrValue Status = WaitForEDT
(
    Value       Date = "",
    Value       Time = "",
    Value       RptRate = 3600,
    DataOrValue EN = true
);
```

Symbol

N/A

2.5.5.9.2 Inputs

Date

Property	Description
Full Name	Date
Short Name	Date
Description	specifies the date at which the first start shall be issued, in format “yyyy-mm-dd”, e.g. “2001-01-01”
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains start event by date
Remark	none

Time

Property	Description
Full Name	Time
Short Name	Time
Description	specifies the time at which the first start shall be issued, in format “hh:mm:ss”, e.g. “10:01:01”
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains start event by time
Remark	none

RptRate

Property	Description
Full Name	RepeatRate
Short Name	RptRate
Description	contains the repetition rate after the first start date and time at which the following starts shall be issued
Direction	input
Class	parameter
Default Value	3600
Unit	s
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	0 … 100000000
Remark	In case this **ANF** is being called after the specified Date and Time and the Repeat Rate is non-zero, it waits until “Start Date/Time + next integer multiple of the RepeatRate” which is after the Date/Time of the first check which is being performed by this **ANF**. In case this **ANF** is being called after the specified Date and Time and the Repeat Rate is zero, the **ANF** exits immediately.

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.5.9.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates how the function has exited
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the specified event has been reached and Status = false in case the specified event could not be reached, e.g. in case the **ANF** is being called after the specified date and time and the repeat rate is 0

2.5.5.9.4 Examples

Analyzing Model

The usage within Analyzing Models is not supported by the Analyzing Function **WaitForEDT()**.

Analyzing Script

Name	Output Data Class	Description
WaitForEDT 01	Offline	Waits for the specified time event and executes current analyzing program forward.

2.5.6 Conversion

2.5.6.1 ConvCToSp() - ConvertCartesianToSpherical()

2.5.6.1.1 General

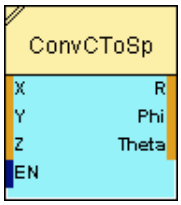
Properties

Property	Description
Full Name	ConvertCartesianToSpherical
Short Name	ConvCToSp
Description	Converts Cartesian coordinates to the spherical coordinate system.
Parent Library	Base
Parent Category	Conversion
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue R = ConvCToSp
(
    DataOrValue  X = 0,
    DataOrValue  Y = 0,
    DataOrValue  Z = 0,
    DataOrValue  EN = true,
    DataOrValue* Phi = NULL,
    DataOrValue* Theta = NULL
);
```

Symbol



2.5.6.1.2 Inputs

X

Property	Description
Full Name	Abscissa
Short Name	X
Description	first input value representing x-coordinate
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Y

Property	Description
Full Name	Ordinate
Short Name	Y
Description	second input value representing y-coordinate
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Z

Property	Description
Full Name	Vertical
Short Name	Z
Description	third input value representing z-coordinate
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.6.1.3 Outputs

R

Property	Description
Full Name	Radial
Short Name	R
Description	receives the result of the point radial distance calculation from the fixed origin
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Phi

Property	Description
Full Name	Phi
Short Name	Phi
Description	receives the result of the point horizontal azimuth angle calculation measured on the xy-plane
Direction	output
Class	data
Default Value	0
Unit	rad
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	the angle is given in radians between 0 and 2π

Theta

Property	Description
Full Name	Theta
Short Name	Theta
Description	receives the result of the point zenith angle calculation measured to the positive z-axis
Direction	output
Class	data
Default Value	0
Unit	rad
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	the angle is given in radians between 0 and π

2.5.6.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
ConvCToSp 01	Online	Converts generated signals from Cartesian to spherical coordinates.
ConvCToSp 02	Offline	Converts offline signals from Cartesian to spherical coordinates.

Analyzing Script

Name	Output Data Class	Description
ConvCToSp 01	Online	Converts generated signals from Cartesian to spherical coordinates.
ConvCToSp 02	Offline	Converts offline signals from Cartesian to spherical coordinates.

2.5.6.2 ConvSpm() - ConvertSpectrum()

2.5.6.2.1 General

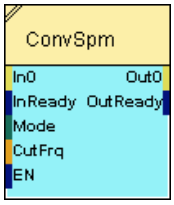
Properties

Property	Description
Full Name	ConvertSpectrum
Short Name	ConvSpm
Description	Converts an input spectrum.
Parent Library	Advanced
Parent Category	Conversion
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
Spectrum Out0 = ConvSpm
(
    Spectrum          In0 = 0,
    DataOrValue       InReady = false,
    Value             Mode = "Integrate",
    DataOrValue       CutFrq = 1,
    DataOrValue       EN = true,
    DataOrValue*      OutReady = NULL
);
```

Symbol



2.5.6.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	order or frequency spectrum which shall be converted
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the spectrum data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Mode

Property	Description
Full Name	Mode
Short Name	Mode
Description	mode which shall be used
Direction	input
Class	parameter
Default Value	“Integrate”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Integrate”, “Derivate”
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the function
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	all amplitudes lower to the cutoff frequency will be set to 0

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.6.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the converted spectrum
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the spectrum data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.6.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
ConvSpm 01	Online	Integrates the frequency spectrum of a generated signal.
ConvSpm 02	Offline	Integrates the frequency spectrum of an offline signal.

Analyzing Script

Name	Output Data Class	Description
ConvSpm 01	Online	Integrates the frequency spectrum of a generated signal.
ConvSpm 02	Offline	Integrates the frequency spectrum of an offline signal.

2.5.6.3 ConvSpToC() - ConvertSphericalToCartesian()

2.5.6.3.1 General

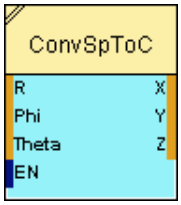
Properties

Property	Description
Full Name	ConvertSphericalToCartesian
Short Name	ConvSpToC
Description	Converts spherical coordinates to the Cartesian coordinate system.
Parent Library	Base
Parent Category	Conversion
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue X = ConvSpToC
(
    DataOrValue  R = 0,
    DataOrValue  Phi = 0,
    DataOrValue  Theta = 0,
    DataOrValue  EN = true,
    DataOrValue* Y = NULL,
    DataOrValue* Z = NULL
);
```

Symbol



2.5.6.3.2 Inputs

R

Property	Description
Full Name	Radial
Short Name	R
Description	first input value representing the point radial distance from the fixed origin
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

Phi

Property	Description
Full Name	Phi
Short Name	Phi
Description	second input value representing the point horizontal azimuth angle measured on the xy-plane
Direction	input
Class	data
Default Value	0
Unit	rad
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … 2π)
Remark	none

Theta

Property	Description
Full Name	Theta
Short Name	Theta
Description	third input value representing the point zenith angle measured to the positive z-axis
Direction	input
Class	data
Default Value	0
Unit	rad
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … π
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.6.3.3 Outputs

X

Property	Description
Full Name	Abscissa
Short Name	X
Description	receives the result of the x-coordinate calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Y

Property	Description
Full Name	Ordinate
Short Name	Y
Description	receives the result of the y-coordinate calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Z

Property	Description
Full Name	Vertical
Short Name	Z
Description	receives the result of the z-coordinate calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.6.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
ConvSpToC 01	Online	Converts generated signals from spherical to Cartesian coordinates.
ConvSpToC 02	Offline	Converts offline signals from spherical to Cartesian coordinates.

Analyzing Script

Name	Output Data Class	Description
ConvSpToC 01	Online	Converts generated signals from spherical to Cartesian coordinates.
ConvSpToC 02	Offline	Converts offline signals from spherical to Cartesian coordinates.

2.5.6.4 ConvVToB1D () - ConvertValueToBuffer1D ()

2.5.6.4.1 General

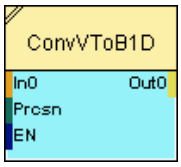
Properties

Property	Description
Full Name	ConvertValueToBuffer1D
Short Name	ConvVToB1D
Description	Converts input data or values to a 1-dimensional buffer.
Parent Library	Base
Parent Category	Conversion
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = ConvVToB1D
(
    DataOrValue  In0 ... In31 = 1;
    Value        Prcsn = "Decimal64",
    DataOrValue  EN = true
);
```

Symbol



2.5.6.4.2 Inputs

In0 … In31

Property	Description
Full Name	Input0 … Input31
Short Name	In0 … In31
Description	data or values which shall be converted
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	32
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_31

Property	Description
Full Name	Input0_31
Short Name	In0_31
Description	determines how many of the input arguments are used by the conversion
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 32
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the data
Direction	input
Class	parameter
Default Value	“Decimal64”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Boolean“, “Integer8“, “Integer16“, “Integer32“, “Integer64“, “Decimal32“, “Decimal64“
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.6.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 1-dimensional buffer with the input data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

2.5.6.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
ConvVToB1D 01	Online	Converts several generated signals into a 1-dimensional buffer.
ConvVToB1D 02	Offline	Converts offline signals into a 1-dimensional buffer.

Analyzing Script

Name	Output Data Class	Description
ConvVToB1D 01	Online	Converts several generated signals into a 1-dimensional buffer.
ConvVToB1D 02	Offline	Converts offline signals into a 1-dimensional buffer.

2.5.6.5 GetVFrmB1D() - GetValueFromBuffer1D()

2.5.6.5.1 General

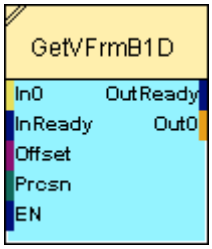
Properties

Property	Description
Full Name	GetValueFromBuffer1D
Short Name	GetVFrmB1D
Description	Extracts one or multiple values out of a 1-dimensional buffer.
Parent Library	Base
Parent Category	Conversion
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue OutReady = GetVFrmB1D
(
    Buffer1D      In0 = 0,
    DataOrValue   InReady = false,
    Value         Offset = 0,
    Value         Prcsn = "Decimal64",
    DataOrValue   EN = true,
    DataOrValue* Out0 ... Out7 = NULL
);
```

Symbol



2.5.6.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer from which the value(s) shall be extracted
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Offset

Property	Description
Full Name	Offset
Short Name	Offset
Description	specifies the offset within the 1-dimensional buffer at which the value of interest is located
Direction	input
Class	parameter
Default Value	0
Unit	bytes
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	0 … max of Integer32
Remark	the offset begins from 0

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	specifies the data type of the value(s) which shall be extracted
Direction	input
Class	parameter
Default Value	“Decimal64”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Boolean“, “Integer8“, “Integer16“, “Integer32“, “Integer64“, “Decimal32“, “Decimal64“
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.6.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 1-dimensional buffer with the input data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

2.5.6.5.4 Remarks

GetVFrmB1D() typically is being used in order to extract data out of the 1-dimensional buffer which is being received by the **Analyzing Function** InTCP(). However, it also can be used for extracting of data from a 1-dimensional buffer from any other data source.

2.5.6.5.5 Examples

Analyzing Model

Name	Output Data Class	Description
InTCP 01	Online	Extracts the data which is being received by the **Analyzing Function** InTCP().

Analyzing Script

Name	Output Data Class	Description
InTCP 01	Online	Extracts the data which is being received by the **Analyzing Function** InTCP().

2.5.6.6 GetVFrmSD() - GetValueFromSimpleData()

2.5.6.6.1 General

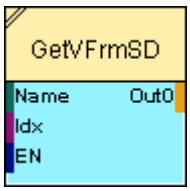
Properties

Property	Description
Full Name	GetValueFromSimpleData
Short Name	GetVFrmSD
Description	Extracts one value out of a time series of simple data.
Parent Library	Base
Parent Category	Conversion
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Value Out0 = GetVFrmSD
(
    Value        Name = "",
    DataOrValue  Idx = 0,
    DataOrValue  EN = true
);
```

Symbol



2.5.6.6.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the simple data from the system which shall be read
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the simple data from the system
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Index

Property	Description
Full Name	Index
Short Name	Idx
Description	specifies the index within the buffer of the input data which shall be provided to the output
Direction	input
Class	control
Default Value	-1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	in case -1 is provided as index, the most recent (= newest) value of the data is returned for online data, values different to -1 do not make sense because the valid range of indexes changes constantly in the area of ~ “BufferGeneration * RingBufferSize … (BufferGeneration +1) * RingBufferSize”

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.6.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the read data value
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	this output always returns a single value, also in sequential operating mode

2.5.6.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
GetVFrmSD 01	Online	The Analyzing Model “Abs 01” must be running before “GetVFrmSD 01” can run. Extracts one value out of the provided input data.
GetVFrmSD 02	Offline	The Analyzing Model “Abs 02” must have been executed before “GetVFrmSD 02” can run. Extracts one value out of the provided input data.

Analyzing Script

Name	Output Data Class	Description
GetVFrmSD 01	Online	The Analyzing Script “Abs 01” must be running before “GetVFrmSD 01” can run. Extracts one value out of the provided input data.
GetVFrmSD 02	Offline	The Analyzing Script “Abs 02” must have been executed before “GetVFrmSD 02” can run. Extracts one value out of the provided input data.

2.5.7 Filter

2.5.7.1 BP() - BandPassFilter()

2.5.7.1.1 General

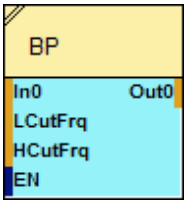
Properties

Property	Description
Full Name	BandPassFilter
Short Name	BP
Description	First order Butterworth band pass filter for the input data or values.
Parent Library	Base
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = BP
(
    DataOrValue In0 = 0,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency LCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency HCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
BP 01	Online	Uses a first order band pass filter on a generated rectangle signal and calculates the spectra.
BP 02	Offline	Uses a first order band pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
BP 01	Online	Uses a first order band pass filter on a generated rectangle signal.
BP 02	Offline	Uses a first order band pass filter on an offline rectangle signal.

2.5.7.2 BS() - BandStopFilter()

2.5.7.2.1 General

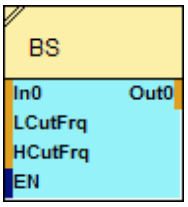
Properties

Property	Description
Full Name	BandStopFilter
Short Name	BS
Description	First order Butterworth band stop filter for the input data or values.
Parent Library	Base
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = BS
(
    DataOrValue In0 = 0,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency LCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency HCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
BS 01	Online	Uses a first order band stop filter on a generated rectangle signal and calculates the spectra.
BS 02	Offline	Uses a first order band stop filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
BS 01	Online	Uses a first order band stop filter on a generated rectangle signal.
BS 02	Offline	Uses a first order band stop filter on an offline rectangle signal.

2.5.7.3 HP() - HighPassFilter()

2.5.7.3.1 General

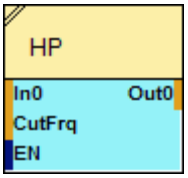
Properties

Property	Description
Full Name	HighPassFilter
Short Name	HP
Description	First order Butterworth high pass filter for the input data or values.
Parent Library	Base
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = HP
(
    DataOrValue In0 = 0,
    Value       CutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.3.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency CutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
HP 01	Online	Uses a first order high pass filter on a generated rectangle signal and calculates the spectra.
HP 02	Offline	Uses a first order high pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
HP 01	Online	Uses a first order high pass filter on a generated rectangle signal.
HP 02	Offline	Uses a first order high pass filter on an offline rectangle signal.

2.5.7.4 LP() - LowPassFilter()

2.5.7.4.1 General

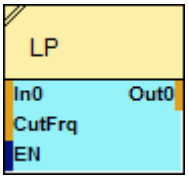
Properties

Property	Description
Full Name	LowPassFilter
Short Name	LP
Description	First order Butterworth low pass filter for the input data or values.
Parent Library	Base
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = LP
(
    DataOrValue In0 = 0,
    Value       CutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.4.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency CutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
LP 01	Online	Uses a first order low pass filter on a generated rectangle signal and calculates the spectra.
LP 02	Offline	Uses a first order low pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
LP 01	Online	Uses a first order low pass filter on a generated rectangle signal.
LP 02	Offline	Uses a first order low pass filter on an offline rectangle signal.

2.5.7.5 IIRBPBT() - IIRBandPassFilterButterworth()

2.5.7.5.1 General

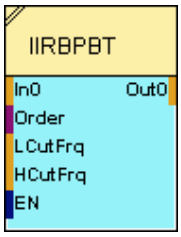
Properties

Property	Description
Full Name	IIRBandPassFilterButterworth
Short Name	IIRBPBT
Description	Butterworth band pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRBPBT
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1, 2, 3, … , 10
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency LCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency HCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRBPBT 01	Online	Uses a tenth order band pass filter on a generated rectangle signal and calculates the spectra.
IIRBPBT 02	Offline	Uses a tenth order band pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRBPBT 01	Online	Uses a tenth order band pass filter on a generated rectangle signal.
IIRBPBT 02	Offline	Uses a tenth order band pass filter on an offline rectangle signal.

2.5.7.6 IIRBSBT() - IIRBandStopFilterButterworth()

2.5.7.6.1 General

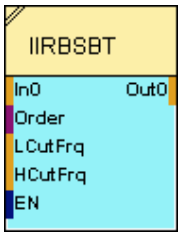
Properties

Property	Description
Full Name	IIRBandStopFilterButterworth
Short Name	IIRBSBT
Description	Butterworth band stop filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRBSBT
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.6.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … , 10
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRBSBT 01	Online	Uses a tenth order band stop filter on a generated rectangle signal and calculates the spectra.
IIRBSBT 02	Offline	Uses a tenth order band stop filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRBSBT 01	Online	Uses a tenth order band stop filter on a generated rectangle signal and calculates the spectra.
IIRBSBT 02	Offline	Uses a tenth order band stop filter on an offline rectangle signal.

2.5.7.7 IIRHPBT() - IIRHighPassFilterButterworth()

2.5.7.7.1 General

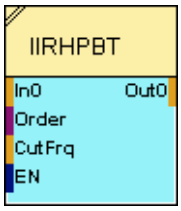
Properties

Property	Description
Full Name	IIRHighPassFilterButterworth
Short Name	IIRHPBT
Description	Butterworth high pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRHPBT
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       CutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.7.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … 20
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.7.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.7.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRHPBT 01	Online	Uses a fifteenth order high pass filter on a generated rectangle signal and calculates the spectra.
IIRHPBT 02	Offline	Uses a twentieth order high pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRHPBT 01	Online	Uses a fifteenth order high pass filter on a generated rectangle signal.
IIRHPBT 02	Offline	Uses a tenth order high pass filter on an offline rectangle signal.

2.5.7.8 IIRLPBT() - IIRLowPassFilterButterworth()

2.5.7.8.1 General

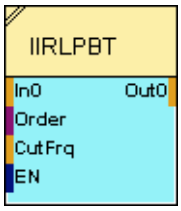
Properties

Property	Description
Full Name	IIRLowPassFilterButterworth
Short Name	IIRLPBT
Description	Butterworth low pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRLPBT
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       CutFrq = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.8.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … 20
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.8.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.8.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRLPBT 01	Online	Uses a fifteenth order low pass filter on a generated rectangle signal and calculates the spectra.
IIRLPBT 02	Offline	Uses a twentieth order low pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRLPBT 01	Online	Uses a fifteenth order low pass filter on a generated rectangle signal.
IIRLPBT 02	Offline	Uses a tenth order low pass filter on an offline rectangle signal.

2.5.7.9 IIRBPCH1() – IIRBandPassFilterChebyshev1()

2.5.7.9.1 General

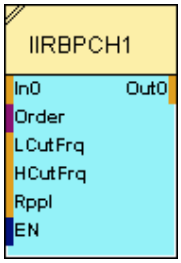
Properties

Property	Description
Full Name	IIRBandPassFilterChebyshev1
Short Name	IIRBPCH1
Description	Chebyshev1 band pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRBPCH1
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.9.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1, 2, 3, … , 10
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency LCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency HCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the passband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.9.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.9.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRBPCH1 01	Online	Uses a tenth order band pass filter on a generated rectangle signal and calculates the spectra.
IIRBPCH1 02	Offline	Uses a tenth order band pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRBPCH1 01	Online	Uses a tenth order band pass filter on a generated rectangle signal.
IIRBPCH1 02	Offline	Uses a tenth order band pass filter on an offline rectangle signal.

2.5.7.10 IIRBSCH1() – IIRBandStopFilterChebyshev1()

2.5.7.10.1 General

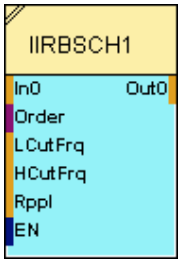
Properties

Property	Description
Full Name	IIRBandStopFilterChebyshev1
Short Name	IIRBSCH1
Description	Chebyshev1 band stop filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRBSCH1
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.10.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … , 10
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the passband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.10.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.10.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRBSCH1 01	Online	Uses a tenth order band stop filter on a generated rectangle signal and calculates the spectra.
IIRBSCH1 02	Offline	Uses a tenth order band stop filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRBSCH1 01	Online	Uses a tenth order band stop filter on a generated rectangle signal.
IIRBSCH1 02	Offline	Uses a tenth order band stop filter on an offline rectangle signal.

2.5.7.11 IIRHPCH1() – IIRHighPassFilterChebyshev1()

2.5.7.11.1 General

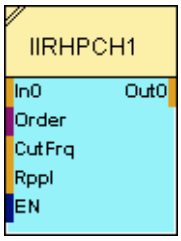
Properties

Property	Description
Full Name	IIRHighPassFilterChebyshev1
Short Name	IIRHPCH1
Description	Chebyshev1 high pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRHPCH1
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       CutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.11.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … 20
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the passband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.11.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.11.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRHPCH1 01	Online	Uses a fifteenth order high pass filter on a generated rectangle signal and calculates the spectra.
IIRHPCH1 02	Offline	Uses a twentieth order high pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRHPCH1 01	Online	Uses a fifteenth order high pass filter on a generated rectangle signal.
IIRHPCH1 02	Offline	Uses a tenth order high pass filter on an offline rectangle signal.

2.5.7.12 IIRLPCH1() – IIRLowPassFilterChebyshev1()

2.5.7.12.1 General

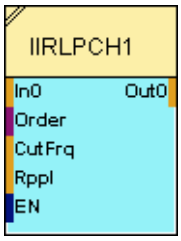
Properties

Property	Description
Full Name	IIRLowPassFilterChebyshev1
Short Name	IIRLPCH1
Description	Chebyshev1 low pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRLPCH1
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       CutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.12.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … 20
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the passband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.12.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.12.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRLPCH1 01	Online	Uses a fifteenth order low pass filter on a generated rectangle signal and calculates the spectra.
IIRLPCH1 02	Offline	Uses a twentieth order low pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRLPCH1 01	Online	Uses a fifteenth order low pass filter on a generated rectangle signal.
IIRLPCH1 02	Offline	Uses a tenth order low pass filter on an offline rectangle signal.

2.5.7.13 IIRBPCH2() – IIRBandPassFilterChebyshev2()

2.5.7.13.1 General

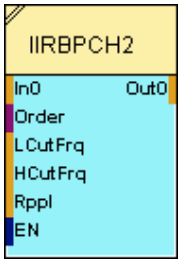
Properties

Property	Description
Full Name	IIRBandPassFilterChebyshev2
Short Name	IIRBPCH2
Description	Chebyshev2 band pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRBPCH2
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.13.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1, 2, 3, … , 10
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency LCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency HCutFrq the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the stopband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.13.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.13.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRBPCH2 01	Online	Uses a tenth order band pass filter on a generated rectangle signal and calculates the spectra.
IIRBPCH2 02	Offline	Uses a tenth order band pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRBPCH2 01	Online	Uses a tenth order band pass filter on a generated rectangle signal.
IIRBPCH2 02	Offline	Uses a tenth order band pass filter on an offline rectangle signal.

2.5.7.14 IIRBSCH2() – IIRBandStopFilterChebyshev2()

2.5.7.14.1 General

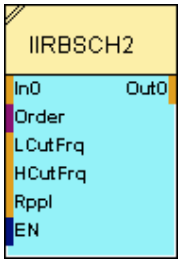
Properties

Property	Description
Full Name	IIRBandStopFilterChebyshev2
Short Name	IIRBSCH2
Description	Chebyshev2 band stop filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRBSCH2
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       LCutFrq = 0,
    Value       HCutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.14.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … , 10
Remark	none

LCutFrq

Property	Description
Full Name	LowCutoffFrequency
Short Name	LCutFrq
Description	lower cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	0
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

HCutFrq

Property	Description
Full Name	HighCutoffFrequency
Short Name	HCutFrq
Description	higher cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the stopband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.14.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.14.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRBSCH2 01	Online	Uses a tenth order band stop filter on a generated rectangle signal and calculates the spectra.
IIRBSCH2 02	Offline	Uses a tenth order band stop filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRBSCH2 01	Online	Uses a tenth order band stop filter on a generated rectangle signal.
IIRBSCH2 02	Offline	Uses a tenth order band stop filter on an offline rectangle signal.

2.5.7.15 IIRHPCH2() – IIRHighPassFilterChebyshev2()

2.5.7.15.1 General

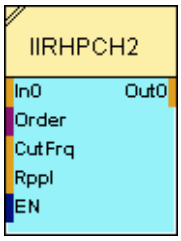
Properties

Property	Description
Full Name	IIRHighPassFilterChebyshev2
Short Name	IIRHPCH2
Description	Chebyshev2 high pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRHPCH2
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       CutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.15.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … 20
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the stopband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.15.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.15.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRHPCH2 01	Online	Uses a fifteenth order high pass filter on a generated rectangle signal and calculates the spectra.
IIRHPCH2 02	Offline	Uses a twentieth order high pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRHPCH2 01	Online	Uses a fifteenth order high pass filter on a generated rectangle signal.
IIRHPCH2 02	Offline	Uses a tenth order high pass filter on an offline rectangle signal.

2.5.7.16 IIRLPCH2() – IIRLowPassFilterChebyshev2()

2.5.7.16.1 General

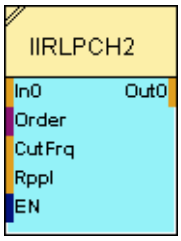
Properties

Property	Description
Full Name	IIRLowPassFilterChebyshev2
Short Name	IIRLPCH2
Description	Chebyshev2 low pass filter for the input data or values.
Parent Library	Advanced
Parent Category	Filter
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = IIRLPCH2
(
    DataOrValue In0 = 0,
    Value       Order = 1,
    Value       CutFrq = 1,
    Value       Rppl = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.7.16.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be filtered
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Order

Property	Description
Full Name	FilterOrder
Short Name	Order
Description	filter order
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1, 2, 3, … 20
Remark	none

CutFrq

Property	Description
Full Name	CutoffFrequency
Short Name	CutFrq
Description	cutoff frequency of the filter
Direction	input
Class	parameter
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	by definition of the cutoff frequency the system sampling rate (`ANS_VARIABLE_NAME_CYCLE_TIME`) must be considered; the system sample frequency must be at least the double of the defined cutoff frequency

Rppl

Property	Description
Full Name	Ripple
Short Name	Rppl
Description	ripple depth for the stopband
Direction	input
Class	parameter
Default Value	1
Unit	dB
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0.1 … 3
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.7.16.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the filtering
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.7.16.4 Examples

Analyzing Model

Name	Output Data Class	Description
IIRLPCH2 01	Online	Uses a fifteenth order low pass filter on a generated rectangle signal and calculates the spectra.
IIRLPCH2 02	Offline	Uses a twentieth order low pass filter on an offline rectangle signal and calculates the spectra.

Analyzing Script

Name	Output Data Class	Description
IIRLPCH2 01	Online	Uses a fifteenth order low pass filter on a generated rectangle signal.
IIRLPCH2 02	Offline	Uses a tenth order low pass filter on an offline rectangle signal.

2.5.8 Generation

2.5.8.1 GenWF() - GenerateWaveForm()

2.5.8.1.1 General

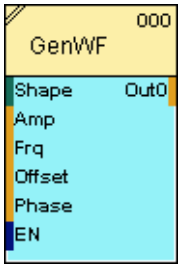
Properties

Property	Description
Full Name	GenerateWaveForm
Short Name	GenWF
Description	Generates a periodical signal with a chosen shape.
Parent Library	Base
Parent Category	Generation
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = GenWF
(
    Value       Shape = "Sine",
    DataOrValue Amp = 1,
    DataOrValue Frq = 1,
    DataOrValue Offset = 0,
    Value       Phase = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.8.1.2 Inputs

Shape

Property	Description
Full Name	Shape
Short Name	Shape
Description	form of the wave which shall be generated
Direction	input
Class	parameter
Default Value	"Sine"
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	"Sine", "Cosine", "Rectangle", "Triangle", "Sawtooth"
Remark	none

Amp

Property	Description
Full Name	Amplitude
Short Name	Amp
Description	amplitude of the waveform which shall be generated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

Frq

Property	Description
Full Name	Frequency
Short Name	Frq
Description	frequency of the waveform which shall be generated
Direction	input
Class	data
Default Value	1
Unit	Hz
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

Offset

Property	Description
Full Name	Offset
Short Name	Offset
Description	steady component of the waveform which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Phase

Property	Description
Full Name	Phase
Short Name	Phase
Description	phase-shift of the waveform which shall be generated
Direction	input
Class	parameter
Default Value	0
Unit	rad
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.8.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the values which has been generated
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.8.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
GenWF 01	Online	Generates different kinds of signals.
GenWF 02	Offline	Generates 2 modulated sine waves depending on 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
GenWF 01	Online	Generates different kinds of signals.

2.5.8.2 Impulser() - Impulser()

2.5.8.2.1 General

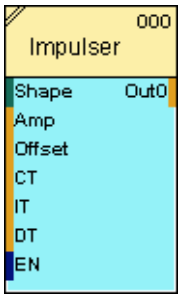
Properties

Property	Description
Full Name	Impulser
Short Name	Impulser
Description	Generates an impulse signal with a chosen shape.
Parent Library	Base
Parent Category	Generation
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Impulser
(
    Value       Shape = "Rectangle",
    DataOrValue Amp = 1,
    DataOrValue Offset = 0,
    DataOrValue CT = 1,
    DataOrValue IT = 0.1,
    Value       DT = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.8.2.2 Inputs

Shape

Property	Description
Full Name	Shape
Short Name	Shape
Description	form of the impulse which shall be generated
Direction	input
Class	parameter
Default Value	"Rectangle"
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	"Rectangle"
Remark	none

Amp

Property	Description
Full Name	Amplitude
Short Name	Amp
Description	amplitude of the impulse which shall be generated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Offset

Property	Description
Full Name	Offset
Short Name	Offset
Description	steady component of the impulse which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CT

Property	Description
Full Name	CycleTime
Short Name	CT
Description	cycle time of the impulse signal which shall be generated
Direction	input
Class	data
Default Value	1
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

IT

Property	Description
Full Name	ImpulseTime
Short Name	IT
Description	duration of the impulse which shall be generated
Direction	input
Class	data
Default Value	0.1
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

DT

Property	Description
Full Name	DelayTime
Short Name	DT
Description	delay time of the impulse signal which shall be generated
Direction	input
Class	parameter
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.8.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the values which has been generated
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.8.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
Impulser 01	Online	Generates an impulse signal with variable cycle time.
Impulser 02	Offline	Generates modulated impulse signal depending on offline signal.

Analyzing Script

Name	Output Data Class	Description
Impulser 01	Online	Generates an impulse signal with variable cycle time.
Impulser 02	Offline	Generates modulated impulse signal depending on offline signal.

2.5.8.3 Random() - Random()

2.5.8.3.1 General

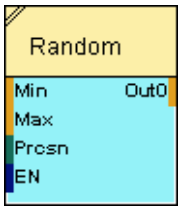
Properties

Property	Description
Full Name	Random
Short Name	Random
Description	Generates random values within a defined range.
Parent Library	Base
Parent Category	Generation
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Random
(
    DataOrValue Min = 0,
    DataOrValue Max = 1,
    Value       Prcsn = "Decimal64",
    DataOrValue EN = true
);
```

Symbol



2.5.8.3.2 Inputs

Min

Property	Description
Full Name	Minimum
Short Name	Min
Description	minimum level of the random values which shall be generated
Direction	input
Class	data
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Max

Property	Description
Full Name	Maximum
Short Name	Max
Description	maximum level of the random values which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the random values which shall be generated
Direction	input
Class	parameter
Default Value	“Decimal64”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Integer32”, “Decimal64”
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.8.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the random values which has been generated
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.8.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
Random 01	Online	Generates a random signal within variable range.
Random 02	Offline	Generates modulated random signal depending on 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
Random 01	Online	Generates a random signal within variable range.
Random 02	Offline	Generates modulated random signal depending on 2 offline signals.

2.5.9 Input

2.5.9.1 In() - Input()

2.5.9.1.1 General

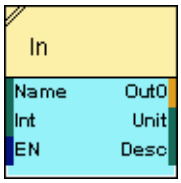
Properties

Property	Description
Full Name	Input
Short Name	In
Description	Reads simple data from the system, which can be used by following Analyzing Functions afterwards.
Parent Library	Base
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = In
(
    Value       Name = 0,
    Value       Int = "Default",
    DataOrValue EN = true,
    Value*      Unit = NULL,
    Value*      Desc = NULL
);
```

Symbol



2.5.9.1.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the simple data from the system which shall be read or directly the value which shall be provided to Output0
Direction	input
Class	parameter
Default Value	""
Unit	1 or n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the simple data from the system or a value with min … max of the supported precision of the parent Analyzing Model/Analyzing Script
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Int

Property	Description
Full Name	Interpolation
Short Name	Int
Description	determines the interpolation method which is used when reading the data from the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Stairs”, “Linear”, “Nearest Neighbor”
Remark	in case the argument is set to "Default" but the currently chosen "Default Input Data Interpolation Mode" of the owning AME/ASE is not supported by the ANF, "Stairs" is being chosen as interpolation mode

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the data interpolation from the system
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	receives the unit of the data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	receives the description of the data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.9.2 InAAF() - InputAntiAliasingFilter()

2.5.9.2.1 General

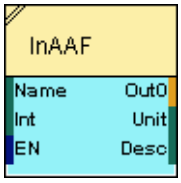
Properties

Property	Description
Full Name	Input
Short Name	In
Description	Reads anti aliasing filtered simple data from the system which can be used by following Analyzing Functions afterwards.
Parent Library	Base
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = InAAF
(
    Value       Name = 0,
    Value       Int = "Default",
    DataOrValue EN = true,
    Value*      Unit = NULL,
    Value*      Desc = NULL
);
```

Symbol



2.5.9.2.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the simple data from the system which shall be read or directly the value which shall be provided to Output0
Direction	input
Class	parameter
Default Value	""
Unit	1 or n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the simple data from the system
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Int

Property	Description
Full Name	Interpolation
Short Name	Int
Description	determines the interpolation method which is used when reading the data from the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Stairs”, “Linear”, “Nearest Neighbor”
Remark	in case the argument is set to "Default" but the currently chosen "Default Input Data Interpolation Mode" of the owning AME/ASE is not supported by the ANF, "Stairs" is being chosen as interpolation mode

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the data interpolation from the system
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	receives the unit of the data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	receives the description of the data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.9.3 InB1D() - InputBuffer1D()

2.5.9.3.1 General

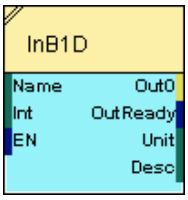
Properties

Property	Description
Full Name	InputBuffer1D
Short Name	InB1D
Description	Reads 1-dimensional buffer data from the system, which can be used by following Analyzing Functions afterwards.
Parent Library	Base
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = InB1D
(
    Value         Name = "",
    Value         Int = "Default",
    DataOrValue   EN = true,
    DataOrValue*  OutReady = NULL,
    Value*        Unit = NULL,
    Value*        Desc = NULL
);
```

Symbol



2.5.9.3.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the 1-dimensional buffer data from the system which shall be provided to Output0
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the 1-dimensional buffer data from the system
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Int

Property	Description
Full Name	Interpolation
Short Name	Int
Description	determines the interpolation method which is used when reading the buffer data from the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Stairs”, “None”, “Nearest Neighbor”
Remark	in case the argument is set to "Default" but the currently chosen "Default Input Data Interpolation Mode" of the owning AME/ASE is not supported by the ANF, "Stairs" is being chosen as interpolation mode

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the buffer data interpolation from the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Outout0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	receives the unit of the buffer data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	receives the description of the buffer data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.9.4 InB2D() – InputBuffer2D()

2.5.9.4.1 General

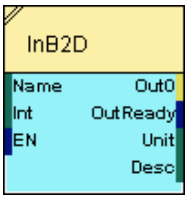
Properties

Property	Description
Full Name	InputBuffer2D
Short Name	InB2D
Description	Reads 2-dimensional buffer data from the system, which can be used by following Analyzing Functions afterwards.
Parent Library	Base
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer2D Out0 = InB2D
(
    Value         Name = "",
    Value         Int = "Default",
    DataOrValue   EN = true,
    DataOrValue*  OutReady = NULL,
    Value*        Unit = NULL,
    Value*        Desc = NULL
);
```

Symbol



2.5.9.4.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the 2-dimensional buffer data from the system which shall be provided to Output0
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the 2-dimensional buffer data from the system
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Int

Property	Description
Full Name	Interpolation
Short Name	Int
Description	determines the interpolation method which is used when reading the buffer data from the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Stairs”, “None”, “Nearest Neighbor”
Remark	in case the argument is set to "Default" but the currently chosen "Default Input Data Interpolation Mode" of the owning AME/ASE is not supported by the ANF, "Stairs" is being chosen as interpolation mode

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the buffer data interpolation from the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer2D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Outout0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	receives the unit of the buffer data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	receives the description of the buffer data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.9.5 InOrdRsmpl() – InputOrderResampling()

2.5.9.5.1 General

Properties

Property	Description
Full Name	InputOrderResampling
Short Name	InOrdRsmpl
Description	Calculates order re-sampled simple data from the system which can be used by following Analyzing Functions afterwards.
Parent Library	Advanced
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = InOrdRsmpl
(
    Value         Signal = "",
    DataOrValue   Trigger = "",
    Value         Int = "Default",
    Value         NoI = 1,
    Value         MaxOrder = 64,
    Value         NoP = 1024,
    Value         Lap = 0,
    DataOrValue   EN = true,
    DataOrValue*  OutReady = NULL,
    Value*        Unit = NULL,
    Value*        Desc = NULL
);
```

Symbol

2.5.9.5.2 Inputs

Signal

Property	Description
Full Name	SignalName
Short Name	Signal
Description	name of the simple data from the system which shall be read order re-sampled and shall be provided to Output0
Direction	input
Class	parameter
Default Value	“”
Unit	1 or n/a
Configurable	1
Supported Data Types	String, Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	string which contains the name of the simple data from the system
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Trigger

Property	Description
Full Name	TriggerName
Short Name	Trigger
Description	name of the trigger data which shall be used to re-sample the signal
Direction	input
Class	data
Default Value	false
Unit	1 or n/a
Configurable	1
Supported Data Types	String, Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	string which contains the name of the trigger data
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Int

Property	Description
Full Name	Interpolation
Short Name	Int
Description	determines the interpolation method which is used when reading the data from the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Stairs”, “None”, “Nearest Neighbor”
Remark	in case the argument is set to "Default" but the currently chosen "Default Input Data Interpolation Mode" of the owning AME/ASE is not supported by the ANF, "Stairs" is being chosen as interpolation mode

NoI

Property	Description
Full Name	NumberOfImpulses
Short Name	NoI
Description	determines number of impulses per revolution
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 8
Remark	none

MaxOrder

Property	Description
Full Name	MaximumOrder
Short Name	MaxOrder
Description	determines the maximum order of the afterwards calculated order spectrum
Direction	input
Class	parameter
Default Value	64
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points for the FFT calculation
Direction	input
Class	parameter
Default Value	1024
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536
Remark	none

Lap

Property	Description
Full Name	Overlap
Short Name	Lap
Description	defines the overlapping length of the input data or values
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … 1)
Remark	in case Overlap = 0 no overlapping will be occurred

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 1-dimensional buffer with the order re-sampled input data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Outout0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	receives the unit of the buffer data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	receives the description of the buffer data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.9.6 InSpm() - InputSpectrum()

2.5.9.6.1 General

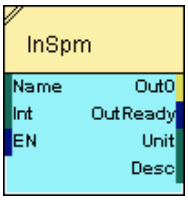
Properties

Property	Description
Full Name	InputSpectrum
Short Name	InSpm
Description	Reads spectrum data from the system, which can be used by following Analyzing Functions afterwards.
Parent Library	Base
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Spectrum Out0 = InSpm
(
    Value         Name = "",
    Value         Int = "Default",
    DataOrValue   EN = true,
    DataOrValue*  OutReady = NULL,
    Value*        Unit = NULL,
    Value*        Desc = NULL
);
```

Symbol



2.5.9.6.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the spectrum data from the system which shall be provided to Output0
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the spectrum data from the system
Remark	in case a data name prefix is being specified for input data of the parent Analyzing Model/Analyzing Script, the specified input data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

Int

Property	Description
Full Name	Interpolation
Short Name	Int
Description	determines the interpolation method which is used when reading the spectrum data from the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Stairs”, “None”, “Nearest Neighbor”
Remark	in case the argument is set to "Default" but the currently chosen "Default Input Data Interpolation Mode" of the owning AME/ASE is not supported by the ANF, "Stairs" is being chosen as interpolation mode

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the spectrum data interpolation from the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the spectrum data in Outout0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	receives the unit of the spectrum data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	receives the description of the spectrum data from the system
Direction	output
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.9.7 InTCP() - InputTCP()

2.5.9.7.1 General

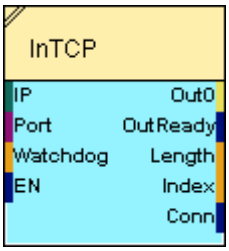
Properties

Property	Description
Full Name	InputTCP
Short Name	InTCP
Description	Receives a data buffer from the specified IP address.
Parent Library	Advanced
Parent Category	Input
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based

Function Prototype

```
Buffer1D Out0 = InTCP
(
    Value         IP = "",
    Value         Port = 1,
    Value         WatchDog = 0,
    DataOrValue   EN = true,
    DataOrValue* OutReady = NULL
    DataOrValue* Length = NULL,
    DataOrValue* Index = NULL,
    DataOrValue* Conn = NULL
);
```

Symbol



2.5.9.7.2 Inputs

IP

Property	Description
Full Name	IPAddress
Short Name	IP
Description	receives the IP address which is allowed to transmit data to this Analyzing Function
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the IP address
Remark	none

Port

Property	Description
Full Name	Port
Short Name	Port
Description	receives the port at which the Analyzing Function waits for a connection from the specified IP address
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 65535
Remark	none

Watchdog

Property	Description
Full Name	Watchdog
Short Name	Watchdog
Description	allows to monitor whether there are still packets arriving from the specified IP address
Direction	input
Class	parameter
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … 3600
Remark	in case the value of Watchdog is 0, the watchdog is disabled in case there is no TCP packet for longer than the specified watchdog time, the **Analyzing Function** breaks the current connection and starts to wait for new connections again

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal function
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.9.7.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the latest data buffer whenever a new data packet has been received from the specified IP address
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

Length

Property	Description
Full Name	Length
Short Name	Length
Description	contains the size of the buffer which is provided via Output0
Direction	output
Class	data
Default Value	0
Unit	bytes
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Index

Property	Description
Full Name	Index
Short Name	Index
Description	contains the index of the TCP telegram which has been received
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.9.7.4 Remarks

In order to use this **Analyzing Function**, a TCP client must be provided for the other side of the connection. The IP address of the TCP client must match the specified “IP” input arguments of this **Analyzing Function**. The **Analyzing Function** waits for a connecting TCP client at the specified “Port”.

In case multiple TCP packets arrive within two calculations of the parent Analyzing Model/Analyzing Script, only the data from the last TCP packet is used and the other TCP packets are discarded.

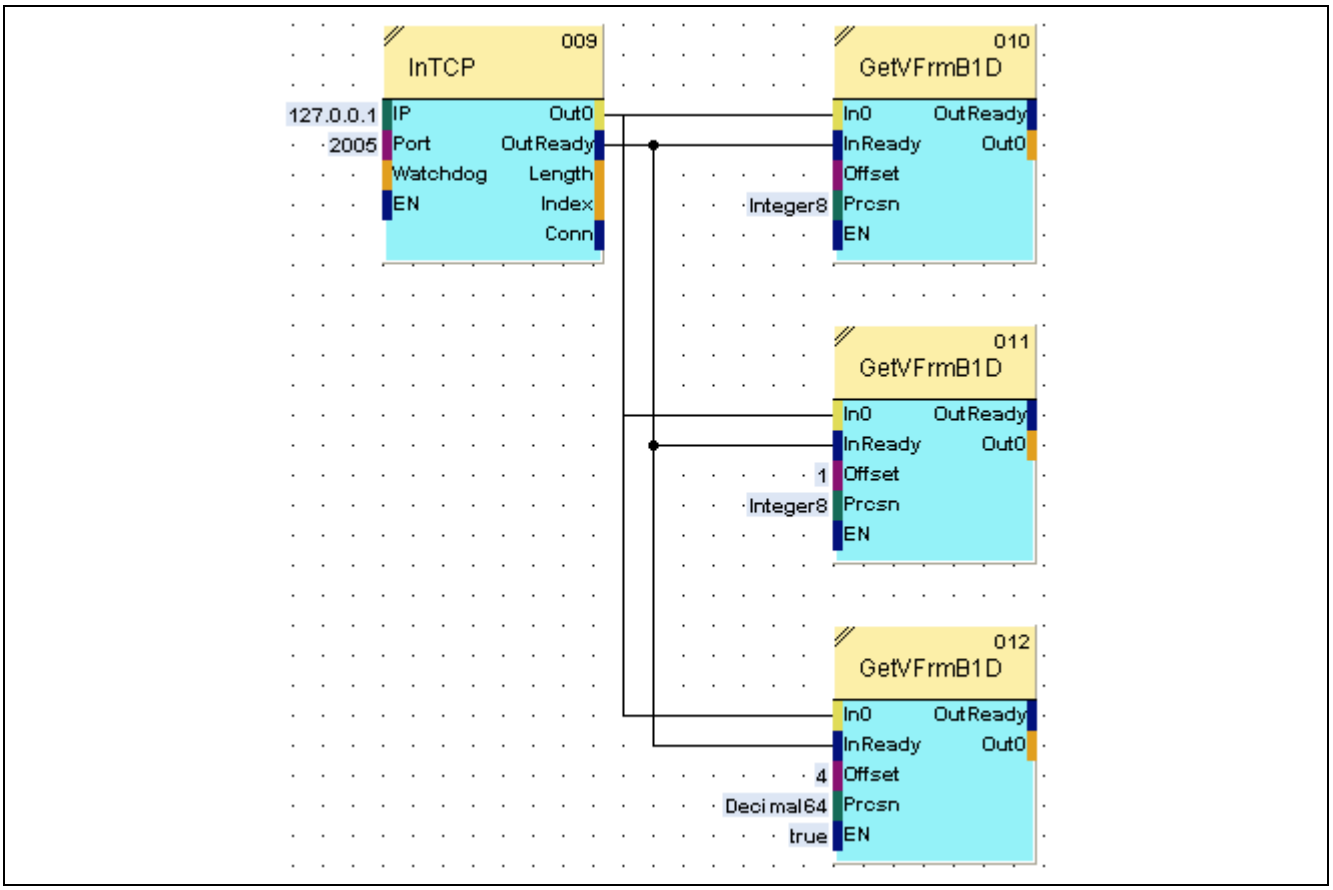
There is no handling of timestamps or time information - the last known TCP packet simply is inserted into the next calculation of a heart beat time.

Besides the standard TCP framework bytes, the following data is being expected by this **Analyzing Function**:

Length	Type	Data Type	Description
4 Byte	Timestamp	unsigned long	reserved for future use
4 Byte	Length	unsigned long	contains the length of the payload (without the “Timestamp”, “Length”, “Index” and “End Delimiter” fields) the length includes also includes possible present padding bytes the received length is output via the “Length” output argument
4 Byte	Index	unsigned long	contains the index the received index is not examined by the **Analyzing Function** but it is output via the “Index” output argument
x Byte	Payload	variable	contains the payload which is provided to “Output0”
4 Byte	End Delimiter	unsigned long	contains the end delimiter (0x7FFFFFFF)

The **Analyzing Function** GetVFrmB1D() must be used in order to extract the single values out of the “Output0” output of InTCP().

Example



2.5.9.7.5 Examples

Analyzing Model

Name	Output Data Class	Description
InTCP 01	Online	Different online data is received via TCP. This Analyzing Model also uses the **Analyzing Function** OutTCP() in order to provide the counterpart which sends the online data.

Analyzing Script

Name	Output Data Class	Description
InTCP 01	Online	Different online data is received via TCP. This Analyzing Script also uses the **Analyzing Function** OutTCP() in order to provide the counterpart which sends the online data.

2.5.10 Logic

2.5.10.1 AND() - AND()

2.5.10.1.1 General

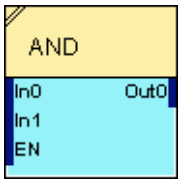
Properties

Property	Description
Full Name	AND
Short Name	AND
Description	Calculates the logical AND for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = AND
(
    DataOrValue In0 … In7 = false;
    DataOrValue EN = true
);
```

Symbol



2.5.10.1.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	binary data or values for the AND logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the AND logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
AND 01	Online	Calculates the logic AND for 2 generated signals.
AND 02	Offline	Calculates the logic AND for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
AND 01	Online	Calculates the logic AND for 2 generated signals.
AND 02	Offline	Calculates the logic AND for 2 offline signals.

2.5.10.2 NAND() - NAND()

2.5.10.2.1 General

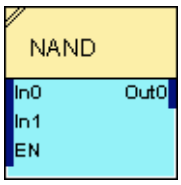
Properties

Property	Description
Full Name	NAND
Short Name	NAND
Description	Calculates the logical NAND for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = NAND
(
    DataOrValue In0 … In7 = false;
    DataOrValue EN = true
);
```

Symbol



2.5.10.2.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	binary data or values for the NAND logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the NAND logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
NAND 01	Online	Calculates the logic NAND for 2 generated signals.
NAND 02	Offline	Calculates the logic NAND for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
NAND 01	Online	Calculates the logic NAND for 2 generated signals.
NAND 02	Offline	Calculates the logic NAND for 2 offline signals.

2.5.10.3 NOR() - NOR()

2.5.10.3.1 General

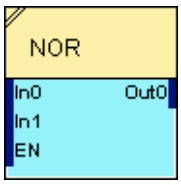
Properties

Property	Description
Full Name	NOR
Short Name	NOR
Description	Calculates the logical NOR for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = NOR
(
    DataOrValue In0 … In7 = false;
    DataOrValue EN = true
);
```

Symbol



2.5.10.3.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	binary data or values for the NOR logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the NOR logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
NOR 01	Online	Calculates the logic NOR for 2 generated signals.
NOR 02	Offline	Calculates the logic NOR for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
NOR 01	Online	Calculates the logic NOR for 2 generated signals.
NOR 02	Offline	Calculates the logic NOR for 2 offline signals.

2.5.10.4 NOT() - NOT()

2.5.10.4.1 General

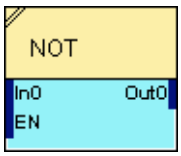
Properties

Property	Description
Full Name	NOT
Short Name	NOT
Description	Calculates the logical NOT for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = NOT
(
    DataOrValue In0 = false,
    DataOrValue EN = true
);
```

Symbol



2.5.10.4.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	binary data or values for the NOT logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the NOT logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
NOT 01	Online	Calculates the logic NOT for 2 generated signals.
NOT 02	Offline	Calculates the logic NOT for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
NOT 01	Online	Calculates the logic NOT for 2 generated signals.
NOT 02	Offline	Calculates the logic NOT for 2 offline signals.

2.5.10.5 NXOR() - NXOR()

2.5.10.5.1 General

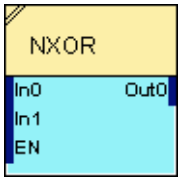
Properties

Property	Description
Full Name	NXOR
Short Name	NXOR
Description	Calculates the logical NXOR for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = NXOR
(
    DataOrValue In0 = false,
    DataOrValue In1 = false,
    DataOrValue EN = true
);
```

Symbol



2.5.10.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	first binary data or values for the NXOR logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of the Decimal64
Remark	none

In1

Property	Description
Full Name	Input1
Short Name	In1
Description	second binary data or values for the NXOR logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Deciamal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the NXOR logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
NXOR 01	Online	Calculates the logic NXOR for 2 generated signals.
NXOR 02	Offline	Calculates the logic NXOR for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
NXOR 01	Online	Calculates the logic NXOR for 2 generated signals.
NXOR 02	Offline	Calculates the logic NXOR for 2 offline signals.

2.5.10.6 OR() - OR()

2.5.10.6.1 General

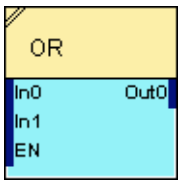
Properties

Property	Description
Full Name	OR
Short Name	OR
Description	Calculates the logical OR for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = OR
(
    DataOrValue In0 … In7 = false;
    DataOrValue EN = true
);
```

Symbol



2.5.10.6.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	binary data or values for the OR logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the calculation
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the OR logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
OR 01	Online	Calculates the logic OR for 2 generated signals.
OR 02	Offline	Calculates the logic OR for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
OR 01	Online	Calculates the logic OR for 2 generated signals.
OR 02	Offline	Calculates the logic OR for 2 offline signals.

2.5.10.7 RSFF() - RSFlipFlop()

2.5.10.7.1 General

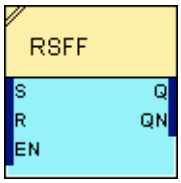
Properties

Property	Description
Full Name	RSFlipFlop
Short Name	RSFF
Description	Simulates an S-dominant RS Flip-Flop.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Q = RSFF
(
    DataOrValue  S = false,
    DataOrValue  R = false,
    DataOrValue  EN = true,
    DataOrValue* QN = NULL
);
```

Symbol



2.5.10.7.2 Inputs

S

Property	Description
Full Name	Set
Short Name	S
Description	binary data or values for the Set input
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

R

Property	Description
Full Name	Reset
Short Name	R
Description	binary data or values for the Reset input
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.7.3 Outputs

Q

Property	Description
Full Name	Q
Short Name	Q
Description	receives the state of the Flip Flop simulation
Direction	output
Class	data
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

QN

Property	Description
Full Name	QN
Short Name	QN
Description	receives the complement state of the Flip Flop simulation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.7.4 Examples

Analyzing Model

Name	Output Data Class	Description
RSFF 01	Online	Calculates the S-dominant RS Flip-Flop of 2 generated signals.
RSFF 02	Offline	Calculates the S-dominant RS Flip-Flop of 2 offline signal.

Analyzing Script

Name	Output Data Class	Description
RSFF 01	Online	Calculates the S-dominant RS Flip-Flop of 2 generated signals.
RSFF 02	Offline	Calculates the S-dominant RS Flip-Flop of 2 offline signal.

2.5.10.8 XOR() - XOR()

2.5.10.8.1 General

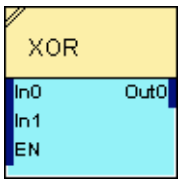
Properties

Property	Description
Full Name	XOR
Short Name	XOR
Description	Calculates the logical XOR for binary data or values.
Parent Library	Base
Parent Category	Logic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = XOR
(
    DataOrValue In0 = false,
    DataOrValue In1 = false,
    DataOrValue EN = true
);
```

Symbol



2.5.10.8.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	first binary data or values for the XOR logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In1

Property	Description
Full Name	Input1
Short Name	In1
Description	second binary data or values for the XOR logical operation
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.10.8.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the XOR logical operation
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.10.8.4 Examples

Analyzing Model

Name	Output Data Class	Description
XOR 01	Online	Calculates the logic XOR for 2 generated signals.
XOR 02	Offline	Calculates the logic XOR for 2 offline signals.

Analyzing Script

Name	Output Data Class	Description
XOR 01	Online	Calculates the logic XOR for 2 generated signals.
XOR 02	Offline	Calculates the logic XOR for 2 offline signals.

2.5.11 Miscellaneous

2.5.11.1 Cnt() - Count()

2.5.11.1.1 General

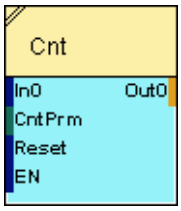
Properties

Property	Description
Full Name	Count
Short Name	Cnt
Description	Counts the rising or falling edges of the input data or values.
Parent Library	Base
Parent Category	Miscellaneous
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Cnt
(
    DataOrValue In0 = false,
    Value       CntPrm = "Rising Edge",
    DataOrValue Reset = false,
    DataOrValue EN = true
);
```

Symbol



2.5.11.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose rising or falling edges shall be counted
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

CntPrm

Property	Description
Full Name	CountParameter
Short Name	CntPrm
Description	indicates whether rising or falling edges shall be counted
Direction	input
Class	parameter
Default Value	“Rising Edge”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Rising Edge”, “Falling Edge”
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the edges counting
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.11.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the edges counting
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.11.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
Cnt 01	Online	Counts the rising and falling edges of a generated signal.
Cnt 02	Offline	Counts the rising and falling edges of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Cnt 01	Online	Counts the rising and falling edges of a generated signal.
Cnt 02	Offline	Counts the rising and falling edges of an offline signal.

2.5.11.2 DeMux() - Demultiplexer()

2.5.11.2.1 General

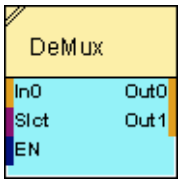
Properties

Property	Description
Full Name	Demultiplexer
Short Name	DeMux
Description	Provides input signal to the selected output.
Parent Library	Base
Parent Category	Miscellaneous
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = DeMux
(
    DataOrValue  In0 = 0,
    DataOrValue  Slct = 1,
    DataOrValue  EN = true,
    DataOrValue* Out1 ... Out7 = NULL
);
```

Symbol



2.5.11.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be provided to the selected output
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Slct

Property	Description
Full Name	Select
Short Name	Slct
Description	identity number of the output which shall be connected to the input signal
Direction	input
Class	control
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … 8
Remark	in case Select = 0 the input signal will be connected to not any outputs

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.11.2.3 Outputs

Out0 … Out7

Property	Description
Full Name	Output0 … Output7
Short Name	Out0 … Out7
Description	receives the input signal to the selected output
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	8
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Out0_7

Property	Description
Full Name	Output0_7
Short Name	Out0_7
Description	determines how many of the output arguments Out0 … Out7 are used by the de-multiplexing
Direction	output
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	none

2.5.11.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
DeMux 01	Online	Uses DeMux to provide a generated signal to different output signals.
DeMux 02	Offline	Uses DeMux to provide an offline signal to a selected output signals.

Analyzing Script

The usage within Analyzing Scripts is not supported by the Analyzing Function **DeMux()**.

2.5.11.3 IsFlgEdg() - IsFallingEdge()

2.5.11.3.1 General

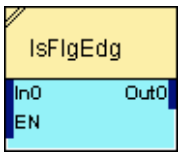
Properties

Property	Description
Full Name	IsFallingEdge
Short Name	IsFlgEdg
Description	Detects the falling edges of the input data or values.
Parent Library	Base
Parent Category	Miscellaneous
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = IsFlgEdg
(
    DataOrValue In0 = false,
    DataOrValue EN = true
);
```

Symbol



2.5.11.3.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose falling edges shall be detected
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.11.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the falling edges detection
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.11.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
IsFlgEdg 01	Online	Detects falling edges of a generated rectangular signal.
IsFlgEdg 02	Offline	Detects falling edges of an offline signal.

Analyzing Script

Name	Output Data Class	Description
IsFlgEdg 01	Online	Detects falling edges of a generated rectangular signal.
IsFlgEdg 02	Offline	Detects falling edges of an offline signal.

2.5.11.4 IsRsgEdg() - IsRisingEdge()

2.5.11.4.1 General

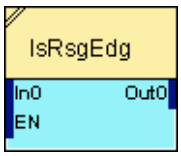
Properties

Property	Description
Full Name	IsRisingEdge
Short Name	IsRsgEdg
Description	Detects the rising edges of the input data or values.
Parent Library	Base
Parent Category	Miscellaneous
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = IsRsgEdg
(
    DataOrValue In0 = false,
    DataOrValue EN = true
);
```

Symbol



2.5.11.4.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose rising edges shall be detected
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.11.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the rising edges detection
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.11.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
IsFlgEdg 01	Online	Detects rising edges of a generated rectangular signal.
IsFlgEdg 02	Offline	Detects rising edges of an offline signal.

Analyzing Script

Name	Output Data Class	Description
IsFlgEdg 01	Online	Detects rising edges of a generated rectangular signal.
IsFlgEdg 02	Offline	Detects rising edges of an offline signal.

2.5.11.5 MonEdg() - MonitorEdge()

2.5.11.5.1 General

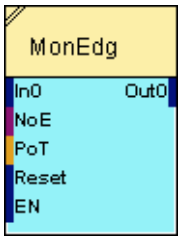
Properties

Property	Description
Full Name	MonitorEdge
Short Name	MonEdg
Description	Detects the occurrences of a certain number of edges in a defined time period.
Parent Library	Base
Parent Category	Miscellaneous
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = MonEdg
(
    DataOrValue In0 = false,
    Value       NoE = 1,
    Value       PoT = 1,
    DataOrValue Reset = false,
    DataOrValue EN = true
);
```

Symbol



2.5.11.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose edges shall be monitored
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

NoE

Property	Description
Full Name	NumberOfEdges
Short Name	NoE
Description	number of edges for occurrences monitoring
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 20
Remark	none

PoT

Property	Description
Full Name	PeriodOfTime
Short Name	PoT
Description	acquisition time of data for the edges monitoring
Direction	input
Class	parameter
Default Value	1
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	(0 … 120
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the edges monitoring
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.11.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the edges monitoring
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.11.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
MonEdg 01	Online	Monitors rising edges of a simulated signal. (Start IPF and DPF)
MonEdg 02	Offline	Monitors rising edges of an offline signal.

Analyzing Script

Name	Output Data Class	Description
MonEdg 01	Online	Monitors rising edges of a simulated signal. (Start IPF and DPF)
MonEdg 02	Offline	Monitors rising edges of an offline signal.

2.5.11.6 Mux() - Multiplexer()

2.5.11.6.1 General

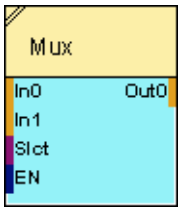
Properties

Property	Description
Full Name	Multiplexer
Short Name	Mux
Description	Provides a selected input signal to the output.
Parent Library	Base
Parent Category	Miscellaneous
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Mux
(
    DataOrValue In0 ... In7 = 0;
    DataOrValue Slct = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.11.6.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	data or values which shall be multiplexed
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	8
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 … In7 are used by the multiplexing
Direction	input
Class	configuration
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	2 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

Slct

Property	Description
Full Name	Select
Short Name	Slct
Description	identity number of the input signal which shall be provided to the output
Direction	input
Class	control
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … 7
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.11.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of multiplexing
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.11.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
Mux 01	Online	Uses Mux to provide different generated signals to the output.
Mux 02	Offline	Uses Mux to provide different offline signals to the output.

Analyzing Script

Name	Output Data Class	Description
Mux 01	Online	Uses Mux to provide different generated signals to the output.
Mux 02	Offline	Uses Mux to provide different offline signals to the output.

2.5.12 Orbit

2.5.12.1 Orbit() - Orbit()

2.5.12.1.1 General

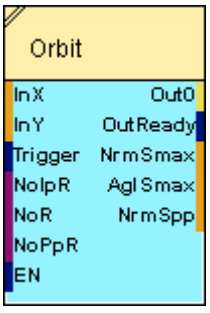
Properties

Property	Description
Full Name	Orbit
Short Name	Orbit
Description	Calculates the orbit data of the input data.
Parent Library	Advanced
Parent Category	Orbit
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice based

Function Prototype

```
Orbit Out0 = Orbit
(
    DataOrValue  InX = 0,
    DataOrValue  InY = 0,
    DataOrValue  Trigger = 0,
    Value        NoIpR = 1,
    Value        NoR = 100,
    Value        NoPpR = 64,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL,
    DataOrValue* NrmSmax = 0,
    DataOrValue* AglSmax = 0,
    DataOrValue* NrmSpp = 0
);
```

Symbol



2.5.12.1.2 Inputs

InX

Property	Description
Full Name	InputX
Short Name	InX
Description	data or value which shall be used as x-data for the orbit calculation
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

InY

Property	Description
Full Name	InputX
Short Name	InY
Description	data or value which shall be used as y-data for the orbit calculation
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Trigger

Property	Description
Full Name	Trigger
Short Name	Trigger
Description	trigger data which provides information about the number of revolutions
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoIpR

Property	Description
Full Name	NumberOfImpulsesPerRevolution
Short Name	NoIpR
Description	specifies the number of trigger impulses per revolution
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer 32
Allowed manual Inputs	1 … 8
Remark	none

NoR

Property	Description
Full Name	NumberOfRevolution
Short Name	NoR
Description	specifies the number of revolutions which shall be collected for one calculation
Direction	input
Class	parameter
Default Value	100
Unit	1
Configurable	1
Supported Data Types	Integer 32
Allowed manual Inputs	1 … 10000
Remark	none

NoPpR

Property	Description
Full Name	NumberOfPointsPerRevolution
Short Name	NoPpR
Description	specifies the number of points per revolution to which the collected input data shall be resampled
Direction	input
Class	parameter
Default Value	64
Unit	1
Configurable	1
Supported Data Types	Integer 32
Allowed manual Inputs	8 … 16384
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	if the internal calculation is disabled, the already received input buffer is discarded and the filling of the internal buffer is restarted when the internal calculation is enabled again

2.5.12.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the orbit calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Orbit
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the orbit data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

NrmSmax

Property	Description
Full Name	NormSmax
Short Name	NrmSmax
Description	receives the calculated norm of Smax
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

AglSmax

Property	Description
Full Name	AngelSmax
Short Name	AglSmax
Description	receives the calculated angle of Smax
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

NrmSpp

Property	Description
Full Name	NormSpp
Short Name	NrmSpp
Description	receives the calculated norm of Spp
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.12.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
Orbit 01	Online	Calculates the orbit of some simulated input data.
Orbit 02	Offline	Calculates the orbit value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Orbit 01	Online	Calculates the orbit of some simulated input data.
Orbit 02	Offline	Calculates the orbit value of an offline signal.

2.5.13 Output

2.5.13.1 Out() - Output()

2.5.13.1.1 General

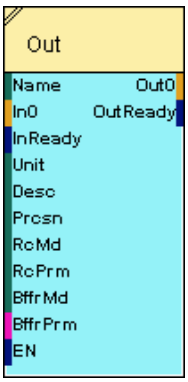
Properties

Property	Description
Full Name	Output
Short Name	Out
Description	Writes simple data to the system, which has been calculated by previous Analyzing Functions.
Parent Library	Base
Parent Category	Output
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Out
(
    Value         Name = "",
    DataOrValue   In0 = 0,
    DataOrValue   InReady = true,
    Value         Unit = "",
    Value         Desc = "",
    Value         Prcsn = "Default",
    Value         RcMd = "Default",
    Value         RcPrm = "",
    Value         BffrMd = "Relative Size",
    Value         BffrPrm = "100",
    DataOrValue   EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.13.1.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the data or values which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the simple data from previous Analyzing Function
Remark	none

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	simple data or values which shall be written to the system
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of the chosen precision which is chosen via Precision argument
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the data in Input0 is valid
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	description of the data or values
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the description of the simple data which shall be written to the system
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	unit of the data or values
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the unit of the simple data which shall be written to the system
Remark	none

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the data or values
Direction	input
Class	parameter
Default Value	precision of the parent Analyzing Model/Analyzing Script
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Boolean”, “Integer8”, “Integer16”, “Integer32”, “Integer64”, “Decimal32”, “Decimal64”
Remark	none

RcMd

Property	Description
Full Name	RecordMode
Short Name	RcMd
Description	record mode which shall be used when writing the data or values to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Each Value”, “Changes only”, “Changes and Interval”, “Interval only”
Remark	none

RcPrm

Property	Description
Full Name	RecordParameters
Short Name	RcPrm
Description	record parameters which shall be used when writing the data or values to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	standard input behavior of record parameters edit boxes
Remark	none

BffrMd

Property	Description
Full Name	BufferMode
Short Name	BffrMd
Description	buffer mode which shall be used when writing the data or values to the system
Direction	input
Class	parameter
Default Value	“Relative Size”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Absolute Size”, “Relative Size”
Remark	none

BffrPrm

Property	Description
Full Name	BufferParameters
Short Name	BffrPrm
Description	buffer parameters which shall be used when writing the data or values to the system
Direction	input
Class	parameter
Default Value	“100”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	standard input behavior of buffer parameters edit boxes
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.13.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the data or values value which has been written to the system
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	equal to the supported precisions of the Analyzing Function
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the data in Output0 and Output1 are valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.13.2 OutB1D() - OutputBuffer1D()

2.5.13.2.1 General

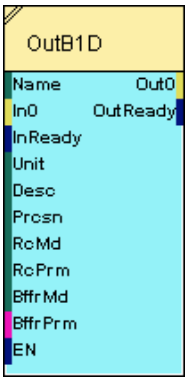
Properties

Property	Description
Full Name	OutputBuffer1D
Short Name	OutB1D
Description	Writes 1-dimensional buffer data to the system, which has been calculated by previous Analyzing Functions.
Parent Library	Base
Parent Category	Output
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = OutB1D
(
    Value         Name = "",
    Buffer1D      In0 = 0,
    DataOrValue   InReady = false,
    Value         Unit = "",
    Value         Desc = "",
    Value         Prcsn = "Default",
    Value         RcMd = "Default",
    Value         RcPrm = "",
    Value         BffrMd = "Absolute Size",
    Value         BffrPrm = "50",
    DataOrValue   EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.13.2.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the buffer data from previous Analyzing Function
Remark	in case a data name prefix is being specified for output data of the parent Analyzing Model/Analyzing Script, the specified output data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	1-dimensional buffer data whose values shall be written to the system
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	unit of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the unit of the buffer data which shall be written to the system
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	description of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the description of the buffer data which shall be written to the system
Remark	none

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Boolean”, “Integer8”, “Integer16”, “Integer32”, “Integer64”, “Decimal32”, “Decimal64”
Remark	in case the argument is set to "Default" the currently "Default Precision" of the owning Analyzing Model/Analyzing Script is being chosen as data precision

RcMd

Property	Description
Full Name	RecordMode
Short Name	RcMd
Description	record mode which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Each Value”, “Interval only”
Remark	none

RcPrm

Property	Description
Full Name	RecordParameters
Short Name	RcPrm
Description	record parameters which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	standard input behavior of record parameters edit boxes
Remark	none

BffrMd

Property	Description
Full Name	BufferMode
Short Name	BffrMd
Description	buffer mode which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“Absolute Size”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Absolute Size”, “Relative Size”
Remark	none

BffrPrm

Property	Description
Full Name	BufferParameters
Short Name	BffrPrm
Description	buffer parameters which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“50”
Unit	n/a
Configurable	1
Supported Data Types	Integer64
Allowed manual Inputs	standard input behavior of buffer parameters edit boxes
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.13.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 1-dimensional buffer data which has been written to the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.13.3 OutB2D() – OutputBuffer2D()

2.5.13.3.1 General

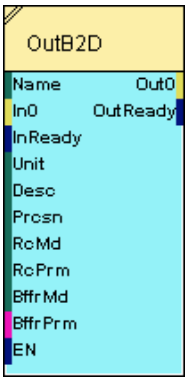
Properties

Property	Description
Full Name	OutputBuffer2D
Short Name	OutB2D
Description	Writes 2-dimensional buffer data to the system, which has been calculated by previous Analyzing Functions.
Parent Library	Base
Parent Category	Output
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer2D Out0 = OutB2D
(
    Value         Name = "",
    Buffer2D      In0 = 0,
    DataOrValue   InReady = false,
    Value         Unit = "",
    Value         Desc = "",
    Value         Prcsn = "Default",
    Value         RcMd = "Default",
    Value         RcPrm = "",
    Value         BffrMd = "Absolute Size",
    Value         BffrPrm = "50",
    DataOrValue   EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.13.3.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the buffer data from previous Analyzing Function
Remark	in case a data name prefix is being specified for output data of the parent Analyzing Model/Analyzing Script, the specified output data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	2-dimensional buffer data whose values shall be written to the system
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer2D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	unit of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the unit of the buffer data which shall be written to the system
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	description of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the description of the buffer data which shall be written to the system
Remark	none

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the buffer data which shall be written to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Boolean”, “Integer8”, “Integer16”, “Integer32”, “Integer64”, “Decimal32”, “Decimal64”
Remark	in case the argument is set to "Default" the currently "Default Precision" of the owning Analyzing Model/Analyzing Script is being chosen as data precision

RcMd

Property	Description
Full Name	RecordMode
Short Name	RcMd
Description	record mode which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Each Value”, “Interval only”
Remark	in case the argument is set to “Default” but the currently chosen “Default Output Data Record Mode” of the owning Analyzing Model/Analyzing Script is not supported by the ANF, “Each Value” is being chosen as record mode

RcPrm

Property	Description
Full Name	RecordParameters
Short Name	RcPrm
Description	record parameters which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	standard input behavior of record parameters edit boxes
Remark	none

BffrMd

Property	Description
Full Name	BufferMode
Short Name	BffrMd
Description	buffer mode which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“Absolute Size”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Absolute Size”, “Relative Size”
Remark	none

BffrPrm

Property	Description
Full Name	BufferParameters
Short Name	BffrPrm
Description	buffer parameters which shall be used when writing the buffer data to the system
Direction	input
Class	parameter
Default Value	“50”
Unit	1
Configurable	1
Supported Data Types	Integer64
Allowed manual Inputs	standard input behavior of buffer parameters edit boxes
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.13.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 2-dimensional buffer data which has been written to the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer2D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.13.4 OutOrbit() - OutputOrbit()

2.5.13.4.1 General

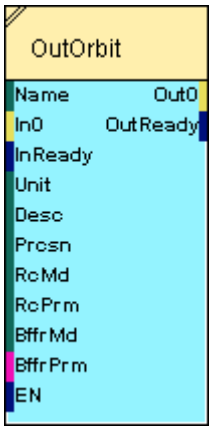
Properties

Property	Description
Full Name	OutputOrbit
Short Name	OutOrbit
Description	Writes orbit data to the system, which has been calculated by previous Analyzing Functions.
Parent Library	Advanced
Parent Category	Output
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Orbit Out0 = OutOrbit
(
    Value           Name = "",
    Orbit           In0 = 0,
    DataOrValue     InReady = false,
    Value           Unit = "",
    Value           Desc = "",
    Value           Prcsn = "Default",
    Value           RcMd = "Default",
    Value           RcPrm = "",
    Value           BffrMd = "Absolute Size",
    Value           BffrPrm = "50",
    DataOrValue     EN = true,
    DataOrValue*    OutReady = NULL
);
```

Symbol



2.5.13.4.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the orbit data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the spectrum data from previous Analyzing Function
Remark	in case a data name prefix is being specified for output data of the parent Analyzing Model/Analyzing Script, the specified output data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	orbit data whose values shall be written to the system
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Orbit
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the orbit data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	unit of the orbit data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the unit of the orbit data which shall be written to the system
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	description of the orbit data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the description of the orbit data which shall be written to the system
Remark	none

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the orbit data which shall be written to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Decimal32”, “Decimal64”
Remark	in case the argument is set to "Default" the currently "Default Precision" of the owning Analyzing Model/Analyzing Script is being chosen as data precision

RcMd

Property	Description
Full Name	RecordMode
Short Name	RcMd
Description	record mode which shall be used when writing the orbit data to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Each Value”, “Interval only”
Remark	in case the argument is set to “Default” but the currently chosen “Default Output Data Record Mode” of the owning Analyzing Model/Analyzing Script is not supported by the ANF, “Each Value” is being chosen as record mode

RcPrm

Property	Description
Full Name	RecordParameters
Short Name	RcPrm
Description	record parameters which shall be used when writing the orbit data to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	standard input behavior of record parameters edit boxes
Remark	none

BffrMd

Property	Description
Full Name	BufferMode
Short Name	BffrMd
Description	buffer mode which shall be used when writing the orbit data to the system
Direction	input
Class	parameter
Default Value	“Absolute Size”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Absolute Size”, “Relative Size”
Remark	none

BffrPrm

Property	Description
Full Name	BufferParameters
Short Name	BffrPrm
Description	buffer parameters which shall be used when writing the orbit data to the system
Direction	input
Class	parameter
Default Value	“50”
Unit	n/a
Configurable	1
Supported Data Types	Integer64
Allowed manual Inputs	standard input behavior of buffer parameters edit boxes
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.13.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the orbit data which has been written to the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Orbit
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the orbit data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.13.5 OutSpm() - OutputSpectrum()

2.5.13.5.1 General

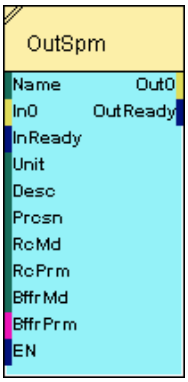
Properties

Property	Description
Full Name	OutputSpectrum
Short Name	OutSpm
Description	Writes spectrum data to the system, which has been calculated by previous Analyzing Functions.
Parent Library	Base
Parent Category	Output
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Spectrum Out0 = OutSpm
(
    Value              Name = "",
    Spectrum           In0 = 0,
    DataOrValue        InReady = false,
    Value              Unit = "",
    Value              Desc = "",
    Value              Prcsn = "Default",
    Value              RcMd = "Default",
    Value              RcPrm = "",
    Value              BffrMd = "Absolute Size",
    Value              BffrPrm = "50",
    DataOrValue        EN = true,
    DataOrValue*       OutReady = NULL
);
```

Symbol



2.5.13.5.2 Inputs

Name

Property	Description
Full Name	Name
Short Name	Name
Description	name of the spectrum data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the name of the spectrum data from previous Analyzing Function
Remark	in case a data name prefix is being specified for output data of the parent Analyzing Model/Analyzing Script, the specified output data name prefix is being added to the specified name before the execution of the Analyzing Model/Analyzing Script

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	spectrum data whose values shall be written to the system
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the spectrum data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Unit

Property	Description
Full Name	Unit
Short Name	Unit
Description	unit of the spectrum data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the unit of the spectrum data which shall be written to the system
Remark	none

Desc

Property	Description
Full Name	Description
Short Name	Desc
Description	description of the spectrum data which shall be written to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the description of the spectrum data which shall be written to the system
Remark	none

Prcsn

Property	Description
Full Name	Precision
Short Name	Prcsn
Description	precision of the spectrum data which shall be written to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Boolean”, “Integer8”, “Integer16”, “Integer32”, “Integer64”, “Decimal32”, “Decimal64”
Remark	in case the argument is set to "Default" the currently "Default Precision" of the owning Analyzing Model/Analyzing Script is being chosen as data precision

RcMd

Property	Description
Full Name	RecordMode
Short Name	RcMd
Description	record mode which shall be used when writing the spectrum data to the system
Direction	input
Class	parameter
Default Value	“Default”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Default”, “Each Value”, “Interval only”
Remark	in case the argument is set to “Default” but the currently chosen “Default Output Data Record Mode” of the owning Analyzing Model/Analyzing Script is not supported by the ANF, “Each Value” is being chosen as record mode

RcPrm

Property	Description
Full Name	RecordParameters
Short Name	RcPrm
Description	record parameters which shall be used when writing the spectrum data to the system
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	standard input behavior of record parameters edit boxes
Remark	none

BffrMd

Property	Description
Full Name	BufferMode
Short Name	BffrMd
Description	buffer mode which shall be used when writing the spectrum data to the system
Direction	input
Class	parameter
Default Value	“Absolute Size”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Absolute Size”, “Relative Size”
Remark	none

BffrPrm

Property	Description
Full Name	BufferParameters
Short Name	BffrPrm
Description	buffer parameters which shall be used when writing the spectrum data to the system
Direction	input
Class	parameter
Default Value	“50”
Unit	n/a
Configurable	1
Supported Data Types	Integer64
Allowed manual Inputs	standard input behavior of buffer parameters edit boxes
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.13.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the spectrum data which has been written to the system
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the spectrum data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.13.6 OutTCP() - OutputTCP()

2.5.13.6.1 General

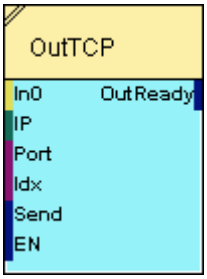
Properties

Property	Description
Full Name	OutputTCP
Short Name	OutTCP
Description	Sends a data buffer to the specified IP address.
Parent Library	Advanced
Parent Category	Output
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue OutReady = OutTCP
(
    Buffer1D    In0 ... In7 = 0,
    Value       IP = "",
    Value       Port = 1,
    Value       Idx = 4,
    DataOrValue Send = false,
    DataOrValue EN = true
);
```

Symbol



2.5.13.6.2 Inputs

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	1-dimensional buffer which shall be sent
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	8
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments In0 ... In7 are used by the sending
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 8
Remark	none

IP

Property	Description
Full Name	IPAddress
Short Name	IP
Description	receives the IP address for sending the buffer data
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the IP address
Remark	none

Port

Property	Description
Full Name	Port
Short Name	Port
Description	receives the port for sending the buffer data
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 65535
Remark	none

Idx

Property	Description
Full Name	Index
Short Name	Idx
Description	receives the index for sending the buffer data
Direction	input
Class	parameter
Default Value	4
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Integer32
Remark	see Examples

Send

Property	Description
Full Name	Send
Short Name	Send
Description	indicates whether the current buffer data shall be sent
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal function
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.13.6.3 Outputs

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the input buffer are sent to the IP address
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.13.6.4 Remarks

In order to use this **Analyzing Function**, a TCP server must be provided for the other side of the connection. The IP address and port which are being used by the TCP server must be specified for the “IP” and “Port” input arguments of this **Analyzing Function**.

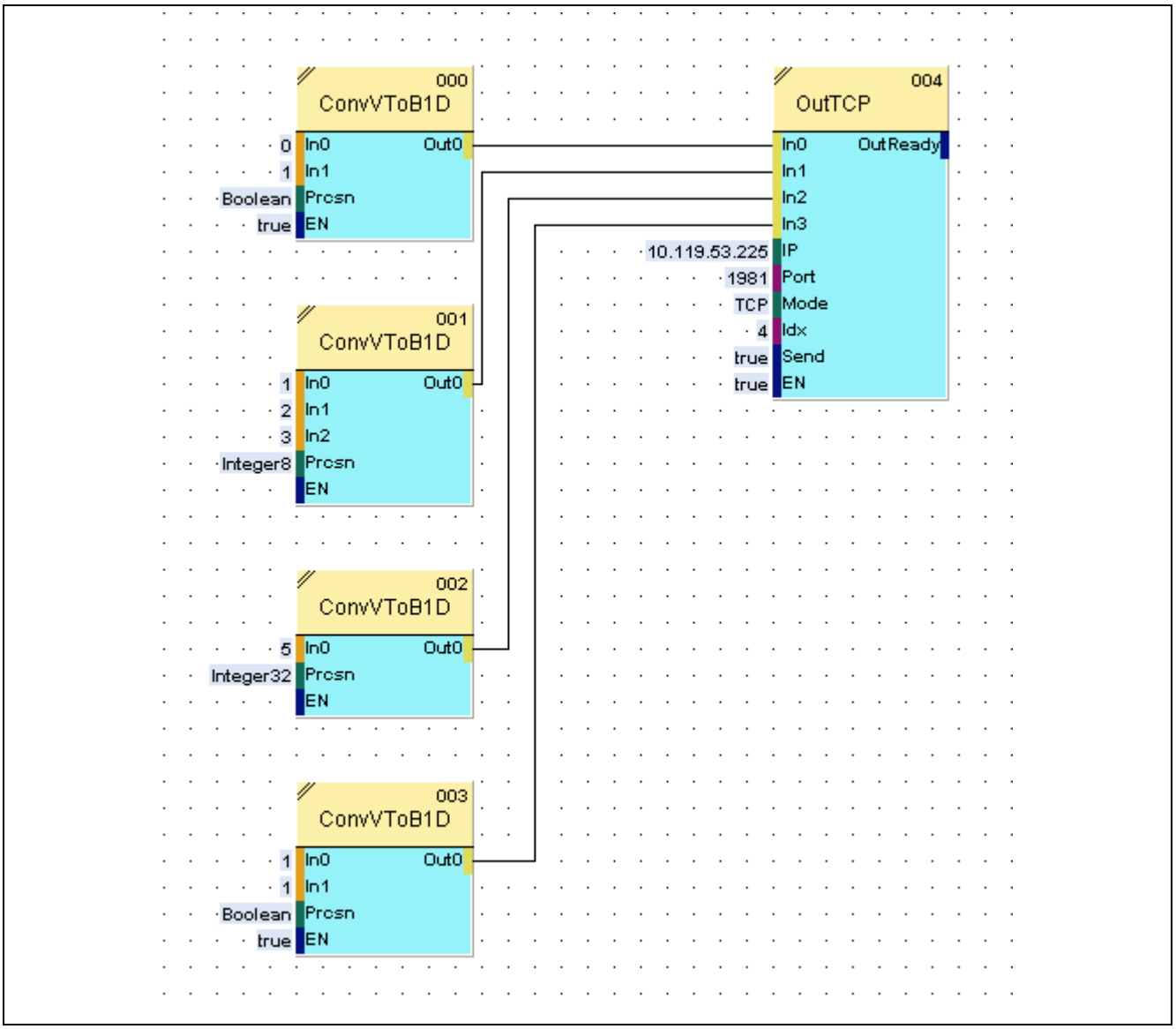
Besides the standard TCP framework bytes, the following data is being transmitted by this **Analyzing Function**:

Length	Type	Data Type	Description
4 Byte	Timestamp	unsigned long	contains the current micro seconds (0 … 3600000000), relative to the time of the executing Analyzing Model/Analyzing Script
4 Byte	Length	unsigned long	contains the length of the payload (without the “Timestamp”, “Length”, “Index” and “End Delimiter” fields), which is calculated automatically according to the provided inputs (“Input0” … “Input7”) the length includes also includes possible present padding bytes
4 Byte	Index	unsigned long	contains the index which is defined at input argument “Index”
x Byte	Payload	variable	contains the payload which is provided through “Input0” … “Input7”
4 Byte	End Delimiter	unsigned long	contains the end delimiter (0x7FFFFFFF)

Data Value Assignment

Precision	Length per Value	Remarks
Boolean	1 Byte	one byte is being used for each bit (the **Analyzing Function** does not put multiple bits together to one byte) in case the number of Boolean values of an input is not a multiple of 4, padding bytes (with a value of 0) are being inserted after the Boolean values of the input
Integer8	1 Byte	in case the number of Integer8 values of an input is not a multiple of 4, padding bytes (with a value of 0) are being inserted after the Integer8 values of the input
Integer16	2 Byte	in case the number of Integer16 values of an input is not a multiple of 4, padding bytes (with a value of 0) are being inserted after the Integer16 values of the input
Integer32	4 Byte	
Integer64	8 Byte	
Decimal32	4 Byte	
Decimal64	8 Byte	

Example


000
ConvVToB1D
0
In0
Out0
1
In1
Boolean
Prcsn
true
EN
001
ConvVToB1D
1
In0
Out0
2
In1
3
In2
Integer8
Prcsn
EN
002
ConvVToB1D
5
In0
Out0
Integer32
Prcsn
EN
003
ConvVToB1D
1
In0
Out0
1
In1
Boolean
Prcsn
true
EN
004
OutTCP
In0
OutReady
In1
In2
In3
10.119.53.225
IP
1981
Port
TCP
Mode
4
Idx
true
Send
true
EN

Data Segment

Byte	Value (Hex)	Description
01 … 04	…	timestamp
05 … 08	0x00000010	length
09 … 12	0x00000004	index
13	0x00	first value of ConvVToB1D (ID 000) of precision Boolean
14	0x01	second value of ConvVToB1D (ID 000) of precision Boolean
15	0x00	filled up with 0, because the number of Booleans is not a multiple of 4
16	0x00	filled up with 0, because the number of Booleans is not a multiple of 4
17	0x01	first value of ConvVToB1D (ID 001) of precision Integer8
18	0x02	second value of ConvVToB1D (ID 001) of precision Integer8
19	0x03	third value of ConvVToB1D (ID 001) of precision Integer8
20	0x00	filled up with 0, because the number of Integer8s is not a multiple of 4
21 … 24	0x00000005	first value of ConvVToB1D (ID 002) of precision Integer32
25	0x01	first value of ConvVToB1D (ID 003) of precision Boolean
26	0x01	second value of ConvVToB1D (ID 003) of precision Boolean
27	0x00	filled up with 0, because the number of Booleans is not a multiple of 4
28	0x00	filled up with 0, because the number of Booleans is not a multiple of 4
29 … 32	0x7FFFFFFF	end delimiter

2.5.13.6.5 Examples

Analyzing Model

Name	Output Data Class	Description
OutTCP 01	Online	Different online data is sent via TCP (server needed).
OutTCP 02	Offline	Different offline data is sent via TCP (server needed).

Analyzing Script

Name	Output Data Class	Description
OutTCP 01	Online	Different online data is sent via TCP (server needed).
OutTCP 02	Offline	Different offline data is sent via TCP (server needed).

2.5.14 Spectral

2.5.14.1 AppBhann() - ApplyBartlettHann()

2.5.14.1.1 General

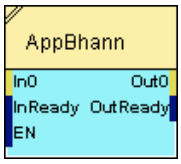
Properties

Property	Description
Full Name	ApplyBartlettHann
Short Name	AppBhann
Description	Applies Bartlett Hanning window to the input buffer.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = AppBhann
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the windowing
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer with the result of the windowing data calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
AppBhann 01	Online	Applies a Bartlett Hanning window to a buffered sine wave.
AppBhann 02	Offline	Applies a Bartlett Hanning window to a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
AppBhann 01	Online	Applies a Bartlett Hanning window to a buffered sine wave.
AppBhann 02	Offline	Applies a Bartlett Hanning window to a buffered offline signal.

2.5.14.2 AppBlack() - ApplyBlackmann()

2.5.14.2.1 General

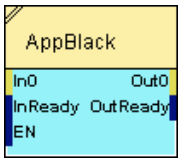
Properties

Property	Description
Full Name	ApplyBlackmann
Short Name	AppBlack
Description	Applies Blackmann window to the input buffer.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = AppBlack
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the windowing
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer with the result of the windowing data calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
AppBlack 01	Online	Applies a Blackman Hanning window to a buffered sine wave.
AppBlack 02	Offline	Applies a Blackman Hanning window to a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
AppBlack 01	Online	Applies a Blackman Hanning window to a buffered sine wave.
AppBlack 02	Offline	Applies a Blackman Hanning window to a buffered offline signal.

2.5.14.3 AppBlett() - ApplyBartlett()

2.5.14.3.1 General

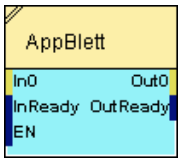
Properties

Property	Description
Full Name	ApplyBartlett
Short Name	AppBlett
Description	Applies Bartlett window to the input buffer.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = AppBlett
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.3.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the windowing
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer with the result of the windowing data calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
AppBlett 01	Online	Applies a Bartlett Hanning window to a buffered sine wave.
AppBlett 02	Offline	Applies a Bartlett Hanning window to a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
AppBlett 01	Online	Applies a Bartlett Hanning window to a buffered sine wave.
AppBlett 02	Offline	Applies a Bartlett Hanning window to a buffered offline signal.

2.5.14.4 AppHamm() - ApplyHamming()

2.5.14.4.1 General

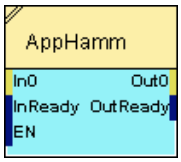
Properties

Property	Description
Full Name	ApplyHamming
Short Name	AppHamm
Description	Applies Hamming window to the input buffer.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = AppHamm
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.4.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the windowing
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer with the result of the windowing data calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
AppHamm 01	Online	Applies a Hamming window to a buffered sine wave.
AppHamm 02	Offline	Applies a Hamming window to a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
AppHamm 01	Online	Applies a Hamming window to a buffered sine wave.
AppHamm 02	Offline	Applies a Hamming window to a buffered offline signal.

2.5.14.5 AppHann() - ApplyHanning()

2.5.14.5.1 General

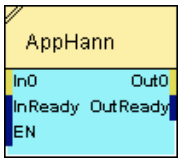
Properties

Property	Description
Full Name	ApplyHanning
Short Name	AppHann
Description	Applies Hanning window to the input buffer.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = AppHann
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values for the windowing
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer with the result of the windowing data calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
AppHann 01	Online	Applies a Hanning window to a buffered sine wave.
AppHann 02	Offline	Applies a Hanning window to a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
AppHann 01	Online	Applies a Hanning window to a buffered sine wave.
AppHann 02	Offline	Applies a Hanning window to a buffered offline signal.

2.5.14.6 AvgSpm() - AverageSpectrum()

2.5.14.6.1 General

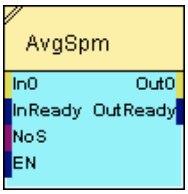
Properties

Property	Description
Full Name	AverageSpectrum
Short Name	AvgSpm
Description	Averages two or more input frequency spectrums.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
Spectrum Out0 = AvgSpm
(
    Spectrum          In0 = 0,
    DataOrValue       InReady = false,
    Value             NoS = 2,
    DataOrValue       EN = true,
    DataOrValue*      OutReady = NULL
);
```

Symbol



2.5.14.6.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	frequency spectrums which shall be averaged
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the spectrum data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

NoS

Property	Description
Full Name	NumberOfSpectrums
Short Name	NoS
Description	determines number of spectrums which shall be used by the averaging
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the averaged frequency spectrum
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the spectrum data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
AvgSpm 01	Online	Calculates the average spectrum of spectra of a generated sine wave with changing frequency.
AvgSpm 02	Offline	Calculates the average spectrum of spectra of an offline signal.

Analyzing Script

Name	Output Data Class	Description
AvgSpm 01	Online	Calculates the average spectrum of spectra of a generated sine wave with changing frequency.
AvgSpm 02	Offline	Calculates the average spectrum of spectra of an offline signal.

2.5.14.7 Cpstr() - Cepstrum()

2.5.14.7.1 General

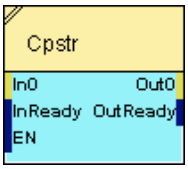
Properties

Property	Description
Full Name	Cepstrum
Short Name	Cpstr
Description	Calculates the cepstrum of the input buffer.
Parent Library	Advanced
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = Cpstr
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.7.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values whose cepstrum shall be calculated
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.7.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the cepstrum calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.7.4 Examples

Analyzing Model

Name	Output Data Class	Description
Cpstr 01	Online	Calculates cepstrum for a generated signal.
Cpstr 02	Offline	Calculates cepstrum for an offline signal.

Analyzing Script

Name	Output Data Class	Description
Cpstr 01	Online	Calculates cepstrum for a generated signal.
Cpstr 02	Offline	Calculates cepstrum for an offline signal.

2.5.14.8 EnvHlb() - EnvelopeHilbert()

2.5.14.8.1 General

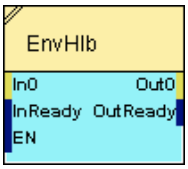
Properties

Property	Description
Full Name	EnvelopeHilbert
Short Name	EnvHlb
Description	Calculates the envelope of the input data or values by using the Hilbert transformation.
Parent Library	Advanced
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = EnvHlb
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.8.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values whose envelope shall be calculated
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.8.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the envelope calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.8.4 Examples

Analyzing Model

Name	Output Data Class	Description
EnvHlb 01	Online	Calculates envelope for a generated signal by using the Hilbert transformation.
EnvHlb 02	Offline	Calculates envelope for an offline signal by using the Hilbert transformation.

Analyzing Script

Name	Output Data Class	Description
EnvHlb 01	Online	Calculates envelope for a generated signal by using the Hilbert transformation.
EnvHlb 02	Offline	Calculates envelope for an offline signal by using the Hilbert transformation.

2.5.14.9 FFT() - FastFourierTransformation()

2.5.14.9.1 General

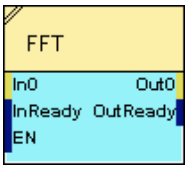
Properties

Property	Description
Full Name	FastFourierTransformation
Short Name	FFT
Description	Calculates the spectrum of the input buffer by using the fast Fourier transformation.
Parent Library	Base
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Spectrum Out0 = FFT
(
    Buffer1D     In0 = 0,
    DataOrValue  InReady = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.14.9.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values whose spectrum shall be calculated
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.9.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the spectrum calculation
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the spectrum data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.14.9.4 Examples

Analyzing Model

Name	Output Data Class	Description
FFT 01	Online	Calculates the spectrum of a generated sine wave by using Fast Fourier Transformation.
FFT 02	Offline	Calculates the spectrum of an offline signal by using Fast Fourier Transformation.
FFT 03	Online	Calculates the spectrum of 2 generated sine waves by using Fast Fourier Transformation.

Analyzing Script

Name	Output Data Class	Description
FFT 01	Online	Calculates the spectrum of a generated sine wave by using Fast Fourier Transformation.
FFT 02	Offline	Calculates the spectrum of an offline signal by using Fast Fourier Transformation.
FFT 03	Online	Calculates the spectrum of 2 generated sine waves by using Fast Fourier Transformation.

2.5.14.10 GenAlrmBnd() - GenerateAlarmBand()

2.5.14.10.1 General

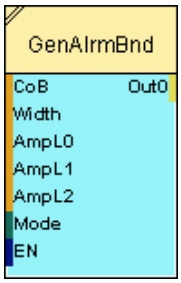
Properties

Property	Description
Full Name	GenerateAlarmBand
Short Name	GenAlrmBnd
Description	Generates an alarm band.
Parent Library	Advanced
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = GenAlrmBnd
(
    DataOrValue CoB = 0,
    DataOrValue Width = 0,
    DataOrValue AmplL0 = 0,
    DataOrValue AmplL1 = 0,
    DataOrValue AmplL2 = 0,
    Value       Mode = "Peak",
    DataOrValue EN = true
);
```

Symbol



2.5.14.10.2 Inputs

CoB

Property	Description
Full Name	CenterOfBand
Short Name	CoB
Description	center of the alarm band which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1 or Hz
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	the unit of CenterOfBand is dependent on the following used Analyzing Function and is “Hz” for the frequency analyzing or “1” for the order analyzing of the alarm spectrum

Width

Property	Description
Full Name	Width
Short Name	Width
Description	width of the alarm band which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1 or Hz
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	the value of Width must be smaller than or equal to the half of CenterOfBand

AmplL0

Property	Description
Full Name	AmplitudeLevel0
Short Name	AmplL0
Description	amplitude level 0 of the alarm band which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

AmplL1

Property	Description
Full Name	AmplitudeLevel1
Short Name	AmplL1
Description	amplitude level 1 of the alarm band which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

AmplL2

Property	Description
Full Name	AmplitudeLevel2
Short Name	AmplL2
Description	amplitude level 2 of the alarm band which shall be generated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

Mode

Property	Description
Full Name	Mode
Short Name	Mode
Description	mode which is used for calculation within a given band
Direction	input
Class	parameter
Default Value	“Peak”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Peak”, “Power”
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.10.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer data with the amount of information describing the alarm band
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Type	Buffer1D
Allowed manual Inputs	n/a
Remark	none

2.5.14.10.4 Examples

See best practises, chapter “Measurement Tasks” - “Spectrum Monitoring”.

2.5.14.11 MonAlrmSpm() - MonitorAlarmSpectrum()

2.5.14.11.1 General

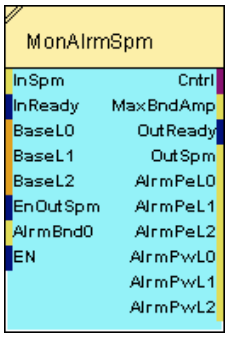
Properties

Property	Description
Full Name	MonitorAlarmSpectrum
Short Name	MonAlrmSpm
Description	Monitors a spectrum with an alarm spectrum.
Parent Library	Advanced
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Cntrl = MonAlrmSpm
(
    Spectrum            InSpm = 0,
    DataOrValue         InReady = false,
    DataOrValue         BaseL0 = 0,
    DataOrValue         BaseL1 = 0,
    DataOrValue         BaseL2 = 0,
    DataOrValue         EnOutSpm = true,
    DataOrValue         AlrmBnd0 ... AlrmBnd15 = 0;
    DataOrValue         EN = true,
    Buffer1D*           MaxBndAmp = NULL,
    DataOrValue*        OutReady = NULL,
    Spectrum*           OutSpm = NULL
    Spectrum*           AlrmPeL0 = NULL,
    Spectrum*           AlrmPeL1 = NULL,
    Spectrum*           AlrmPeL2 = NULL,
    Spectrum*           AlrmPwL0 = NULL,
    Spectrum*           AlrmPwL1 = NULL,
    Spectrum*           AlrmPwL2 = NULL
);
```

Symbol



2.5.14.11.2 Inputs

InSpm

Property	Description
Full Name	InputSpectrum
Short Name	InSpm
Description	spectrum which shall be monitored
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the spectrum data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

BaseL0

Property	Description
Full Name	BaseLevel0
Short Name	BaseL0
Description	base amplitude level 0 of the alarm spectrum
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

BaseL1

Property	Description
Full Name	BaseLevel1
Short Name	BaseL1
Description	base amplitude level 1 of the alarm spectrum
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

BaseL2

Property	Description
Full Name	BaseLevel2
Short Name	BaseL2
Description	base amplitude level 2 of the alarm spectrum
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

EnOutSpm

Property	Description
Full Name	EnableOutputSpectrum
Short Name	EnOutSpm
Description	enables or disables the calculation and generation of the output spectrum
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Precisions	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

AlrmBnd0 … AlrmBnd15

Property	Description
Full Name	AlarmBand0 … AlarmBand15
Short Name	AlrmBnd0 … AlrmBnd15
Description	alarm bands which shall be used for the monitoring
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	16
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	the amplitudes of alarm bands which are attached to inputs with a higher number can overwrite the amplitudes of alarm bands which are attached to inputs with lower numbers - thus, in case two or more alarm bands overlap at any frequency, the amplitude from the alarm band which is connected to the highest input is being used

AlrmBnd0_15

Property	Description
Full Name	AlarmBand0_15
Short Name	AlrmBnd0_15
Description	determines how many of the alarm bands AlarmBand0 … AlarmBand15 are used by the monitoring
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 16
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Precisions	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.11.3 Outputs

Cntrl

Property	Description
Full Name	Control
Short Name	Cntrl
Description	receives the result of the monitoring
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	n/a
Remark	none

MaxBndAmp

Property	Description
Full Name	MaxBandAmplitude
Short Name	MaxBndAmp
Description	receives the buffer data with the maximum amplitude of each alarm band
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Type	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the spectrum data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

OutSpm

Property	Description
Full Name	OutputSpectrum
Short Name	OutSpm
Description	receives the result of the derivated spectrum
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

AlrmPeL0

Property	Description
Full Name	AlarmSpectrumPeakLevel0
Short Name	AlrmPeL0
Description	receives the result of the “Peak” alarm spectrum level 0
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

AlrmPeL1

Property	Description
Full Name	AlarmSpectrumPeakLevel1
Short Name	AlrmPeL1
Description	receives the result of the “Peak” alarm spectrum level 1
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

AlrmPeL2

Property	Description
Full Name	AlarmSpectrumPeakLevel2
Short Name	AlrmPeL2
Description	receives the result of the “Peak” alarm spectrum level 2
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

AlrmPwL0

Property	Description
Full Name	AlarmSpectrumPowerLevel0
Short Name	AlrmPwL0
Description	receives the result of the “Power” alarm spectrum level 0
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

AlrmPwL1

Property	Description
Full Name	AlarmSpectrumPowerLevel1
Short Name	AlrmPwL1
Description	receives the result of the “Power” alarm spectrum level 1
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

AlrmPwL2

Property	Description
Full Name	AlarmSpectrumPowerLevel2
Short Name	AlrmPwL2
Description	receives the result of the “Power” alarm spectrum level 2
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

2.5.14.11.4 Examples

Analyzing Model

Name	Output Data Class	Description
MonAlrmSpm 01	Online	Monitors the spectrum of a superposition of three sine waves.
MonAlrmSpm 02	Offline	Monitors the spectrum of an offline signal.

Analyzing Script

Name	Output Data Class	Description
MonAlrmSpm 01	Online	Monitors the spectrum of a superposition of three sine waves.
MonAlrmSpm 02	Offline	Monitors the spectrum of an offline signal.

2.5.14.12 SelRMS() – SelectiveRootMeanSquare()

2.5.14.12.1 General

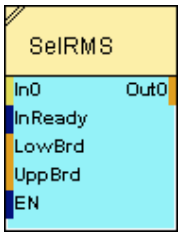
Properties

Property	Description
Full Name	SelectiveRootMeanSquare
Short Name	SelRMS
Description	Calculates the root mean square for the specified order or frequency range of the input spectrum.
Parent Library	Advanced
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = SelRMS
(
    Spectrum          In0 = 0,
    DataOrValue       InReady = false,
    Value             LFrqBrd = 0,
    Value             HFrqBrd = 0,
    DataOrValue       EN = true
);
```

Symbol



2.5.14.12.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	order or frequency spectrum whose selective root mean square shall be calculated
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the spectrum data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

LowBrd

Property	Description
Full Name	LowerBorder
Short Name	LowBrd
Description	lower border of the order or frequency range used for the selective root mean square calculation
Direction	input
Class	parameter
Default Value	0
Unit	1 or Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

UppBrd

Property	Description
Full Name	UpperBorder
Short Name	UppBrd
Description	upper border of the order or frequency range used for the selective root mean square calculation
Direction	input
Class	parameter
Default Value	0
Unit	1 or Hz
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.12.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the selective root mean square calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.14.12.4 Examples

Analyzing Model

Name	Output Data Class	Description
SelRMS 01	Online	Calculates root mean square of a generated signal by using its frequency spectrum.
SelRMS 02	Offline	Calculates root mean square of an offline signal by using its frequency spectrum.

Analyzing Script

Name	Output Data Class	Description
SelRMS 01	Online	Calculates root mean square of a generated signal by using its frequency spectrum.
SelRMS 02	Offline	Calculates root mean square of an offline signal by using its frequency spectrum.

2.5.14.13 THD() – TotalHarmonicDistortion()

2.5.14.13.1 General

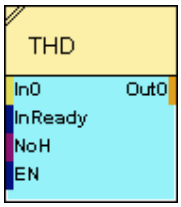
Properties

Property	Description
Full Name	TotalHarmonicDistortion
Short Name	THD
Description	Calculates the total harmonic distortion of the input signals.
Parent Library	Advanced
Parent Category	Spectral
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = THD
(
    Spectrum          In0 = 0,
    DataOrValue       InReady = false,
    Value             NoH = 1,
    DataOrValue       EN = true
);
```

Symbol



2.5.14.13.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	spectrum containing the values whose total harmonic distortion shall be calculated
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Spectrum
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the spectrum data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

NoH

Property	Description
Full Name	NumberOfHarmonics
Short Name	NoH
Description	determines number of harmonics which shall be used by the total harmonic distortion calculation
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 49
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.14.13.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the total harmonic distortion calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.14.13.4 Examples

Analyzing Model

Name	Output Data Class	Description
THD 01	Online	Calculates total harmonic distortion of a generated signal.
THD 02	Offline	Calculates total harmonic distortion of an offline signal.

Analyzing Script

Name	Output Data Class	Description
THD 01	Online	Calculates total harmonic distortion of a generated signal.
THD 02	Offline	Calculates total harmonic distortion of an offline signal.

2.5.15 Statistic

2.5.15.1 Avg() - Average()

2.5.15.1.1 General

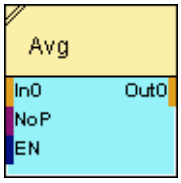
Properties

Property	Description
Full Name	Average
Short Name	Avg
Description	Calculates arithmetical average of the input data or values.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Avg
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose arithmetical average shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the arithmetical average calculation
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the arithmetical average calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
Avg 01	Online	Calculates the average of a generated sine wave.
Avg 02	Offline	Calculates the average of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Avg 01	Online	Calculates the average of a generated sine wave.
Avg 02	Offline	Calculates the average of an offline signal.

2.5.15.2 AvgBffr() - AverageBuffer()

2.5.15.2.1 General

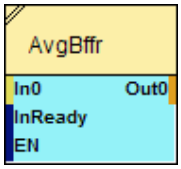
Properties

Property	Description
Full Name	AverageBuffer
Short Name	AvgBffr
Description	Calculates arithmetical average of the input buffer.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = AvgBffr
(
    Buffer1D    In0 = 0,
    DataOrValue InReady = false,
    DataOrValue EN = true
);
```

Symbol



2.5.15.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values whose arithmetical average shall be calculated
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the buffer arithmetical average calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
AvgBffr 01	Online	Calculates the average of a buffered generated sine wave.
AvgBffr 02	Offline	Calculates the average of a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
AvgBffr 01	Online	Calculates the average of a buffered generated sine wave.
AvgBffr 02	Offline	Calculates the average of a buffered offline signal.

2.5.15.3 Crest() - Crest()

2.5.15.3.1 General

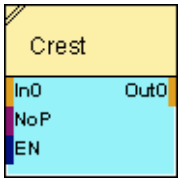
Properties

Property	Description
Full Name	Crest
Short Name	Crest
Description	Calculates the crest factor of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Crest
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.3.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose crest factor shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the crest factor calculation
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the crest factor calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
Crest 01	Online	Calculates the crest factor of a generated sine wave.
Crest 02	Offline	Calculates the crest factor of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Crest 01	Online	Calculates the crest factor of a generated sine wave.
Crest 02	Offline	Calculates the crest factor of an offline signal.

2.5.15.4 H1D() - Histogram1D()

2.5.15.4.1 General

Properties

Property	Description
Full Name	Histogram1D
Short Name	H1D
Description	Creates 1-dimensional histogram of the input data or values.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
Buffer1D Out0 = H1D
(
    DataOrValue  InX = 0,
    Value        LowBrdX = 0,
    Value        UppBrdX = 1,
    Value        DimX = 1,
    Value        ClssMdX = "Inside",
    Value        Update = 0.1,
    DataOrValue  Reset = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol

2.5.15.4.2 Inputs

InX

Property	Description
Full Name	InputX
Short Name	InX
Description	data or values whose 1-dimensional histogram shall be created
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

LowBrdX

Property	Description
Full Name	LowerBorderX
Short Name	LowBrdX
Description	data or value which shall be used as lower border of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

UppBrdX

Property	Description
Full Name	UpperBorderX
Short Name	UppBrdX
Description	data or value which shall be used as upper border of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

DimX

Property	Description
Full Name	DimensionX
Short Name	DimX
Description	data or value which shall be used as classes number of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 1024
Remark	none

ClssMdX

Property	Description
Full Name	ClassificationModeX
Short Name	ClssMdX
Description	contains the mode of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	“Inside”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Inside”, “Outside”
Remark	none

Update

Property	Description
Full Name	Update
Short Name	Update
Description	acquisition time for the release of the histogram
Direction	input
Class	parameter
Default Value	0.1
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the histogram calculation
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 1-dimensional histogram of the input data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the histogram data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.15.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
H1D 01	Online	Calculates the 1-dimensional histogram values of a generated signal.
H1D 02	Offline	Calculates the 1-dimensional histogram values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
H1D 01	Online	Calculates the 1-dimensional histogram values of a generated signal.
H1D 02	Offline	Calculates the 1-dimensional histogram values of an offline signal.

2.5.15.5 H2D() – Histogram2D()

2.5.15.5.1 General

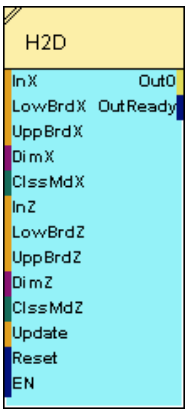
Properties

Property	Description
Full Name	Histogram2D
Short Name	H2D
Description	Creates 2-dimensional histogram of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
Buffer2D Out0 = H2D
(
    DataOrValue  InX = 0,
    Value        LowBrdX = 0,
    Value        UppBrdX = 1,
    Value        DimX = 1,
    Value        ClssMdX = "Inside",
    DataOrValue  InZ = 0,
    Value        LowBrdZ = 0,
    Value        UppBrdZ = 1,
    Value        DimZ = 1,
    Value        ClssMdZ = "Inside",
    Value        Update = 0.1,
    DataOrValue  Reset = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.15.5.2 Inputs

InX

Property	Description
Full Name	InputX
Short Name	InX
Description	data or values used as first input for the 2-dimensional histogram calculation
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

LowBrdX

Property	Description
Full Name	LowerBorderX
Short Name	LowBrdX
Description	data or value which shall be used as lower border of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

UppBrdX

Property	Description
Full Name	UpperBorderX
Short Name	UppBrdX
Description	data or value which shall be used as upper border of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

DimX

Property	Description
Full Name	DimensionX
Short Name	DimX
Description	data or value which shall be used as classes number of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 1024
Remark	none

ClssMdX

Property	Description
Full Name	ClassificationModeX
Short Name	ClssMdX
Description	contains the mode of InputX for the histogram classification
Direction	input
Class	parameter
Default Value	“Inside”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Inside”, “Outside”
Remark	none

InZ

Property	Description
Full Name	InputZ
Short Name	InZ
Description	data or values used as second input for the 2-dimensional histogram calculation
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

LowBrdZ

Property	Description
Full Name	LowerBorderZ
Short Name	LowBrdZ
Description	data or value which shall be used as lower border of InputZ for the histogram classification
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

UppBrdZ

Property	Description
Full Name	UpperBorderZ
Short Name	UppBrdZ
Description	data or value which shall be used as upper border of InputZ for the histogram classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

DimZ

Property	Description
Full Name	DimensionZ
Short Name	DimZ
Description	data or value which shall be used as classes number of InputZ for the histogram classification
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 1024
Remark	none

ClssMdZ

Property	Description
Full Name	ClassificationModeZ
Short Name	ClssMdZ
Description	contains the mode of InputZ for the histogram classification
Direction	input
Class	parameter
Default Value	“Inside”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Inside”, “Outside”
Remark	none

Update

Property	Description
Full Name	Update
Short Name	Update
Description	acquisition time for the release of the histogram
Direction	input
Class	parameter
Default Value	0.1
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	0 … max of Decimal64
Remark	in case Update = 0 the histogram will be released each Cycle Time defined in the parent Analyzing Model / Analyzing Script

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the histogram calculation
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 2-dimensional histogram of the input data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer2D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the histogram data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.15.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
H2D 01	Online	Calculates the 2-dimensional histogram values of generated signals.
H2D 02	Offline	Calculates the 2-dimensional histogram values of offline signals.

Analyzing Script

Name	Output Data Class	Description
H2D 01	Online	Calculates the 2-dimensional histogram values of generated signals.
H2D 02	Offline	Calculates the 2-dimensional histogram values of offline signals.

2.5.15.6 Kurtosis() - Kurtosis()

2.5.15.6.1 General

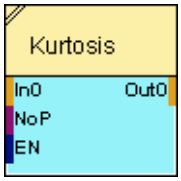
Properties

Property	Description
Full Name	Kurtosis
Short Name	Kurtosis
Description	Calculates the kurtosis of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Kurtosis
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.6.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose kurtosis shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the kurtosis calculation
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.6.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the kurtosis calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.6.4 Examples

Analyzing Model

Name	Output Data Class	Description
Kurtosis 01	Online	Calculates the kurtosis values of a generated signal.
Kurtosis 02	Offline	Calculates the kurtosis values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Kurtosis 01	Online	Calculates the kurtosis values of a generated signal.
Kurtosis 02	Offline	Calculates the kurtosis values of an offline signal.

2.5.15.7 Max() - Maximum()

2.5.15.7.1 General

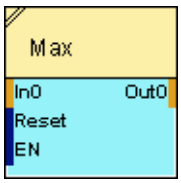
Properties

Property	Description
Full Name	Maximum
Short Name	Max
Description	Detects global maximum of the input data or values.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Max
(
    DataOrValue In0 = 0,
    DataOrValue Reset = false,
    DataOrValue EN = true
);
```

Symbol



2.5.15.7.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose global maximum shall be detected
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the global maximum detection
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.7.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the global maximum detection
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.7.4 Examples

Analyzing Model

Name	Output Data Class	Description
Max 01	Online	Calculates the global maximum value of a generated signal.
Max 02	Offline	Calculates the global maximum value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Max 01	Online	Calculates the global maximum value of a generated signal.
Max 02	Offline	Calculates the global maximum value of an offline signal.

2.5.15.8 MaxBffr() - MaximumBuffer()

2.5.15.8.1 General

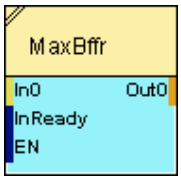
Properties

Property	Description
Full Name	MaximumBuffer
Short Name	MaxBffr
Description	Detects global maximum of the input buffer.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = MaxBffr
(
    Buffer1D    In0 = 0,
    DataOrValue InReady = false,
    DataOrValue EN = true
);
```

Symbol



2.5.15.8.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values whose global maximum shall be detected
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.8.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the buffer global maximum detection
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.8.4 Examples

Analyzing Model

Name	Output Data Class	Description
MaxBffr 01	Online	Calculates the global maximum value of a buffered generated signal.
MaxBffr 02	Offline	Calculates the global maximum value of a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
MaxBffr 01	Online	Calculates the global maximum value of a buffered generated signal.
MaxBffr 02	Offline	Calculates the global maximum value of a buffered offline signal.

2.5.15.9 Min() - Minimum()

2.5.15.9.1 General

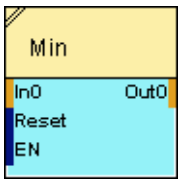
Properties

Property	Description
Full Name	Minimum
Short Name	Min
Description	Detects global minimum of the input data or values.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Min
(
    DataOrValue In0 = 0,
    DataOrValue Reset = false,
    DataOrValue EN = true
);
```

Symbol



2.5.15.9.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose global minimum shall be detected
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the global minimum detection
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.9.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the global minimum detection
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.9.4 Examples

Analyzing Model

Name	Output Data Class	Description
Min 01	Online	Calculates the global minimum value of a generated signal.
Min 02	Offline	Calculates the global minimum value of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Min 01	Online	Calculates the global minimum value of a generated signal.
Min 02	Offline	Calculates the global minimum value of an offline signal.

2.5.15.10 MinBffr() - MinimumBuffer()

2.5.15.10.1 General

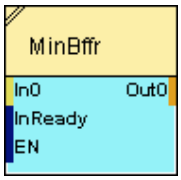
Properties

Property	Description
Full Name	MinimumBuffer
Short Name	MinBffr
Description	Detects global minimum of the input buffer.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = MinBffr
(
    Buffer1D    In0 = 0,
    DataOrValue InReady = false,
    DataOrValue EN = true
);
```

Symbol



2.5.15.10.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer containing the values whose global minimum shall be detected
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.10.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the buffer global minimum detection
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.10.4 Examples

Analyzing Model

Name	Output Data Class	Description
MinBffr 01	Online	Calculates the global minimum value of a buffered generated signal.
MinBffr 02	Offline	Calculates the global minimum value of a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
MinBffr 01	Online	Calculates the global minimum value of a buffered generated signal.
MinBffr 02	Offline	Calculates the global minimum value of a buffered offline signal.

2.5.15.11 Range() - Range()

2.5.15.11.1 General

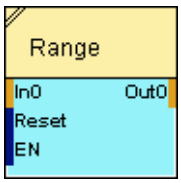
Properties

Property	Description
Full Name	Range
Short Name	Range
Description	Calculates the global range of the input data or values.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Range
(
    DataOrValue In0 = 0,
    DataOrValue Reset = false,
    DataOrValue EN = true
);
```

Symbol



2.5.15.11.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose global range shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the global range calculation
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.11.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the global range calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.11.4 Examples

Analyzing Model

Name	Output Data Class	Description
Range 01	Online	Calculates the global range values of a generated sinus wave.
Range 02	Offline	Calculates the global range values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Range 01	Online	Calculates the global range values of a generated sine wave.
Range 02	Offline	Calculates the global range values of an offline signal.

2.5.15.12 RMS() - RootMeanSquare()

2.5.15.12.1 General

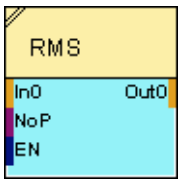
Properties

Property	Description
Full Name	RootMeanSquare
Short Name	RMS
Description	Calculates the root mean square of the input data or values.
Parent Library	Base
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = RMS
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.12.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose root mean square shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the root mean square calculation
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.12.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the root mean square calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.12.4 Examples

Analyzing Model

Name	Output Data Class	Description
RMS 01	Online	Calculates the root mean square values of a generated sine wave.
RMS 02	Offline	Calculates the root mean square values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
RMS 01	Online	Calculates the root mean square values of a generated sine wave.
RMS 02	Offline	Calculates the root mean square values of an offline signal.

2.5.15.13 Skewness () - Skewness()

2.5.15.13.1 General

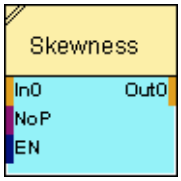
Properties

Property	Description
Full Name	Skewness
Short Name	Skewness
Description	Calculates the skewness of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Skewness
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.13.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose skewness shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the skewness calculation
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.13.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the skewness calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.13.4 Examples

Analyzing Model

Name	Output Data Class	Description
Skewness 01	Online	Calculates the skewness values of a generated signal.
Skewness 02	Offline	Calculates the skewness values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Skewness 01	Online	Calculates the skewness values of a generated sine wave.
Skewness 02	Offline	Calculates the skewness values of an offline signal.

2.5.15.14 StdDev() – StandardDeviation()

2.5.15.14.1 General

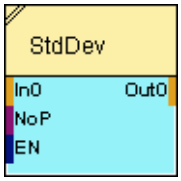
Properties

Property	Description
Full Name	StandardDeviation
Short Name	StdDev
Description	Calculates the sample standard deviation of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = StdDev
(
    DataOrValue In0 = 0,
    Value       NoP = 2,
    DataOrValue EN = true
);
```

Symbol



2.5.15.14.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose sample standard deviation shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the sample standard deviation calculation
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.14.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the sample standard deviation calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.14.4 Examples

Analyzing Model

Name	Output Data Class	Description
StdDev 01	Online	Calculates the sample standard deviation values of a generated signal.
StdDev 02	Offline	Calculates the sample standard deviation values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
StdDev 01	Online	Calculates the sample standard deviation values of a generated signal.
StdDev 02	Offline	Calculates the sample standard deviation values of an offline signal.

2.5.15.15 StdDevN() – StandardDeviationN()

2.5.15.15.1 General

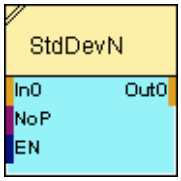
Properties

Property	Description
Full Name	StandardDeviationN
Short Name	StdDevN
Description	Calculates the standard deviation of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = StdDevN
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.15.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose standard deviation shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the standard deviation calculation
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.15.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the standard deviation calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.15.4 Examples

Analyzing Model

Name	Output Data Class	Description
StdDevN 01	Online	Calculates the standard deviation values of a generated signal.
StdDevN 02	Offline	Calculates the standard deviation values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
StdDevN 01	Online	Calculates the standard deviation values of a generated signal.
StdDevN 02	Offline	Calculates the standard deviation values of an offline signal.

2.5.15.16 Vrnc() – Variance()

2.5.15.16.1 General

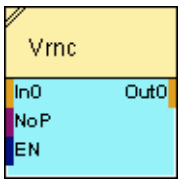
Properties

Property	Description
Full Name	Variance
Short Name	Vrnc
Description	Calculates the empirical variance of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = Vrnc
(
    DataOrValue In0 = 0,
    Value       NoP = 2,
    DataOrValue EN = true
);
```

Symbol



2.5.15.16.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose empirical variance shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the empirical variance calculation
Direction	input
Class	parameter
Default Value	2
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	2 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.16.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the empirical variance calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.16.4 Examples

Analyzing Model

Name	Output Data Class	Description
Vrnc 01	Online	Calculates the empirical variance values of a generated signal.
Vrnc 02	Offline	Calculates the empirical variance values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Vrnc 01	Online	Calculates the empirical variance values of a generated signal.
Vrnc 02	Offline	Calculates the empirical variance values of an offline signal.

2.5.15.17 VrncN() – VarianceN()

2.5.15.17.1 General

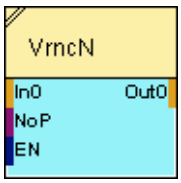
Properties

Property	Description
Full Name	VarianceN
Short Name	VrncN
Description	Calculates the variance of the input data or values.
Parent Library	Advanced
Parent Category	Statistic
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-based, Sequential

Function Prototype

```
DataOrValue Out0 = VrncN
(
    DataOrValue In0 = 0,
    Value       NoP = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.15.17.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values whose variance shall be calculated
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points which shall be used for the variance calculation
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.15.17.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the result of the variance calculation
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.15.17.4 Examples

Analyzing Model

Name	Output Data Class	Description
VrncN 01	Online	Calculates the variance values of a generated signal.
VrncN 02	Offline	Calculates the variance values of an offline signal.

Analyzing Script

Name	Output Data Class	Description
VrncN 01	Online	Calculates the variance values of a generated signal.
VrncN 02	Offline	Calculates the variance values of an offline signal.

2.5.16 String

2.5.16.1 FrmtStr() - FormatString()

2.5.16.1.1 General

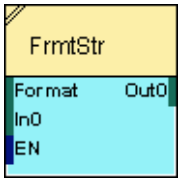
Properties

Property	Description
Full Name	FormatString
Short Name	FrmtStr
Description	Produces string using parameterized format text-string.
Parent Library	Advanced
Parent Category	String
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = FrmtStr
(
    Value       Format,
    DataOrValue In0 ... In7 = 0;
    DataOrValue EN = true
);
```

Symbol



2.5.16.1.2 Inputs

Format

Property	Description
Full Name	Format
Short Name	Format
Description	template format for the outputted string
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string contained format which shall be used for the text-string parameterization
Remark	none

In0 … In7

Property	Description
Full Name	Input0 … Input7
Short Name	In0 … In7
Description	data or values used as parameters for Format pattern
Direction	input
Class	data
Default Value	“”
Unit	n/a
Configurable	8
Supported Data Types	String, Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	string contained parameters for the text-string parameterization
Remark	none

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments Input0 … Input7 are used by the calculation
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.16.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the formatted text-string
Direction	output
Class	data
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.16.1.4 Examples

Analyzing Model

For the Analyzing Function **FrmtStr()** there are no example Analyzing Models available in Example Location.

Analyzing Script

Name	Output Data Class	Description
FrmtStr 01	Online	Calculates and prints result of the arithmetical operation after a defined pattern.
FrmtStr 02	Offline	Calculates and prints result of the arithmetical operation after a defined pattern.

2.5.17 System

2.5.17.1 GetTime() - GetTime()

2.5.17.1.1 General

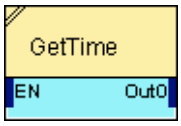
Properties

Property	Description
Full Name	GetTime
Short Name	GetTime
Description	Returns the total number of seconds which have elapsed since 1970-01-01.
Parent Library	Advanced
Parent Category	System
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = GetTime
(
    DataOrValue EN = true
);
```

Symbol



2.5.17.1.2 Inputs

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.17.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the amount of seconds which have elapsed since 1970-01-01 for the GMT time zone
Direction	output
Class	data
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	the resolution of the returned value is in ms (three digits after the comma)

2.5.17.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
GetTime 01	Online	Calculates the number of elapsed seconds for the current time.

Analyzing Script

Name	Output Data Class	Description
GetTime 01	Online	Calculates the number of elapsed seconds for the current time.

2.5.17.2 MonCF() - MonitorConfigurationFile()

2.5.17.2.1 General

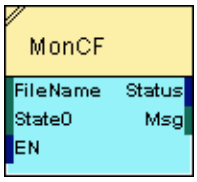
Properties

Property	Description
Full Name	MonitorConfigurationFile
Short Name	MonCF
Description	Monitors the state of the specified Configuration File.
Parent Library	Advanced
Parent Category	System
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based

Function Prototype

```
DataOrValue Status = MonCF
(
    Value         FileName = "",
    Value         State0 ... State3 = "All States";
    DataOrValue   EN = true,
    DataOrValue* Msg = NULL
);
```

Symbol



2.5.17.2.2 Inputs

FileName

Property	Description
Full Name	FileName
Short Name	FileName
Description	full path and name of the Configuration File which shall be monitored, in format "Location Name://[Directory/]Configuration File Name.Extension", e.g. "Example Location://Arithmetic/Abs 01.amf.xts"
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the file name and its symbolic path
Remark	the value of FileName is case insensitive; the specified file name must point to an existing Interface Profile File ("*.ipf.xts"), Device Profile File ("*.dpf.xts"), Analyzing Model ("*.amf.xts"), Analyzing Script ("*.asf.xts") or Storage Profile File ("*.spf.xts")

State0 … State3

Property	Description
Full Name	State0 … State3
Short Name	State0 … State3
Description	specifies the states of the Configuration File which shall be monitored
Direction	input
Class	parameter
Default Value	“All States”
Unit	n/a
Configurable	4
Supported Data Types	String
Allowed manual Inputs	“All States”, “Started”, “Stopped”, “Suspended”, “Paused”, “Triggering”
Remark	in case the Configuration File is in a specified state already at the first call to MonCF(), MonCF() creates a message with the current state immediately at its first execution

In0_3

Property	Description
Full Name	Input0_3
Short Name	In0_3
Description	determines how many of the input arguments State0 … State3 are used by the monitoring
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 4
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.17.2.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates whether a new message is available
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case one of the specified states has been reached, otherwise Status = false

Msg

Property	Description
Full Name	Message
Short Name	Msg
Description	receives the message which indicates a Configuration File state
Direction	output
Class	data
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.17.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
MonCF 01	Online	Monitors a specified Configuration File for the started state.

Analyzing Script

Name	Output Data Class	Description
MonCF 01	Online	Monitors a specified Configuration File for the started state.

2.5.17.3 MonDev() - MonitorDevice()

2.5.17.3.1 General

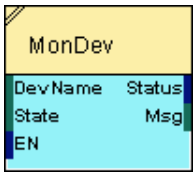
Properties

Property	Description
Full Name	MonitorDevice
Short Name	MonDev
Description	Monitors the state of the specified device.
Parent Library	Advanced
Parent Category	System
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based

Function Prototype

```
DataOrValue Status = MonDev
(
    Value        DevName = "",
    Value        State = "All States",
    DataOrValue  EN = true,
    DataOrValue* Msg = NULL
);
```

Symbol



2.5.17.3.2 Inputs

DevName

Property	Description
Full Name	DeviceName
Short Name	DevName
Description	name of the device which shall be monitored
Direction	input
Class	parameter
Default Value	""
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the device name
Remark	The value of DeviceName is case sensitive.

State

Property	Description
Full Name	State
Short Name	State
Description	specifies the state of the device which shall be monitored
Direction	input
Class	parameter
Default Value	“All States”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“All States”, “Attached”, “Detached”
Remark	In case the device is in a specified state already at the first call to MonDev(), MonDev() creates a message with the current state immediately at its first execution.

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.17.3.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates whether a new message is available
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case one of the specified states has been reached, otherwise Status = false

Msg

Property	Description
Full Name	Message
Short Name	Msg
Description	receives the message which indicates a device state
Direction	output
Class	data
Default Value	“”
Unit	1
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.17.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
MonDev 01	Online	Monitors a specified simulation device for the attached state.

Analyzing Script

Name	Output Data Class	Description
MonDev 01	Online	Monitors a specified simulation device for the attached state.

2.5.17.4 MonDisk() - MonitorDisk()

2.5.17.4.1 General

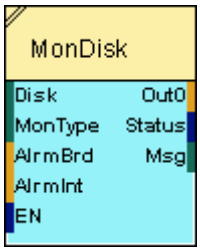
Properties

Property	Description
Full Name	MonitorDisk
Short Name	MonDisk
Description	Monitors the usage of the specified disk.
Parent Library	Advanced
Parent Category	System
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based

Function Prototype

```
DataOrValue Out0 = MonDisk
(
    Value         Disk = "C:",
    Value         MonType = "Free Space",
    Value         AlrmBrd = 1000,
    Value         AlrmInt = 3600,
    DataOrValue   EN = true,
    DataOrValue* Status = NULL,
    DataOrValue* Msg = NULL
);
```

Symbol



2.5.17.4.2 Inputs

Disk

Property	Description
Full Name	Disk
Short Name	Disk
Description	specifies the name of the disk which shall be monitored
Direction	input
Class	parameter
Default Value	"C:"
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains a valid disk partition in format "XXX:"
Remark	The value of Disk is case insensitive.

MonType

Property	Description
Full Name	MonitoringType
Short Name	MonType
Description	specifies the property of the disk which shall be monitored
Direction	input
Class	parameter
Default Value	“Free Space”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“Free Space”, “Used Space”
Remark	none

AlrmBrd

Property	Description
Full Name	AlarmBorder
Short Name	AlrmBrd
Description	specifies the border which will generate the message in case it is exceeded
Direction	input
Class	parameter
Default Value	1000
Unit	MByte
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	in case MonitoringType = “Free Space” alarm message will be produced if detected free space drops below than specified AlarmBorder, in case MonitoringType = “Used Space” alarm message will be produced if detected used space rises higher than specified AlarmBorder, in case Alarm-Border <= 0 alarming is disabled

AlrmInt

Property	Description
Full Name	AlarmInterval
Short Name	AlrmInt
Description	specifies the shortest interval at which the alarm message can be created
Direction	input
Class	parameter
Default Value	3600
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … 86400
Remark	a value of 0 disables the alarm interval. In this case the alarm message is not being repeated automatically and the next alarm message is created only in case the violation of the alarm border ended and the alarm border becomes violated again

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.17.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the current value of the monitored property of the disk
Direction	output
Class	data
Default Value	0
Unit	MByte
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates whether a new message is available
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case the specified alarm constraint has been broken, otherwise Status = false

Msg

Property	Description
Full Name	Message
Short Name	Msg
Description	receives the message which indicates an exceeded border
Direction	output
Class	data
Default Value	“”
Unit	1
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.17.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
MonDisk 01	Online	Monitors a specified disk partition for using of the free space.

Analyzing Script

Name	Output Data Class	Description
MonDisk 01	Online	Monitors a specified disk partition for using of the free space setting.

2.5.17.5 MonLog() - MonitorSystemLog()

2.5.17.5.1 General

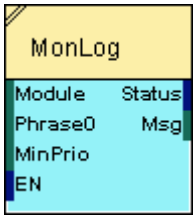
Properties

Property	Description
Full Name	MonitorSystemLog
Short Name	MonLog
Description	Extracts the entries of the system log which match the specified source modules, phrases and priorities.
Parent Library	Advanced
Parent Category	System
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based

Function Prototype

```
DataOrValue Status = MonLog
(
    Value         Module = "All",
    Value         Phrase0 ... Phrase7 = "";
    Value         MinPrio = "Warning",
    DataOrValue   EN = true,
    DataOrValue* Msg = NULL
);
```

Symbol



2.5.17.5.2 Inputs

Module

Property	Description
Full Name	Module
Short Name	Module
Description	specifies from which source modules an entry to the system log must be created
Direction	input
Class	parameter
Default Value	"All"
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	"All", "ANE", "DME", "MTE", "STE", "ANE+DME+STE", "DME+STE", "Other"
Remark	none

Phrase0 … Phrase7

Property	Description
Full Name	Phrase0 … Phrase7
Short Name	Phrase0 … Phrase7
Description	specifies which phrases must be present within an entry to the system log
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	8
Supported Data Types	String
Allowed manual Inputs	string which contains a phrase for the monitoring
Remark	spelling of the Filter0 … Filter7 is case insensitive, in case of the empty string all system log entries will be monitored

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments Phrase0 … Phrase7 are used by the monitoring
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

MinPrio

Property	Description
Full Name	MinimumPriority
Short Name	MinPrio
Description	specifies the minimum priority of an entry to the system log
Direction	input
Class	parameter
Default Value	“Warning”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	“User”, “Information”, “Normal”, “Warning”, “Error”, “Fatal”
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.17.5.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates whether a new message is available
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case a matching entry to the system log has been detected, otherwise Status = false

Msg

Property	Description
Full Name	Message
Short Name	Msg
Description	receives a message which contains the matching entries of the system log
Direction	output
Class	data
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.17.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
MonLog 01	Online	Monitors a phrase in the system log for all modules.

Analyzing Script

Name	Output Data Class	Description
MonLog 01	Online	Monitors a phrase in the system log for all modules.

2.5.17.6 SndEmail() - SendEmail()

2.5.17.6.1 General

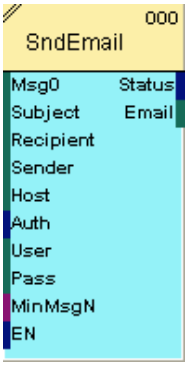
Properties

Property	Description
Full Name	SendEmail
Short Name	SndEmail
Description	Builds emails from text messages and sends to the specified address afterwards.
Parent Library	Advanced
Parent Category	System
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Status = SndEmail
(
    DataOrValue  Msg0 ... Msg7 = "";
    Value        Subject = "",
    Value        Recipient = "",
    Value        Sender = "",
    Value        Host = "",
    Value        Auth = true,
    Value        User = "",
    Value        Pass = "",
    Value        MinMsgN = 1,
    DataOrValue  EN = true,
    DataOrValue* Email = NULL
);
```

Symbol



2.5.17.6.2 Inputs

Msg0 … Msg7

Property	Description
Full Name	Message0 … Message7
Short Name	Msg0 … Msg7
Description	messages which shall be sent to the specified email address
Direction	input
Class	data
Default Value	“”
Unit	n/a
Configurable	8
Supported Data Types	String
Allowed manual Inputs	string which contains a message for the sending
Remark	empty messages are not treated as messages and are not sent

In0_7

Property	Description
Full Name	Input0_7
Short Name	In0_7
Description	determines how many of the input arguments Message0 … Message7 are used by the sending
Direction	input
Class	configuration
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer8
Allowed manual Inputs	1 … 8
Remark	within the Analyzing Script Editor, a semicolon must be placed after the last used value of the configurable argument

Subject

Property	Description
Full Name	Subject
Short Name	Subject
Description	identifies email address of the recipient
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains the subject which shall be used when the email is being sent
Remark	in case the subject is left empty, the Email Subject setting from the global options is being used for the sending of the email

Recipient

Property	Description
Full Name	RecipientAddress
Short Name	Recipient
Description	identifies email address of the recipient
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains a valid email address in format “xxx@yyy.zzz”
Remark	In case the recipient is left empty, the “Sender”, “Host”, “Auth”, “User” and “Pass” inputs are ignored and the email notification settings from the global options are being used for the sending of the email.

Sender

Property	Description
Full Name	SenderAddress
Short Name	Sender
Description	identifies email address of the sender
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains a valid email address in format “xxx@yyy.zzz”
Remark	none

Host

Property	Description
Full Name	MailServerHostName
Short Name	Host
Description	specifies email gateway address
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains a valid host name of the email server
Remark	none

Auth

Property	Description
Full Name	Authentication
Short Name	Auth
Description	indicates whether the user authentication at the email server shall be performed
Direction	input
Class	parameter
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

User

Property	Description
Full Name	UserName
Short Name	User
Description	specifies user name for the user authentication at the email server
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains a valid user name for the user authentication
Remark	none

Pass

Property	Description
Full Name	Password
Short Name	Pass
Description	specifies password for the user authentication at the email server
Direction	input
Class	parameter
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	string which contains a valid password for the user authentication
Remark	none

MinMsgN

Property	Description
Full Name	MinimumMessageNumber
Short Name	MinMsgN
Description	determines how many of the messages shall be collected for the email sending
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … 100
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	if the internal calculation is disabled, the already received messages are discarded and the filling of the internal message stack is restarted when the internal calculation is enabled again

2.5.17.6.3 Outputs

Status

Property	Description
Full Name	Status
Short Name	Status
Description	indicates whether a new email has been sent
Direction	output
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	Status = true in case a new email has been sent, otherwise Status = false

Email

Property	Description
Full Name	Email
Short Name	Email
Description	receives the content of sent email
Direction	output
Class	data
Default Value	“”
Unit	n/a
Configurable	1
Supported Data Types	String
Allowed manual Inputs	n/a
Remark	none

2.5.17.6.4 Examples

The Example Location does not provide ready Analyzing Models or Analyzing Scripts which use the Analyzing Function **SendEmail()** because the necessary information about the email server (IP, port, username

2.5.18 Time

2.5.18.1 Bffr() - Buffer()

2.5.18.1.1 General

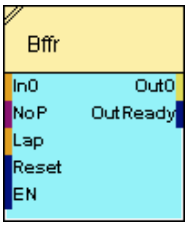
Properties

Property	Description
Full Name	Buffer
Short Name	Bffr
Description	Creates 1-dimensional buffer with defined size contained input data or values.
Parent Library	Base
Parent Category	Time
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
Buffer1D Out0 = Bffr
(
    DataOrValue  In0 = 0,
    Value        NoP = 1024,
    Value        Lap = 0,
    DataOrValue  Reset = false,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.18.1.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be stored in the buffer
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points containing in the buffer
Direction	input
Class	parameter
Default Value	1024
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

Lap

Property	Description
Full Name	Overlap
Short Name	Lap
Description	defines the overlapping length of the input data or values
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	0 … 1)
Remark	in case Overlap = 0 no overlapping will be occurred

Reset

Property	Description
Full Name	Reset
Short Name	Reset
Description	resets the buffer filling
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	if the internal calculation is disabled, the already received input buffer is discarded and the filling of the internal buffer is restarted when the internal calculation is enabled again

2.5.18.1.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the 1-dimensional buffer with the input data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

OutReady

Property	Description
Full Name	OutputReady
Short Name	OutReady
Description	indicates whether the buffer data in Output0 is valid
Direction	output
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean
Allowed manual Inputs	n/a
Remark	none

2.5.18.1.4 Examples

Analyzing Model

Name	Output Data Class	Description
Bffr 01	Online	Creates a 1-dimensional buffer of a generated sine wave.
Bffr 02	Offline	Creates a 1-dimensional buffer of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Bffr 01	Online	Creates a 1-dimensional buffer of a generated sine wave.
Bffr 02	Offline	Creates a 1-dimensional buffer of an offline signal.

2.5.18.2 Dly() - Delay()

2.5.18.2.1 General

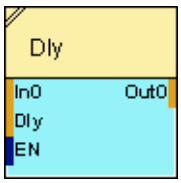
Properties

Property	Description
Full Name	Delay
Short Name	Dly
Description	Delays input signal.
Parent Library	Base
Parent Category	Time
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Dly
(
    DataOrValue In0 = 0,
    Value       Dly = 1,
    DataOrValue EN = true
);
```

Symbol



2.5.18.2.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be delayed
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Dly

Property	Description
Full Name	Delay
Short Name	Dly
Description	delay time for the input data or values
Direction	input
Class	parameter
Default Value	1
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	2^{-52} … max of Decimal64
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.18.2.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the delayed input data or values
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.18.2.4 Examples

Analyzing Model

Name	Output Data Class	Description
Dly 01	Online	Delays a generated sine wave by 1 second.
Dly 02	Offline	Delays an offline signal by 2 seconds.

Analyzing Script

Name	Output Data Class	Description
Dly 01	Online	Delays a generated sine wave by 1.34 seconds.
Dly 02	Offline	Delays an offline signal by 2 seconds.

2.5.18.3 GetBffrM() - GetBufferMember()

2.5.18.3.1 General

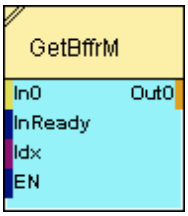
Properties

Property	Description
Full Name	GetBufferMember
Short Name	GetBffrM
Description	Gets indicated member of the input buffer.
Parent Library	Base
Parent Category	Time
Equidistant Inputs	No
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
DataOrValue Out0 = GetBffrM
(
    Buffer1D    In0 = 0,
    DataOrValue InReady = false,
    Value       Idx = 0,
    DataOrValue EN = true
);
```

Symbol



2.5.18.3.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	buffer whose indicated member shall be provided to the output
Direction	input
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the buffer data in Input0 is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

Idx

Property	Description
Full Name	Index
Short Name	Idx
Description	indicates the sequence number of the buffer member which shall be provided to the output
Direction	input
Class	parameter
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	0 … max of Integer32
Remark	indexation of the buffer member begins from 0

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.18.3.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the indicated member of the input buffer
Direction	output
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.18.3.4 Examples

Analyzing Model

Name	Output Data Class	Description
GetBffrM 01	Online	Returns a value from a buffered online signal.
GetBffrM 02	Offline	Returns a value from a buffered offline signal.

Analyzing Script

Name	Output Data Class	Description
GetBffrM 01	Online	Returns a value from a buffered online signal.
GetBffrM 02	Offline	Returns a value from a buffered offline signal.

2.5.18.4 Rsmpl() - Resample()

2.5.18.4.1 General

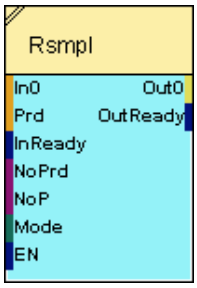
Properties

Property	Description
Full Name	Resample
Short Name	Rsmpl
Description	Re-samples input data or values with specified number of points.
Parent Library	Advanced
Parent Category	Time
Equidistant Inputs	Yes
Supported Operating Modes	Time-Slice-Based, Sequential

Function Prototype

```
Buffer1D Out0 = Rsmpl
(
    DataOrValue  In0 = 0,
    DataOrValue  Prd = 0,
    DataOrValue  InReady = false,
    Value        NoPrd = 1,
    Value        NoP = 1024,
    DataOrValue  EN = true,
    DataOrValue* OutReady = NULL
);
```

Symbol



2.5.18.4.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	data or values which shall be used for the re-sampling
Direction	input
Class	data
Default Value	0
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	min … max of Decimal64
Remark	none

Prd

Property	Description
Full Name	Period
Short Name	Prd
Description	period duration of the fundamental oscillation of the input data or values
Direction	input
Class	data
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	(0 … max of Decimal64
Remark	none

InReady

Property	Description
Full Name	InputReady
Short Name	InReady
Description	indicates whether the data in Period is valid
Direction	input
Class	control
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

NoPrd

Property	Description
Full Name	NumberOfPeriods
Short Name	NoPrd
Description	determines number of periods which shall be used for the re-sampling
Direction	input
Class	parameter
Default Value	1
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	1 … max of Integer32
Remark	none

NoP

Property	Description
Full Name	NumberOfPoints
Short Name	NoP
Description	determines number of points for the re-sampled data or values
Direction	input
Class	parameter
Default Value	1024
Unit	1
Configurable	1
Supported Data Types	Integer32
Allowed manual Inputs	128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.18.4.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the buffer with the re-sampled data or values
Direction	output
Class	data
Default Value	0
Unit	n/a
Configurable	1
Supported Data Types	Buffer1D
Allowed manual Inputs	n/a
Remark	none

2.5.18.4.4 Examples

Analyzing Model

Name	Output Data Class	Description
Rsmpl 01	Online	Calculates a buffer data with re-sampled input online signal.
Rsmpl 02	Offline	Calculates a buffer data with re-sampled input offline signal.

Analyzing Script

Name	Output Data Class	Description
Rsmpl 01	Online	Calculates a buffer data with re-sampled input online signal.
Rsmpl 02	Offline	Calculates a buffer data with re-sampled input offline signal.

2.5.18.5 Stopper() - Stopper()

2.5.18.5.1 General

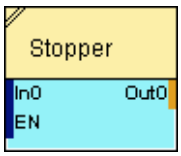
Properties

Property	Description
Full Name	Stopper
Short Name	Stopper
Description	Calculates duration of a binary signal.
Parent Library	Base
Parent Category	Time
Equidistant Inputs	No
Supported Operating Modes	Time-Slice based, Sequential

Function Prototype

```
DataOrValue Out0 = Stopper
(
    DataOrValue In0 = false,
    DataOrValue EN = true
);
```

Symbol



2.5.18.5.2 Inputs

In0

Property	Description
Full Name	Input0
Short Name	In0
Description	binary data or values whose duration shall be calculated
Direction	input
Class	data
Default Value	false
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

EN

Property	Description
Full Name	Enable
Short Name	EN
Description	enables or disables the internal calculation
Direction	input
Class	control
Default Value	true
Unit	1
Configurable	1
Supported Data Types	Boolean, Integer8, Integer16, Integer32, Integer64, Decimal32, Decimal64
Allowed manual Inputs	false, true
Remark	none

2.5.18.5.3 Outputs

Out0

Property	Description
Full Name	Output0
Short Name	Out0
Description	receives the calculated duration of the input data or values
Direction	output
Class	data
Default Value	0
Unit	s
Configurable	1
Supported Data Types	Decimal64
Allowed manual Inputs	n/a
Remark	none

2.5.18.5.4 Examples

Analyzing Model

Name	Output Data Class	Description
Stopper 01	Online	Stop clock of a generated signal.
Stopper 02	Offline	Stop clock of an offline signal.

Analyzing Script

Name	Output Data Class	Description
Stopper 01	Online	Stop clock of a generated signal.
Stopper 02	Offline	Stop clock of an offline signal.

3 Contact Information

Address

Siemens AG

I IA CE SE

Wuerzburger Strasse 121

90766 Fuerth

Germany

Internet

http://www.siemens.com/siplus-cms