

**SIEMENS**

*Ingenuity for life*

*Industry Online Support*

Home

# Library of General Functions (LGF) for SIMATIC S7-1200 / S7-1500

STEP 7 Basic/Professional (TIA PORTAL)

<https://support.industry.siemens.com/cs/ww/en/view/109479728>

Siemens  
Industry  
Online  
Support



# Legal information

## Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

## Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

## Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (<https://support.industry.siemens.com>) shall also apply.

## Security information

Siemens provides products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place. For additional information on industrial security measures that may be implemented, please visit <https://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: <https://www.siemens.com/industrialsecurity>.

# Table of contents

	<b>Legal information</b> .....	<b>2</b>
<b>1</b>	<b>Library overview</b> .....	<b>7</b>
1.1	General information .....	7
1.2	Hardware and software requirements .....	7
<b>2</b>	<b>Working with the library</b> .....	<b>8</b>
2.1	General information .....	8
2.2	Simulation capability with SIMATIC S7-PLCSIM Advanced .....	9
2.3	User-defined documentation (user help) .....	10
<b>3</b>	<b>Explanation of the blocks</b> .....	<b>12</b>
<b>4</b>	<b>Program blocks</b> .....	<b>13</b>
4.1	Bit logic operations .....	13
4.1.1	LGF_BitReset (FC / V3.0.1) .....	13
4.1.2	LGF_BitSet (FC / V3.0.1) .....	14
4.1.3	LGF_BitSetTo (FC / V3.0.1) .....	15
4.1.4	LGF_BitTest (FC / V3.0.1) .....	16
4.1.5	LGF_BitToggle (FC / V3.0.1) .....	17
4.1.6	LGF_PulseRelay (FB / V3.0.1) .....	18
4.2	Date and timer operations .....	20
4.2.1	LGF_GetCalendarDay (FC / V3.0.1) .....	20
4.2.2	LGF_GetCalendarWeek_ISO (FC / V3.0.1) .....	22
4.2.3	LGF_GetCalendarWeek_US (FC / V3.0.1) .....	24
4.2.4	LGF_IsGermanHoliday (FC / V3.0.1) .....	26
4.2.5	LGF_AstroClock (FB / V3.0.1) .....	28
4.2.6	LGF_SetTime (FB / V3.0.2) .....	33
4.2.7	LGF_TimerSwitch (FB / V3.0.1) .....	38
4.3	Counter operations .....	41
4.3.1	LGF_BitCount (FC / V3.0.2) .....	41
4.3.2	LGF_CountBooleanEdges (FB / V1.0.0) .....	42
4.3.3	LGF_CountFallInDWord (FB / V3.0.1) .....	43
4.3.4	LGF_CountRiseInDWord (FB / V3.0.1) .....	45
4.3.5	LGF_GetBitStates (FB / V1.0.0) .....	47
4.4	Comperator operations .....	48
4.4.1	LGF_BinaryMaskCompare (FC / V1.0.0) .....	48
4.4.2	LGF_CompareLReal (FC / V3.0.1) .....	49
4.4.3	LGF_CompareLRealByPrecision (FC / V3.0.1) .....	51
4.4.4	LGF_CompareVariant (FC / V3.0.1) .....	52
4.5	Math operations .....	55
4.5.1	LGF_CalcDistance_2D (FC / V3.0.1) .....	55
4.5.2	LGF_CalcDistance_3D (FC / V3.0.1) .....	57
4.5.3	LGF_GetFactorial (FC / V3.0.1) .....	59
4.5.4	LGF_IsValueInLimits (FC / V3.0.1) .....	61
4.5.5	LGF_IsValueInRange (FC / V3.0.1) .....	63
4.5.6	LGF_IsValueInTolerance (FC / V3.0.2) .....	65
4.5.7	LGF_NthRoot (FC / V3.0.1) .....	67
4.5.8	LGF_Random_DInt (FC / V3.0.1) .....	69
4.5.9	LGF_Random_Real (FC / V3.0.1) .....	71
4.5.10	LGF_Random_UDInt (FC / V3.0.1) .....	73
4.5.11	LGF_RandomRange_Dint (FC / V3.0.1) .....	75
4.5.12	LGF_RandomRange_Real (FC / V3.0.1) .....	77

4.5.13	LGF_RandomRange_UDInt (FC / V3.0.1) .....	79
4.5.14	LGF_ScaleLinear (FC / V3.0.1).....	81
4.5.15	LGF_SearchMinMax (FC / V3.0.1).....	84
4.5.16	LGF_SearchMinMax_DInt (FC / V3.0.1) .....	87
4.5.17	LGF_SearchMinMax_LReal (FC / V3.0.1) .....	89
4.5.18	LGF_SearchMinMax_UDInt (FC / V3.0.1) .....	91
4.5.19	LGF_Integration (FB / V3.0.2).....	93
4.5.20	LGF_StoreMinMax (FB / V3.0.1).....	96
4.6	Math operations / Matrix.....	97
4.6.1	LGF_MatrixAddition (FC / V3.0.1).....	97
4.6.2	LGF_MatrixCompare (FC / V3.0.1) .....	99
4.6.3	LGF_MatrixInverse (FC / V3.0.1) .....	101
4.6.4	LGF_MatrixMultiplication (FC / V3.0.1) .....	103
4.6.5	LGF_MatrixScalarMultiplication (FC / V3.0.1).....	105
4.6.6	LGF_MatrixSubtraction (FC / V3.0.1).....	107
4.6.7	LGF_MatrixTranspose (FC / V3.0.1).....	109
4.7	Data handling .....	111
4.7.1	LGF_CalcCRC16 (FC / V3.0.1).....	111
4.7.2	LGF_CalcCRC16Advanced (FC / V3.0.1).....	113
4.7.3	LGF_CalcCRC32 (FC / V3.0.1).....	115
4.7.4	LGF_CalcCRC32Advanced (FC / V3.0.1).....	117
4.7.5	LGF_CalcCRC8 (FC / V3.0.1).....	119
4.7.6	LGF_CalcCRC8Advanced (FC / V3.0.1).....	121
4.7.7	LGF_CalcCRC8For1Byte (FC / V3.0.1) .....	123
4.7.8	LGF_IsParityEven (FC / V3.0.1) .....	125
4.7.9	LGF_IsParityOdd (FC / V3.0.1) .....	126
4.7.10	LGF_FIFO (FB / V3.0.1).....	127
4.7.11	LGF_LIFO (FB / V3.0.1) .....	130
4.7.12	LGF_ShellSort_DInt (FB / V3.0.1).....	133
4.7.13	LGF_ShellSort_LReal (FB / V3.0.1) .....	135
4.7.14	LGF_ShellSort_UDInt (FB / V3.0.1) .....	137
4.7.15	LGF_ShiftRegister (FB / V1.0.3) .....	139
4.8	Converter operations.....	142
4.8.1	LGF_BinaryToGray (FC / V3.0.1) .....	142
4.8.2	LGF_DTLtoString_DE (FC / V3.0.1) .....	143
4.8.3	LGF_DTLtoString_ISO (FC / V3.0.1) .....	145
4.8.4	LGF_DTLtoUnixTime (FC / V3.0.1) .....	147
4.8.5	LGF_GpsDDToGps (FC / V3.0.2) .....	149
4.8.6	LGF_GpsToGpsDD (FC / V3.0.2) .....	151
4.8.7	LGF_GrayToBinary (FC / V3.0.1) .....	153
4.8.8	LGF_IntToString (FC / V3.0.1).....	154
4.8.9	LGF_StringToDTL_DE (FC / V3.0.1) .....	155
4.8.10	LGF_StringToDTL_ISO (FC / V3.0.1) .....	157
4.8.11	LGF_StringToInt (FC / V3.0.1).....	159
4.8.12	LGF_StringToTaddr (FC / V3.0.1).....	160
4.8.13	LGF_StringToTime (FC / V3.0.1) .....	162
4.8.14	LGF_TaddrToString (FC / V3.0.1).....	163
4.8.15	LGF_TimeToString (FC / V3.0.1) .....	165
4.8.16	LGF_UnixTimeToDTL (FC / V3.0.1) .....	167
4.9	Converter operations / Binary types - Split and Merge .....	169
4.9.1	LGF_MergeBitsToByte (FC / V3.0.1) .....	169
4.9.2	LGF_MergeBitsToDWord (FC / V3.0.1) .....	171
4.9.3	LGF_MergeBitsToWord (FC / V3.0.1).....	174
4.9.4	LGF_MergeBytesToDWord (FC / V3.0.1) .....	176
4.9.5	LGF_MergeBytesToWord (FC / V3.0.1).....	177
4.9.6	LGF_MergeWordsToDWord (FC / V3.0.1).....	178



4.9.7	LGF_SplitByteToBits (FC / V3.0.1)	179
4.9.8	LGF_SplitDWordToBits (FC / V3.0.1)	181
4.9.9	LGF_SplitDWordToBytes (FC / V3.0.1)	184
4.9.10	LGF_SplitDWordToWords (FC / V3.0.1)	185
4.9.11	LGF_SplitWordToBits (FC / V3.0.1)	186
4.9.12	LGF_SplitWordToBytes (FC / V3.0.1)	188
4.10	Converter operations / Temperature	189
4.10.1	LGF_CelsiusToFahrenheit (FC / V3.0.1)	189
4.10.2	LGF_CelsiusToKelvin (FC / V3.0.1)	190
4.10.3	LGF_ConvertTemperature (FC / V3.0.1)	191
4.10.4	LGF_FahrenheitToCelsius (FC / V3.0.1)	193
4.10.5	LGF_FahrenheitToKelvin (FC / V3.0.1)	194
4.10.6	LGF_KelvinToCelsius (FC / V3.0.1)	195
4.10.7	LGF_KelvinToFahrenheit (FC / V3.0.1)	196
4.10.8	LGF_KelvinToRankine (FC / V3.0.1)	197
4.10.9	LGF_RankineToKelvin (FC / V3.0.1)	198
4.11	Signal generators	199
4.11.1	LGF_CosinusCI (FB / V3.0.1)	199
4.11.2	LGF_Frequency (FB / V3.0.1)	202
4.11.3	LGF_Impulse (FB / V3.0.1)	204
4.11.4	LGF_RectangleCI (FB / V3.0.1)	206
4.11.5	LGF_SawToothCI (FB / V3.0.1)	209
4.11.6	LGF_SinusCI (FB / V3.0.1)	212
4.11.7	LGF_TriangleCI (FB / V3.0.1)	215
4.12	Technology operations	218
4.12.1	LGF_LimRateOfChangeAdvancedCI (FB / V3.0.1)	218
4.12.2	LGF_LimRateOfChangeCI (FB / V3.0.1)	223
4.12.3	LGF_NonLinearInterpolation (FB / V3.0.1)	226
4.12.4	LGF_RampCI (FB / V3.0.1)	230
4.12.5	Rules of simulated controlled systems	236
4.13	Measurement operations	237
4.13.1	LGF_AverageAndDeviation (FC / V3.0.1)	237
4.13.2	LGF_DifferenceQuotientFC (FC / V3.0.1)	239
4.13.3	LGF_RegressionLine (FC / V3.0.1)	241
4.13.4	LGF_SimpleSmoothingFC (FC / V3.0.1)	244
4.13.5	LGF_SmoothByPolynomFC (FC / V3.0.1)	246
4.13.6	LGF_Boxplot_DInt (FB / V3.0.1)	248
4.13.7	LGF_Boxplot_LReal (FB / V3.0.1)	252
4.13.8	LGF_Boxplot_UDInt (FB / V3.0.1)	256
4.13.9	LGF_DifferenceQuotientFB (FB / V3.0.1)	260
4.13.10	LGF_FloatingAverage (FB / V3.0.2)	262
4.13.11	LGF_Histogram_DInt (FB / V3.0.1)	264
4.13.12	LGF_Histogram_LReal (FB / V3.0.1)	267
4.13.13	LGF_Histogram_UDInt (FB / V3.0.1)	270
4.13.14	LGF_SimpleSmoothingFB (FB / V3.0.1)	273
4.13.15	LGF_SmoothByPolynomFB (FB / V3.0.1)	275
4.14	Legacy / Counter operations	277
4.14.1	LGF_CountFallInDWord (FC / V3.0.1)	277
4.14.2	LGF_CountRisInDWord (FC / V3.0.1)	280
4.15	Legacy / Signal generators	283
4.15.1	LGF_SawTooth (FB / V3.0.1)	283
<b>5</b>	<b>PLC data types</b>	<b>286</b>
5.1	Date and timer operations	286
5.1.1	LGF_typeGPS (UDT)	286

## Table of contents

---

5.1.2	LGF_typeGPS_DD (UDT) .....	286
5.1.3	LGF_typeGPS_DMS (UDT) .....	286
5.2	Technology operations .....	287
5.2.1	LGF_typeNonLinSetpoints (UDT) .....	287
5.2.2	LGF_typeRampTimeTable (UDT) .....	287
5.3	Measurement operations .....	288
5.3.1	LGF_typeRegressionLine (UDT) .....	288
<b>6</b>	<b>PLC tags &amp; constants</b> .....	<b>289</b>
6.1.1	LGF_CONSTANTS .....	289
6.1.2	LGF_CONSTANTS_UNITS .....	291
6.1.3	LMATH_CONSTANTS .....	292
6.1.4	LMATH_CONSTANTS_EXTENDED .....	296
6.1.5	LPHYSICS_CONSTANTS .....	299
<b>7</b>	<b>Appendix</b> .....	<b>301</b>
7.1	Service and support .....	301
7.2	Industry Mall .....	302
7.3	Links and Literature .....	302
7.4	Change documentation .....	303
<b>8</b>	<b>Change log</b> .....	<b>304</b>

# 1 Library overview

## 1.1 General information

TIA Portal has an extensive number of “ready-to-use” instructions (mathematical functions, times, counters, etc.). In addition, there are other useful basic functions. These functions are provided in the form of a library and they can be used freely. The finished functions are freely customizable and can therefore be used universally.

The library described here is versioned and it will be continuously extended. For information on versioning, see Chapter “Versioning”.

## 1.2 Hardware and software requirements

### Requirements for this library

In order to be able to use the functionality of the library described here, the following hardware and software requirements must be met.

#### Hardware

All blocks (FB, FC, DB, ...) in the library can be used universally with the following controllers:

- SIMATIC S7-1200 and SIMATIC S7-1200 F product family (from firmware V4.2)
- SIMATIC S7-1500 and SIMATIC S7-1500 F product family (from firmware V2.0)
- Simulation with SIMATIC S7-PLCSIM (from V14)

#### Software

- SIMATIC STEP 7 Basic/Professional (TIA PORTAL)

#### Note

In general, it is possible to open a library with STEP 7 Basic, although STEP 7 Professional elements (e.g. SIMATIC S7-1500 controller) are included. In this case you will be informed with a message when opening the library.

All elements (types and copy templates) can be used if they are supported by the hardware installed in the TIA Portal.

If you try to copy elements with STEP 7 Basic from the library that are not supported (e.g. SIMATIC S7-1500 controller), an error message is displayed.

## 2 Working with the library

### 2.1 General information

All blocks in the “LGF” library can be used freely in conjunction with SIMATIC S7-1200 and SIMATIC S7-1500 controllers.

Most of the blocks are stored as types in the library. This means that the blocks are versioned and can thus use all advantages.

- Central update function for library elements
- Versioning of library elements

#### Note

Information on the general handling of libraries is provided in the Guideline for Library Handling

<https://support.industry.siemens.com/cs/ww/en/view/109747503>

and in the Programming Guideline for S7-1200/1500 in the chapter “Libraries”.

<https://support.industry.siemens.com/cs/ww/en/view/81318674>

#### Note

All blocks in the LGF were created in accordance with the Programming Style Guide.

<https://support.industry.siemens.com/cs/ww/en/view/81318674>

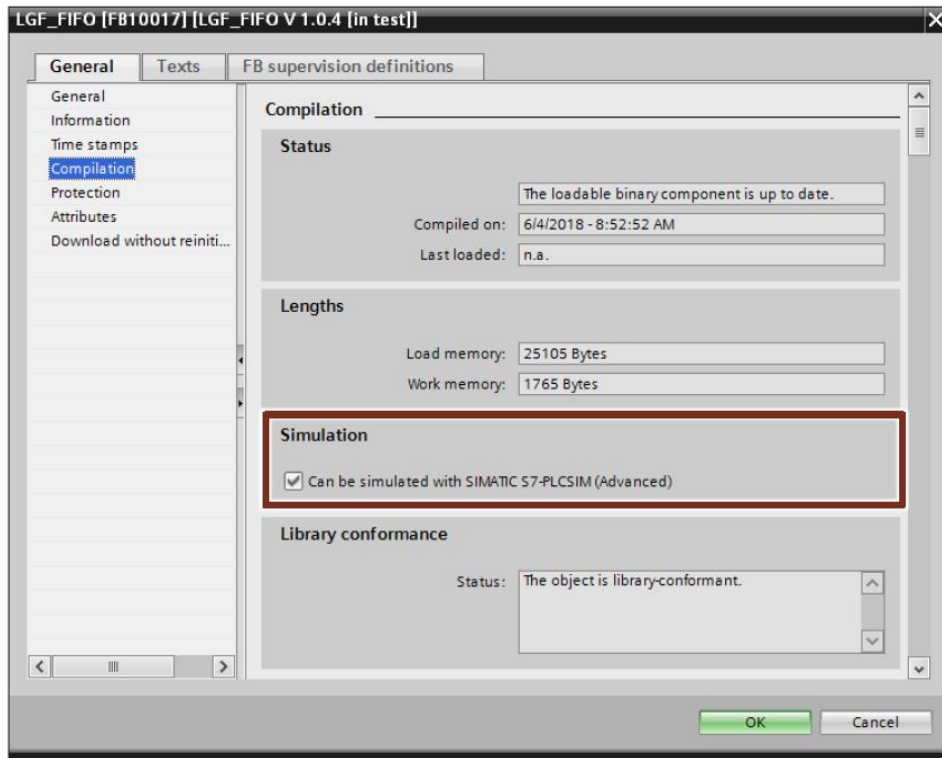
For more information on libraries, visit the TIA Portal:

- Libraries in the TIA Portal  
<https://support.industry.siemens.com/cs/ww/en/view/109738702>
- How do you open libraries in STEP 7 (TIA Portal)?  
<https://support.industry.siemens.com/cs/ww/en/view/37364723>
- Automate in less than 10 minutes TIA Portal: Time Savers – Global libraries  
<https://support.industry.siemens.com/cs/ww/en/view/78529894>
- Which elements from STEP 7 (TIA Portal) can be stored in a library as a type or as a copy template?  
<https://support.industry.siemens.com/cs/ww/en/view/109476862>
- How can you automatically open a global library when starting TIA Portal V13 or higher and use it e.g. as a corporate library?  
<https://support.industry.siemens.com/cs/ww/en/view/100451450>
- Library with PLC data types for IO module / technology modules and PROFIdrive drives (LPD)  
<https://support.industry.siemens.com/cs/ww/en/view/109482396>



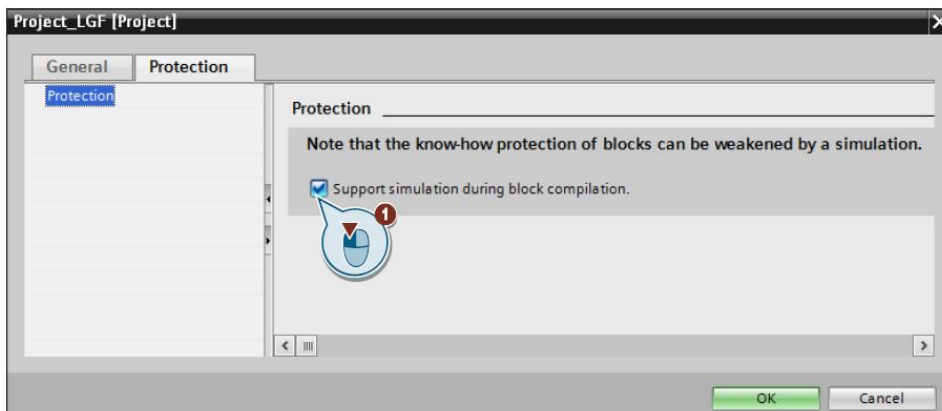
## 2.2 Simulation capability with SIMATIC S7-PLCSIM Advanced

Simulation with SIMATIC S7-PLCSIM Advanced is already activated in the properties of the LGF blocks.



After translation with SIMATIC S7-PLCSIM Advanced, proceed as follows so that the blocks can be simulated.

1. Open the properties of your project and activate the option “Support simulation during block compilation” in the “Protection” tab.



**Note** Blocks with activated simulation capability take up more memory space in the PLC.

## 2.3 User-defined documentation (user help)

In order to explain the principle of operation and use of the blocks to users of the LGF library, user-defined documentation has been created for each block. The user-defined documentation per block is available in German and English as a PDF file. The PDF files are stored in the following directories of the LGF library.

- German: "UserFiles\UserDocumentation\en-DE\Library Types"
- English: "UserFiles\UserDocumentation\en-US\Library Types"

The user-defined documentation for a block can be called up in the Task Card "Library" and in the library view with the key combination <Shift+F1>.

The respective PDF is always opened with the standard program defined in Microsoft Windows.

So that the user-defined documentation of the blocks can also be called up in the project navigation, you must copy the directories with the PDF files into the project directory `UserFiles`.

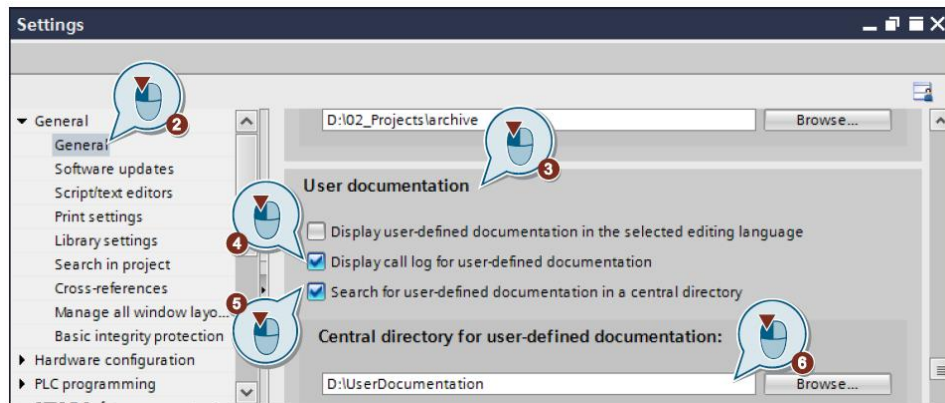
### Note

For user-defined documentation, you require SIMATIC STEP 7 Basic / Professional V15.1 Update 1.

### Central directory for user-defined documentation

Alternatively, you can store the user-defined documentation in a central directory for all projects. To define a central storage location for user help, proceed as follows:

1. In the "Options" menu, select the "Settings" command.
2. Open the area "General > General".
3. Navigate to the "User documentation" section.
4. Activate the checkbox "Display call log for user-defined documentation" to display a log of the call-up of the user-defined documentation in the Inspector window.
5. Activate the "Search for user-defined documentation in a central directory" checkbox to store user-defined documentation in a central directory for projects.
6. In the "Central directory for user-defined documentation" field, specify the path where you want to store cross-project documentation.



**Note** Do not change the names of the PDF, because the file name must precisely match the name of the object in the TIA Portal.

**Note** Further information on the user-defined documentation can be found in the system manual “SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1”:  
<https://support.industry.siemens.com/cs/ww/en/view/109755202/114872699275>

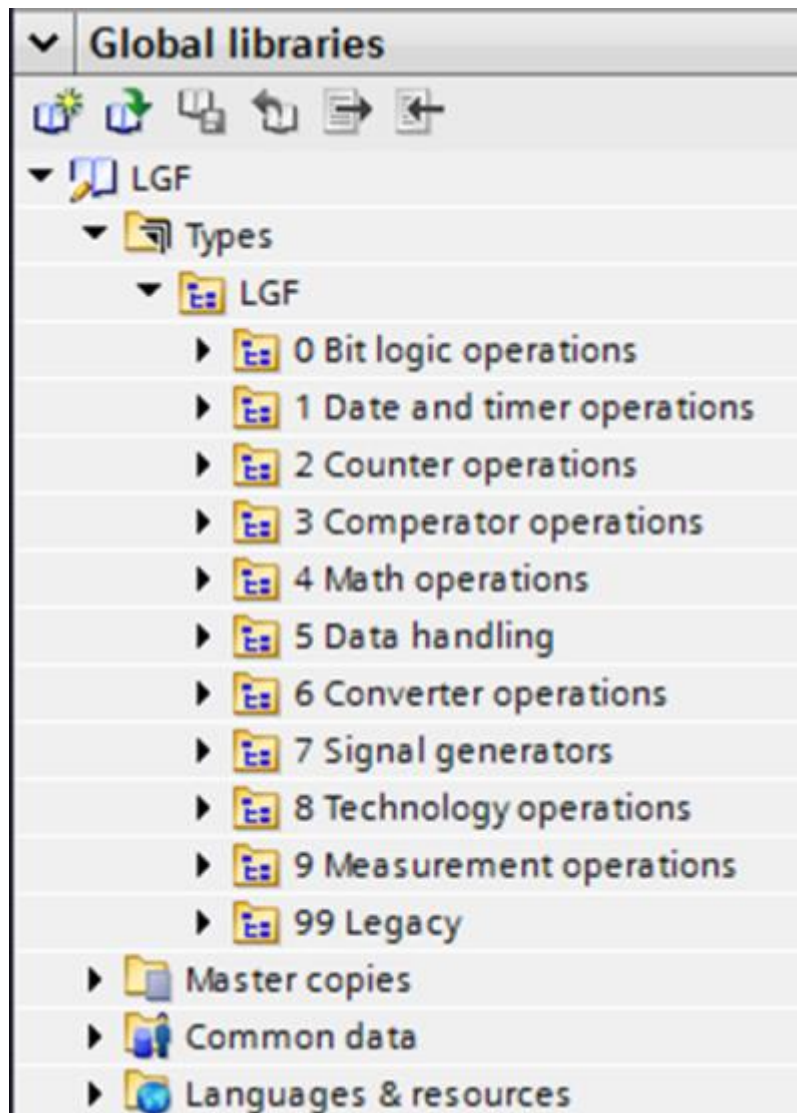
## 3 Explanation of the blocks

The chapters below describe all blocks of the library, "Library of General Functions".

The chapters have the same structure as the library itself.

All blocks are divided into application areas or categories:

- Bit logic operations
- Date and timer operations
- Counter operations
- Comparator operations
- Math operations
- Data handling
- Converter operations
- Signal generators
- Technology operations
- Measurement operations





## 4 Program blocks

### 4.1 Bit logic operations

#### 4.1.1 LGF\_BitReset (FC / V3.0.1)

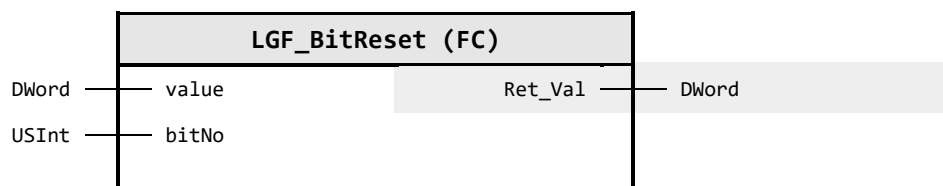
Author: Siemens SIMATIC Systems Support

##### Short description

This block resets a bit at a predefined position in a variable of the data type DWORD.

Alternatively, Word and Byte can be used instead of DWord by converting the passed parameter with, for example, BYTE\_TO\_DWORD and the result with DWORD\_TO\_BYTE.

##### Block Interface



##### Input parameter

Identifier	Data type	Description
value	DWord	Tag where the bit has to be reset
bitNo	USInt	Bit number to reset in "value" parameter

##### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Tag with reset bit

##### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Support</b> first release
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation

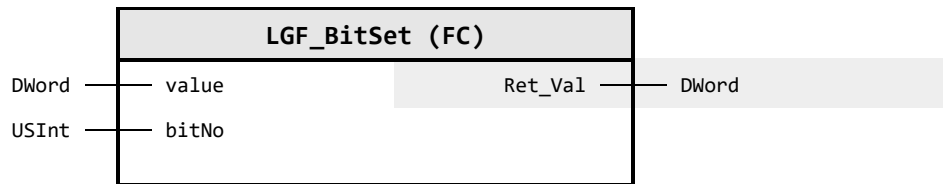
### 4.1.2 LGF\_BitSet (FC / V3.0.1)

Author: Siemens SIMATIC Systems Support

#### Short description

This block sets a bit at a given position in a variable of the data type DWORD. Alternatively, Word and Byte can be used instead of DWord by converting the passed parameter with, for example, BYTE\_TO\_DWORD and the result with DWORD\_TO\_BYTE.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	DWord	Tag where the bit has to be set
bitNo	USInt	Bit number to set in "value" parameter

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Tag with the set bit

#### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Support</b> first release
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.1.3 LGF\_BitSetTo (FC / V3.0.1)

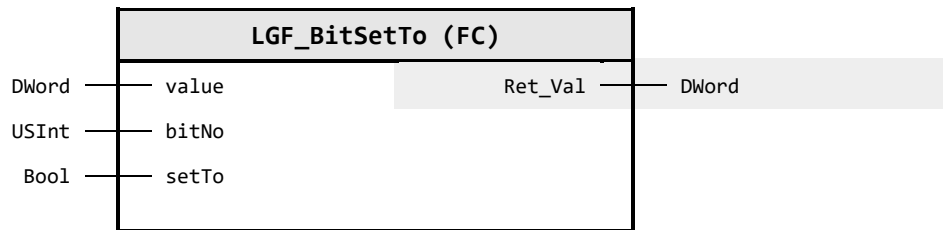
Author: Siemens SIMATIC Systems Support

#### Short description

This block sets a bit to TRUE or FALSE at a predefined position in a variable of the data type DWORD.

Alternatively, Word and Byte can be used instead of DWord by converting the passed parameter with, for example, BYTE\_TO\_DWORD and the result with DWORD\_TO\_BYTE.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	DWord	Tag where the bit has to be set / reset
bitNo	USInt	Bit number to set in "value" parameter
setTo	Bool	Set bit to FALSE / TRUE

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Tag with set bit

#### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Support</b> first release
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation

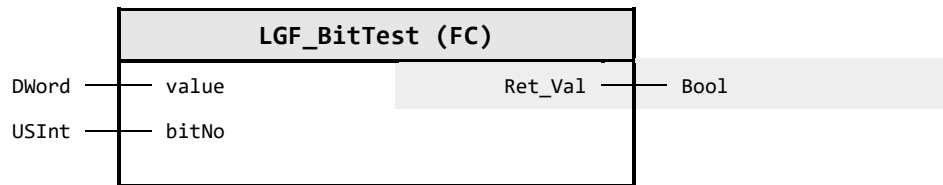
#### 4.1.4 LGF\_BitTest (FC / V3.0.1)

Author: Siemens SIMATIC Systems Support

##### Short description

This block checks whether a bit is TRUE or FALSE at a given position in a variable of the data type DWORD.  
Alternatively, Word and Byte can be used instead of DWord by converting the passed parameter with, for example, BYTE\_TO\_DWORD and the result with DWORD\_TO\_BYTE.

##### Block Interface



##### Input parameter

Identifier	Data type	Description
value	DWord	Tag where the bit has to be tested
bitNo	USInt	bit number to test in "value" parameter

##### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	Value of the checked bit.

##### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Support</b> first release
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation



### 4.1.5 LGF\_BitToggle (FC / V3.0.1)

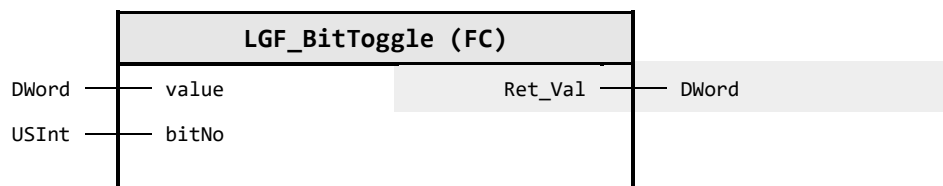
Author: Siemens SIMATIC Systems Support

#### Short description

This block toggles (from TRUE to FALSE and vice versa) a bit at a predefined position in a variable of the data type DWord.

Alternatively, Word and Byte can be used instead of DWord by converting the passed parameter with, for example, BYTE\_TO\_DWORD and the result with DWORD\_TO\_BYTE

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	DWord	Tag where the bit has to be toggled
bitNo	USInt	Bit number to be toggled in the "value" parameter.

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Tag with toggled bit

#### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Support</b> first release
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation

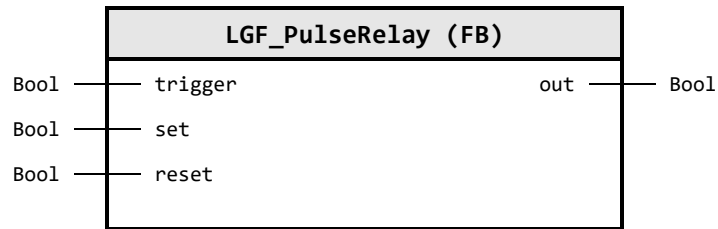
### 4.1.6 LGF\_PulseRelay (FB / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This block corresponds to an impulse relay or a toggle flip-flop including set and reset input. Pulse relay, Surge relay, Toggle-Flip-Flop, Frequency divider `reset` is leading / prior to `set` or `trigger`

#### Block Interface



#### Input parameter

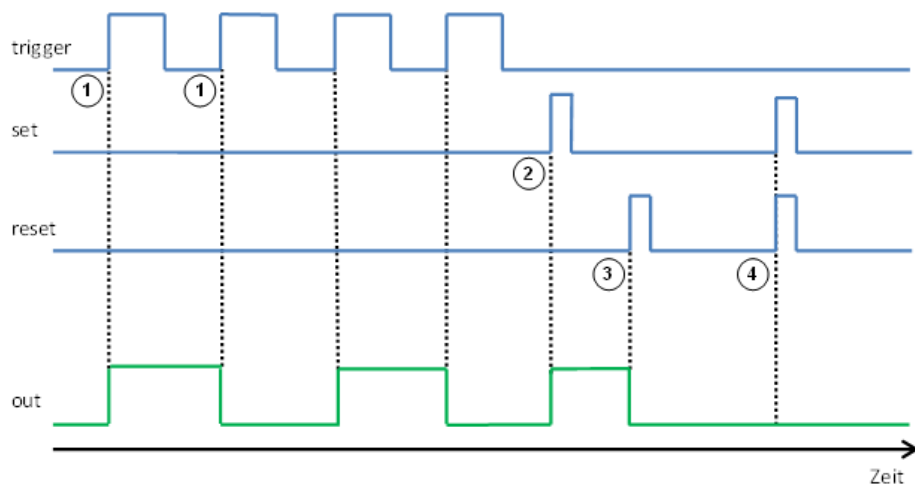
Identifier	Data type	Default value	Description
trigger	Bool	FALSE	Trigger to toggle output signal (rising edge)
set	Bool	FALSE	Set output signal. rising edge
reset	Bool	FALSE	Reset signal, rising edge (prior to set)

#### Output parameter

Identifier	Data type	Description
out	Bool	Output signal

#### Functional description

Figure: LGF\_PulseRelay Signal diagram



1. Each rising edge of the input `trigger` changes the Boolean value of the output `out`.
2. Each rising edge of the input `set` sets the Boolean value of the output `out` to TRUE.

## 4 Program blocks

- Each rising edge of the input `reset` sets the Boolean value of the output `out` to `FALSE`.
- If the inputs `set` and `reset` are set in the same cycle, the `reset` input has priority.

The block can also be used as a frequency divider. If the input `trigger` is supplied with a fixed frequency, the output `out` delivers half the frequency.

### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Comment correction
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.05</b> 24.05.2019	<b>Simatic Systems Support</b> Refactoring and performance improvement add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation

## 4.2 Date and timer operations

### 4.2.1 LGF\_GetCalendarDay (FC / V3.0.1)

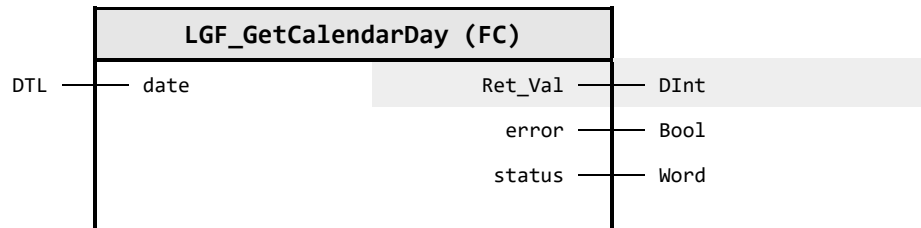
Author: Siemens Digital Industry

#### Short description

This function uses the specified date to calculate the number of days that have passed since the beginning of the year (1st January).

The function is used in the functions “LGF\_GetCalendarWeek\_ISO” and “LGF\_GetCalendarWeek\_US”.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	DTL	Date for the calculation of the calendar days since 1 January.

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Days past since January 1st.
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8201	ERR_LIM_DATE Date out of the range, has to be greater than <1970-01-01 ; 2262-04-11>

### Change log

Version & Date	Change description
<b>01.00.00</b> 16.07.2019	<b>Simatic Systems Support</b> First release ENO used for internal error handling, interface has error and status temp tag naming, insert constant
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation

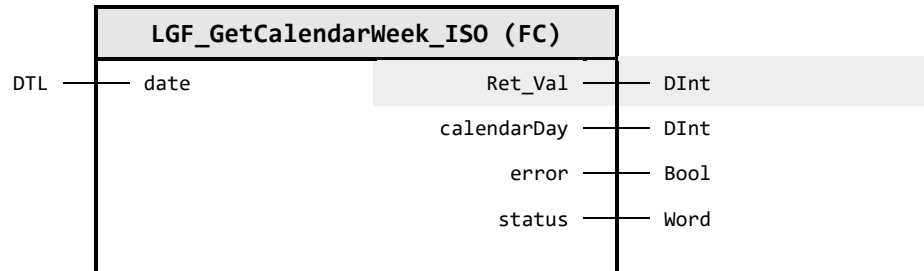
### 4.2.2 LGF\_GetCalendarWeek\_ISO (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function uses the specified date to calculate the calendar week and the number of days that have passed since the beginning of the year for ISO 8601 European countries.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	DTL	Date used to calculate the calendar week and days since 1 January

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Number of the calendar week.
calendarDay	DInt	Days past since January 1st on given date
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8201	ERR_LIM_DATE Date out of the range, has to be greater than <1970-01-01 ; 2262-04-11>

#### Functional description

##### Counting method for European countries in accordance with ISO 8601

- Calendar weeks have 7 days, start on a Monday, and they are counted continuously throughout the year
- Calendar week 1 of a year is the week that contains the first Thursday.
- Each year has either 52 or 53 calendar weeks.
- A year has 53 calendar weeks if the following characteristics apply:
  - A common year begins on a Thursday and ends on a Thursday.
  - A leap year begins either on a Wednesday and ends on a Thursday or it begins on a Thursday and ends on a Friday.

## 4 Program blocks

---

- The 29th, 30th and 31st December can belong to the calendar week 1 of the following year.
- The 1st, 2nd, and 3rd January can still belong to the last calendar week of the previous year.

### Change log

Version & Date	Change description
<b>01.00.00</b> 27.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 16.07.2019	<b>Simatic Systems Support</b> Renamed from LGF_CalenderWeek to LGF_CalenderWeek_ISO Function split into week for ISO and US Format and as well day counter. Result passed as return value. Standard header implemented Constant, temp variable naming Update function call of CalendarDay
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation

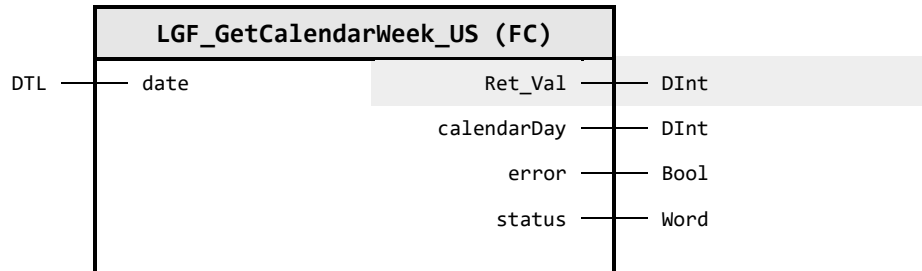
### 4.2.3 LGF\_GetCalendarWeek\_US (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function uses the specified date to calculate the calendar week and the number of days that have passed since the beginning of the year for the USA and many other countries.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	DTL	Date used to calculate the calendar week and days since 1 January

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Number of the calendar week.
calendarDay	DInt	Days past since January 1st on given date
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8201	ERR_LIM_DATE Date out of the range, has to be greater than <1970-01-01 ; 2262-04-11>

#### Functional description

##### Counting method for the USA and many other countries

- Calendar weeks have 7 days, start on a Sunday and are counted continuously throughout the year
- Calendar week 1 of a year is the week that contains January 1.
- Each year has either 52 or 53 calendar weeks.
- A year has 53 calendar weeks if the following characteristics apply:
  - A common year begins on a Saturday and ends on a Saturday.
  - A leap year begins either on a Saturday and ends on a Sunday or it begins on a Friday and ends on a Saturday.



#### 4 Program blocks

---

- The days after the last Saturday in December can belong to the first calendar week of the following year.

#### Change log

Version & Date	Change description
<b>01.00.00</b> 13.11.2019	<b>Simatic Systems Support</b> First release based on spli from LGF_CalenderWeek (previously LGF_CalenderWeek_ISO)
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation

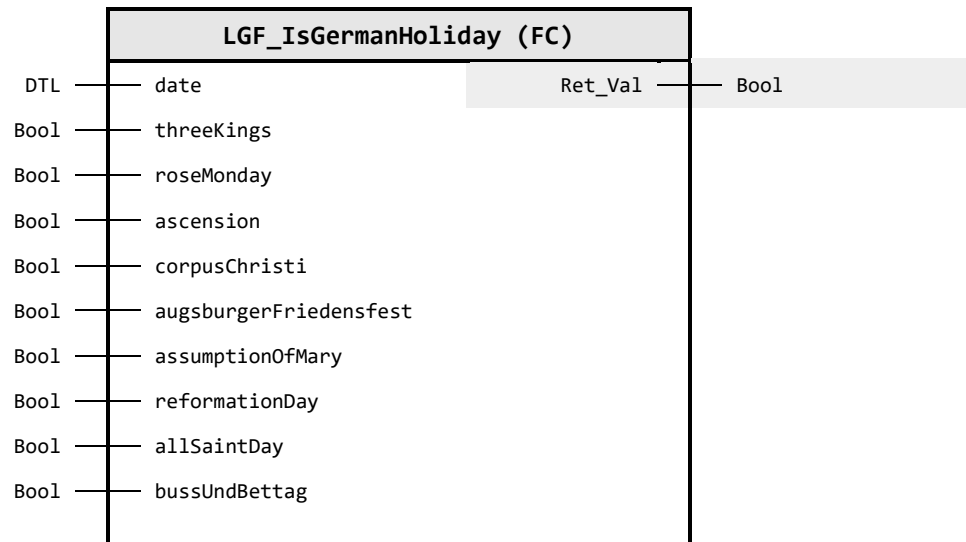
### 4.2.4 LGF\_IsGermanHoliday (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

The function determines whether a given date is a public holiday.  
 All public holidays in Germany are taken into account.  
 Holidays that are NOT uniform nationwide can be switched on or off

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	DTL	Date, which has to be evaluated
threeKings	Bool	Three Kings
roseMonday	Bool	Rose Monday
ascension	Bool	Ascension
corpusChristi	Bool	Corpus Christi
augsburgerFriedensfest	Bool	Augsburger Friedensfest
assumptionOfMary	Bool	Assumption Of Mary
reformationDay	Bool	Reformation Day
allSaintDay	Bool	All Saint Day
bussUndBetttag	Bool	Day of Prayer and Repentance (Buss und Betttag)

#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	If the date at the input parameter is a public holiday - returning TRUE, otherwise returning FALSE

#### Functional description

The block calculates the public holiday calendar of the year for a given date and displays whether the given date is a public holiday.

Optionally, holidays that are not uniform nationwide, such as Epiphany (Three Kings), can be taken into account via the appropriate input parameters in the block.

#### Change log

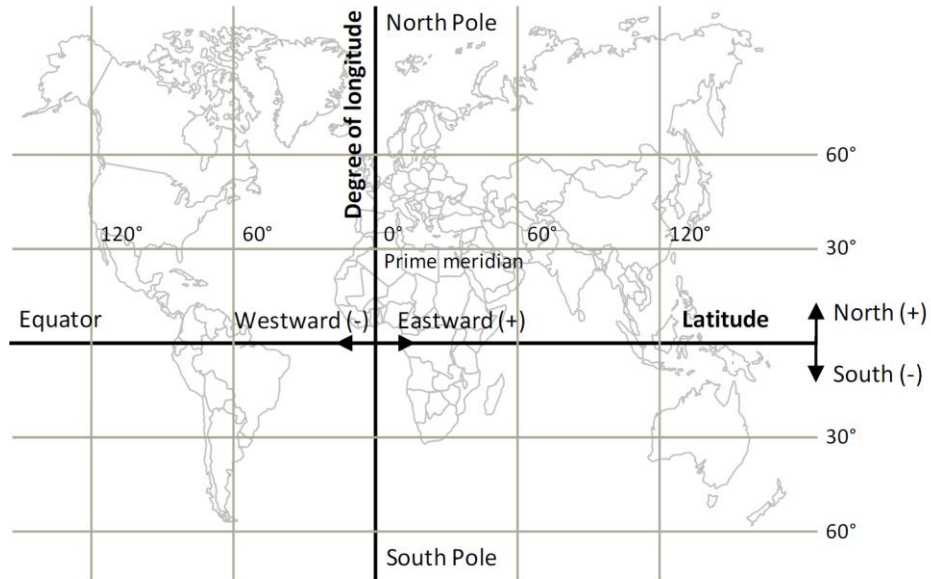
Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 17.07.2019	<b>Simatic Systems Support</b> Standard header, comments, style updated refactoring code
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> fix bug in Constant "DAYS_AFTER_EASTER_60" from 6 to 60 Insert documentation

### 4.2.5 LGF\_AstroClock (FB / V3.0.1)

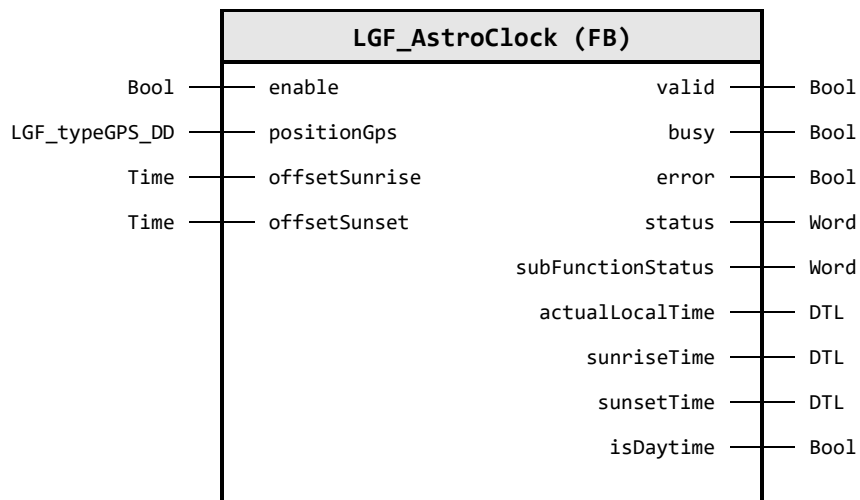
Author: Siemens Digital Industry

#### Short description

This function calculates the times of sunrise and sunset based on the local time for a specific place on Earth. The exact position is transferred in the form of geographical GPS coordinates (longitude and latitude).



#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
enable	Bool	FALSE	TRUE: Activates the functionality of the FB
positionGps	LGF_typeGPS_D D	---	GPS position to calculate the time of sunrise and sunset
offsetSunrise	Time	T#0s	Offset to sunrise (added to sunrise time, considered at `isDaytime`, negative time allowed)

#### 4 Program blocks

Identifier	Data type	Default value	Description
offsetSunset	Time	T#0s	Offset to sunset (added to sunset time, considered at `isDaytime`, negative time allowed)

#### Output parameter

Identifier	Data type	Description
valid	Bool	TRUE: Valid set of output values available at the FB
busy	Bool	TRUE: FB is active and new output values can be expected
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
actualLocalTime	DTL	Current time (local time)
sunriseTime	DTL	Sunrise time (localtime)
sunsetTime	DTL	Sunset time (localtime)
isDaytime	Bool	TRUE: If the local time of the controller is between "sunrise" and "sunset".

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED_NO_ERROR Execution finished without errors
16#7000	STATUS_NO_CALL No job being currently processed - not `enable`
16#7001	STATUS_IN_OPERATION Block is in Operation - Enabled
16#8204	ERR_LATITUDE_VALUE Error: Wrong Latitude DD value
16#8205	ERR_LONGITUDE_VALUE Error: Wrong Longitude DD value
16#8601	ERR_RD_SYS_T Error instruction RD_SYS_T, check `subFunctionStatus` code
16#8602	ERR_RD_LOC_T Error instruction RD_LOC_T, check `subFunctionStatus` code

**User defined datatype(s)**

**LGF\_typeGPS\_DD (UDT)**

Datatype for GPS Coordinates in decimal degrees.  
 For latitude and longitude.  
 Datatype for a whole GPS Data set.

Identifier	Data type	Default value	Description
latitude	Real	0.0	Degrees latitude with decimal places (Unit: degree decimal), North = positive; South = negative) valid value range [-90.00000..90.00000]
longitude	Real	0.0	Degrees longitude in degrees with decimal places (Unit: degree decimal), East = positive; West = negative) valid range [-180.0000..180.0000]

**Functional description**

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

If processes must run automatically depending on the change between day and night, the function of an astronomical clock is required. Examples of this would be switching outdoor lighting on and off or opening and closing roller shutters.  
 If these processes are to be executed with a time delay i.e. a defined time before or after sunrise or sunset an offset is required in each case.

**Note** For precise execution of the function, it must be ensured that system time and local time of the SIMATIC controller are set correctly.

Based on the system time/local time of the SIMATIC controller and the set coordinates, the block calculates the times for sunrise and sunset. The offset times are added to the sunrise and sunset and output on the `sunrise` and `sunset` outputs. If the systems local time of the SIMATIC controller is between these values, the output `isDaytime` is set to the value `TRUE`.

**Note** Since the times for sunrise and sunset change daily, it is possible that the `isDaytime` output remains permanently on `TRUE` or `FALSE` over a longer period of time:

- with correspondingly large offset values
- for a place on the other side of the Arctic Circle

The input of the GPS coordinate values is checked for valid values. If there are invalid values, an appropriate error code is output to `status`.  
 If there is an invalid coordinate value for a formal parameter, the outputs `sunrise` and `sunset` are set to the value `DTL#1970-01-01-00:00:00`.

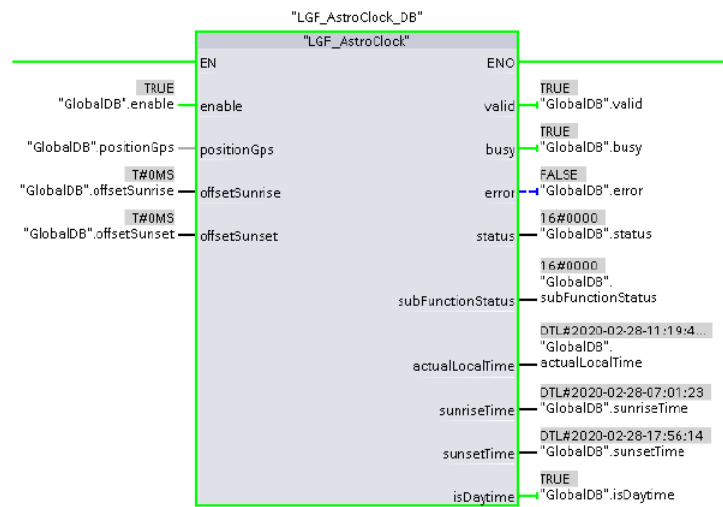
**Example**

The following example illustrates the block's functionality.

Geographical coordinates for Nuremberg-Moorenbrunn, date, and system time:

Description	Value
Longitude:	+ 11.07675°
Latitude:	+ 49.45203°
Date:	02/28/2020
Local time:	11:22:37
Sunrise:	07:01:23
Sunset:	17:56:14

Figure: FB LGF\_AstroClock, Observation of the block online with the parameters and the actual parameters via the Observation Table



Name	Anzeigeformat	Beobachtungswert
"GlobalDB".enable	BOOL	TRUE
"GlobalDB".positionGps.latitude	Gleitpunktzahl	49.45203
"GlobalDB".positionGps.longitude	Gleitpunktzahl	11.07675
"GlobalDB".offsetSunrise	Zeit	T#0MS
"GlobalDB".offsetSunset	Zeit	T#0MS
"GlobalDB".valid	BOOL	TRUE
"GlobalDB".busy	BOOL	TRUE
"GlobalDB".error	BOOL	FALSE
"GlobalDB".status	Hex	16#0000
"GlobalDB".subFunctionStatus	Hex	16#0000
"GlobalDB".actualLocalTime	DATE_AND_TIME	DTL#2020-02-28-11:22:37.220143296
"GlobalDB".sunriseTime	DATE_AND_TIME	DTL#2020-02-28-07:01:23
"GlobalDB".sunsetTime	DATE_AND_TIME	DTL#2020-02-28-17:56:14
"GlobalDB".isDaytime	BOOL	TRUE

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 01.10.2015	<b>Siemens Industry Online Support</b> T_ADD instruction is replaced with "+"
<b>01.00.02</b> 16.11.2015	<b>Siemens Industry Online Support</b> "offsetSunrise", "offsetSunset" is calculated in "daytime" Bug fix at "Adjust back TO UTC"
<b>01.01.00</b> 07.06.2015	<b>Siemens Industry Online Support</b> Add output actSystemTime and actLocalTime
<b>01.01.01</b> 15.06.2015	<b>Siemens Industry Online Support</b> Add comments
<b>01.01.02</b> 04.01.2017	<b>Siemens Industry Online Support</b> Bug fix at calculation sunrise and sunset
<b>01.01.03</b> 20.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.01.04</b> 22.02.2017	<b>Siemens Industry Online Support</b> Code optimization
<b>01.01.05</b> 09.07.2018	<b>Siemens Industry Online Support</b> Initialize #tempIntSunrise, #tempIntSunset, #tempDate1Jan
<b>01.01.06</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.07</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.08</b> 30.08.2019	<b>Simatic Systems Support</b> Rename from Astro to AstroClock Update Type name to positionGps - "LGF_typeGPS_DD" - GPS position as decimal degree Refactoring of interface - one input type for GPS data - refactored for better usability - refactoring of whole block to "ENABLE" behavior
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Bug fix - not enabled - block still running Insert documentation



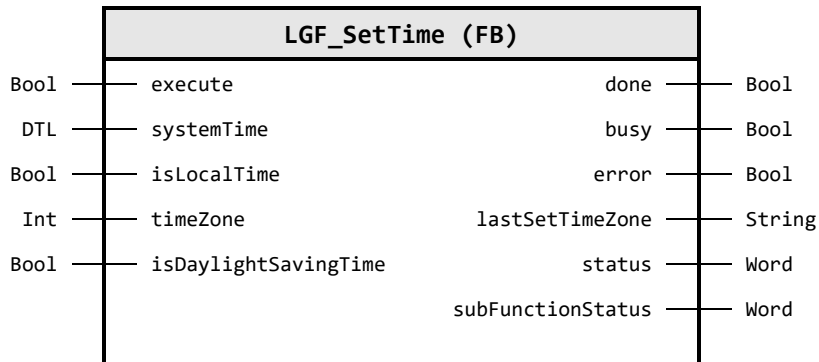
### 4.2.6 LGF\_SetTime (FB / V3.0.2)

Author: Siemens Digital Industry

#### Short description

This block combines the functions of system time, local time, and set time zone.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Rising edge starts action once
systemTime	DTL	---	System time to be set in PLC
isLocalTime	Bool	FALSE	TRUE: `systemTime` is local time, FALSE: `systemTime` is UTC time
timeZone	Int	0	Timezones HHMM [-1200.. -330.. 0.. 930.. 1200.. 1300]
isDaylightSavingTime	Bool	FALSE	Daylight saving time changeover, TRUE: activated, FALSE: deactivated (more infos at "Adjusting parameters in the `statTimeZone` variable")

#### Output parameter

Identifier	Data type	Description
done	Bool	TRUE: Commanded functionality has been completed successfully
busy	Bool	TRUE: FB is active and new output values can be expected
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
lastSetTimeZone	String	Time zone that was set last by this block
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED_NO_ERROR Execution finished without errors

## 4 Program blocks

Code / Value	Identifier / Description
16#7000	STATUS_NO_CALL No job being currently processed
16#7001	STATUS_FIRST_CALL First call after incoming new job (rising edge 'execute')
16#7002	STATUS_SUBSEQUENT_CALL Subsequent call during active processing without further details
16#8201	ERR_SET_TIME_LOCAL Error instruction WR_LOC_T: Write local time, check `subFunctionStatus` code
16#8202	ERR_SET_TIME_UTC Error instruction WR_SYS_T: Set time-of-day, check `subFunctionStatus` code
16#8203	ERR_SET_TIMEZONE Error instruction SET_TIMEZONE, check subFunctionStatus code
16#8600	ERR_UNDEFINED_STATE Error due to an undefined state in state machine
16#8601	ERR_WRONG_TIMEZONE Error due to an undefined time zone

### Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

This block combines the functions of system time, local time, and set time zone.

#### 4 Program blocks

The following time zones are possible on the `timeZone` input:

Input timeZone	Time zone
-1200	(UTC -12:00) Eniwetok, Kwajalein
-1100	(UTC -11:00) Midway Island
-1000	(UTC -10:00) Hawaii
-930	(UTC -09:30) (French) Polynesia
-900	(UTC -09:00) Alaska
-800	(UTC -08:00) Tijuana, Los Angeles, Seattle, Vancouver
-700	(UTC -07:00) Arizona, Denver, Salt Lake City, Calgary
-600	(UTC -06:00) Chicago, Dallas, Kansas City, Winnipeg
-500	(UTC -05:00) Eastern Time (USA & Canada)
-400	(UTC -04:00) La Paz, Georgetown
-330	(UTC -03:30) Newfoundland
-300	(UTC -03:00) Brasilia, Buenos Aires
-200	(UTC -02:00) Mid-Atlantic
-100	(UTC -01:00) Azores, Cape Verde Is.
0	(UTC) Dublin, Edinburgh, Lisbon, London
100	(UTC +01:00) Berlin, Bern, Brussels, Rome, Stockholm, Vienna
200	(UTC +02:00) Athens, Istanbul, Minsk, Bucharest
300	(UTC +03:00) Moscow, St. Petersburg, Baghdad, Kuwait, Riyadh
330	(UTC +03:30) Iran: Tehran
400	(UTC +04:00) Abu Dhabi, Muscat
430	(UTC +04:30) Afghanistan: Kabul
500	(UTC +05:00) Islamabad, Karachi, Tashkent
530	(UTC +05:30) India, Sri Lanka
545	(UTC +05:45) Nepal
600	(UTC +06:00) Astana, Almaty, Dhaka, Colombo
630	(UTC +06:30) Coco Island, Myanmar
700	(UTC +07:00) Bangkok, Hanoi, Jakarta
800	(UTC +08:00) Beijing, Chongqing, Hong Kong, Urumqi
830	(UTC +08:30) North Korea old
845	(UTC +08:45) Western Australia: Eucla
900	(UTC +09:00) Yakutsk, Osaka, Sapporo, Tokyo, Seoul
930	(UTC +09:30) Australia: Northern Territory, South Australia
1000	(UTC +10:00) Brisbane, Canberra, Melbourne, Sydney
1030	(UTC +10:30) Australia: Lord Howe Island
1100	(UTC +11:00) Vladivostok, Magadan, Solomon Is., New Caledonia
1200	(UTC +12:00) Auckland, Wellington
1245	(UTC +12:45) Chatham Islands
1300	(UTC +13:00) Tonga, Samoa
1400	(UTC +14:00) Kiribati

**Note**

**Daylight saving time/standard time**

The parameters (time difference, start summer time, start winter time) must be adapted to the desired time zone in the static variable `statTimeZone`.

**Adjusting parameters in the `statTimeZone` variable**

The static variable `statTimeZone` in the block interface is of the system data type `TimeTransformationRule`. In this system data type, the parameters for the local time zone and the summer/winter time changeover are stored.

The default values of the static variable `statTimeZone` are set to Central European Summer Time in the block interface:

- Time difference: 60 min
- Start summer time: last Sunday in March, 02:00 a.m.
- Start winter time: last Sunday in October, 03:00 a.m.

The following Figure shows the settings for the summer/winter time changeover of Central European Summer Time.

The parameter `Bias` is determined by the input parameter `timeZone`. The parameter `DaylightBias` depends on the input parameter `daylightSavingTime` and is either 0 or 60.

For other time zones, the parameters for summer/winter time changeover must be adjusted (marked below).

Name	Data type	Default value
Static		
statTimeZone	TimeTransformationRule	
Bias	Int	0
DaylightBias	Int	60
DaylightStartMonth	USInt	3
DaylightStartWeek	USInt	5
DaylightStartWeekday	USInt	1
DaylightStartHour	USInt	2
DaylightStartMinute	USInt	0
StandardStartMonth	USInt	10
StandardStartWeek	USInt	5
StandardStartWeekday	USInt	1
StandardStartHour	USInt	3
StandardStartMinute	USInt	0
TimeZoneName	String[80]	'not even set ...

## Change log

Version & Date	Change description
<b>01.00.00</b> 08.06.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.00.02</b> 02.03.2017	<b>Siemens Industry Online Support</b> Bugfix: FB number: automatic
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.05</b> 20.02.2019	<b>Siemens Industry Online Support</b> Bugfix: Rising edge at input REQ of SET_TIMEOUT
<b>01.00.06</b> 23.08.2019	<b>Simatic Systems Support</b> Reworked interface to PLC Open "execute" behavior Magic numbers removed, tag naming added, code reworked
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.02</b> 13.01.2020	<b>Simatic Systems Support</b> Bug fix - bias correction for time offsets (200 / 330) Insert documentation

## 4.2.7 LGF\_TimerSwitch (FB / V3.0.1)

Author: Siemens Digital Industry

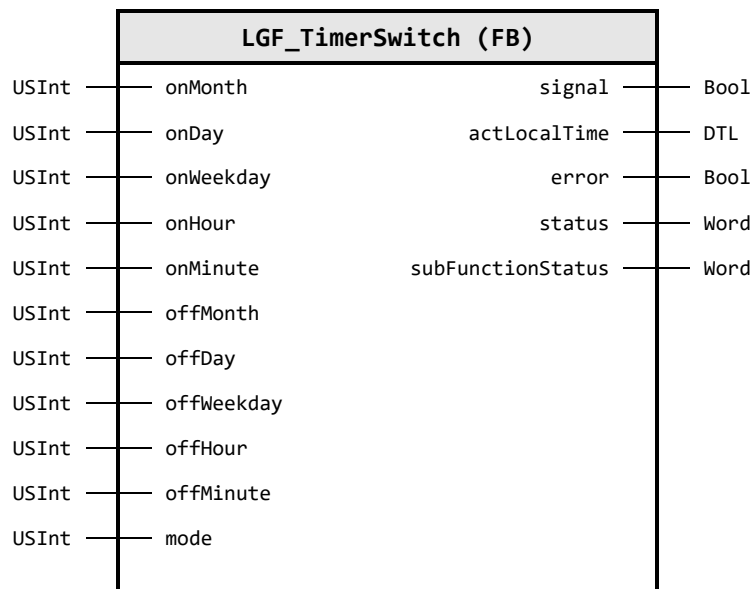
### Short description

This block is a timer. It is possible to define daily, weekly, monthly, yearly time switch points and time switch points for working days or weekend days.

Mode: Daily: 1, weekly: 2, monthly: 3, yearly: 4, workday: 5, weekend: 6

The time value is always compared with the local time of the PLC, therefore the time value specified at the On and Off parameters must be specified as local time.

### Block Interface



### Input parameter

Identifier	Data type	Default value	Description
onMonth	USInt	0	Month, in which the signal shall be set.
onDay	USInt	0	Day, at which the signal shall be set.
onWeekday	USInt	0	Day of the week on which the signal will be set; Sunday: 1, Monday: 2, Tuesday: 3, ...
onHour	USInt	0	Hour, at which the signal shall be set.
onMinute	USInt	0	Minute, at which the signal shall be set.
offMonth	USInt	0	Month, in which the signal shall be reset.
offDay	USInt	0	Day, at which the signal shall be reset.
offWeekday	USInt	0	Day of the week on which the signal will be reset; Sunday: 1, Monday: 2, Tuesday: 3, ...
offHour	USInt	0	Hour, at which the signal shall be reset.
offMinute	USInt	0	Minute, at which the signal shall be reset.
mode	USInt	0	Specifies the mode (see Principle of operation); Daily: 1, weekly: 2, monthly: 3, yearly: 4, workday: 5, weekend: 6

**Output parameter**

Identifier	Data type	Description
signal	Bool	Output signal
actLocalTime	DTL	Current local time
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Sub function status code

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8200	ERR_NO_MODE_SELECTED Error: No suitable mode is selected, check input "mode"
16#8600	ERR_RD_LOC_T Error in function RD_LOC_T, see in `subFunctionStatus`

**Functional description****Note**

The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The block offers various timer types, which are determined in the `mode` parameter:

- Daily timer (mode = 1)
- Weekly timer (mode = 2)
- Monthly timer (mode = 3)
- Yearly timer (mode = 4)
- Weekdays, Monday to Friday (mode = 5)
- Weekend, Saturday and Sunday (mode = 6)

The time value is always compared with the local time of the PLC, therefore the time value specified at the On and Off parameters must be specified as local time.

Depending on the mode, the following formal parameters must be interconnected:

Mode	Mode	Required formal parameters
1 .	Daily timer	- onHour / offHour - onMinute / offMinute
2 .	Weekly timer	- onWeekday / offWeekday - onHour / offHour - onMinute / offMinute
3 .	Monthly timer	- onDay / offDay - onHour / offHour - onMinute / offMinute

## 4 Program blocks

Mode	Mode	Required formal parameters
4 .	Yearly timer	<ul style="list-style-type: none"> <li>- onMonth / offMonth</li> <li>- onDay / offDay</li> <li>- onHour / offHour</li> <li>- onMinute / offMinute</li> </ul>
5 .	Weekdays	<ul style="list-style-type: none"> <li>- onHour / offHour</li> <li>- onMinute / offMinute</li> </ul>
6 .	Weekend	<ul style="list-style-type: none"> <li>- onHour / offHour</li> <li>- onMinute / offMinute</li> </ul>

If the set start time equals the current local time of the controller, the output `signal` is set to `TRUE`. If the set switch-off time equals the current local time of the controller, the `signal` output is reset again.

### Note

Please note that the block can be used in the “Monthly timer” modes (mode = 3) or “yearly timer” (mode = 4) the block only switches if the days that you specify at the input parameters, “onDay” and “offDay”, actually occur in this month.

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 16.11.2015	<b>Siemens Industry Online Support</b> Fix in mode 2
<b>01.01.00</b> 23.05.2016	<b>Siemens Industry Online Support</b> New mode 5 + 6 New output: actLocalTime
<b>01.01.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.01.02</b> 14.09.2018	<b>Siemens Industry Online Support</b> Fix in modes 1, 3, 5, 6
<b>01.01.03</b> 17.09.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.04</b> 10.10.2018	<b>Siemens Industry Online Support</b> Connection to type restored
<b>01.01.05</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.10</b> 13.11.2019	<b>Simatic Systems Support</b> Magic numbers removed, tag naming added, code reworked
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation



## 4.3 Counter operations

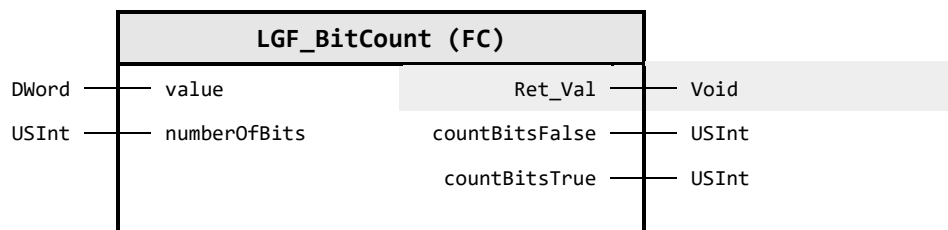
### 4.3.1 LGF\_BitCount (FC / V3.0.2)

Author: SiemensSIMATICSystemsSupport

#### Short description

This block counts in a variable of type DWord how many bits are set (TRUE) and how many are not set (FALSE) and outputs the number at the outputs. Instead of DWord, Word and Byte can also be used by converting the past parameter with e.g. BYTE\_TO\_DWORD and connecting the corresponding bit length of the data type at the parameter "numberOfBits".  
 Byte=8, Word=16, DWord=32

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	DWord	Tag where the bit states has to be counted
numberOfBits	USInt	Number of bits in input tag "value" (bit size of Datatype), in case of Byte=8, Word=16, DWord=32

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
countBitsFalse	USInt	Number of bits are FALSE in input tag "value"
countBitsTrue	USInt	Number of bits are TRUE in input tag "value"

#### Change log

Version & Date	Change description
<b>01.00.00</b> 06.06.2015	<b>Siemens Industry Support</b> first release
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.02</b> 19.01.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.3.2 LGF\_CountBooleanEdges (FB / V1.0.0)

Author: Siemens Simatic Systems Support

#### Short description

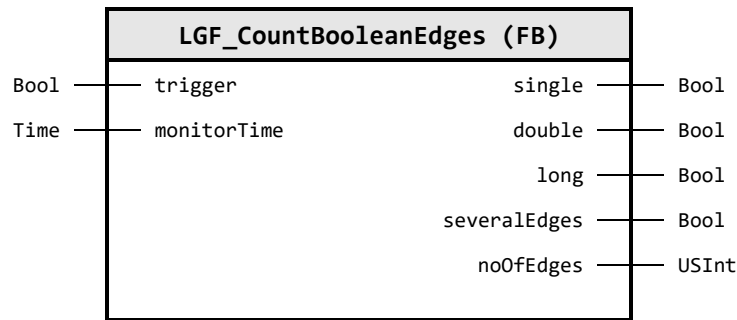
This function evaluates a input signal for different states in a certain amount of time.

The states are:

- One edge and input present over the whole monitoring time
- Single edge
- Double edge
- N-Edges in between the monitoring time

The Output signal is present for at least on cycle after the monitoring time has expired, or as long as the input `trigger` remains `TRUE`.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
trigger	Bool	FALSE	Trigger to evaluate signal signal (rising edges)
monitorTime	Time	T#1s	Time to monitor fand count edges on `trigger` input

#### Output parameter

Identifier	Data type	Description
single	Bool	Single edge until monitoring time expires
double	Bool	Two edges in between the monitoring time
long	Bool	Just a single edge in the monitoring time, the `trigger` input stays TRUE after the edge appears
severalEdges	Bool	Numeros Edges occure within the monitoring time, see `noOfEdges` to get the number of edges
noOfEdges	USInt	Number of edges in between the monitoring time frame

#### Change log

Version & Date	Change description
01.00.00 19.01.2021	Simatic Systems Support First released version

### 4.3.3 LGF\_CountFallInDWord (FB / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

The function analyzes a variable of the type DWord and outputs how often a 1-0 sequence (falling edge) occurs in the variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
value	DWord	16#00000000	Input Double word in which the falling edges are counted

#### Output parameter

Identifier	Data type	Description
numberOfEdges	Int	Number of falling edges in the DWord

#### Functional description

In a variable of the data type DWord, the block counts the falling edges (1-0 transitions) from left to right. The output `numberOfEdges` outputs the number of falling edges.

So that falling edges at the variable limit are also detected, the input `value` is copied to the static variable `statDWordPrevCycle` at the end of the evaluation and evaluated in the next cycle.

#### Example

The following example illustrates the block's functionality. In this case, it is assumed that a signal of unknown length is continuously sampled in the form of double words (DWord) per cycle.

Within this signal, the 1-0 sequences (falling edges) must be counted and output continuously.

Table: Example

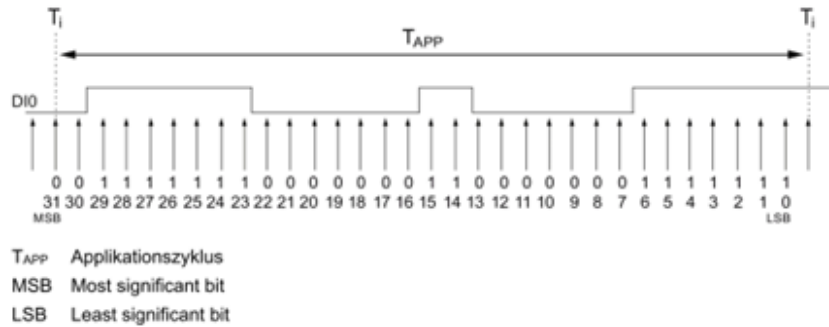
DWord previous cycle <code>statDWordPrevCycle</code>	DWord actual cycle <code>value</code>
1001_0000_0001_1010_1001_0000_0001_1011	0010_1010_0001_1111_0100_0011_1000_0101

Number of 1-0 sequences (falling edges): `Ret_Val1= 8`

### Application example

Excerpt from the manual of the technology module TM Timer DIDQ 16x24V:  
 With the oversampling function, the technology module records the state of the respective digital input per application cycle (e.g. OB61) at 32 points in time with a uniform time interval. The 32 states are jointly returned as 32-bit values in the check back interface.

Figure: Example of an oversampling of DI0 on TM Timer DIDQ 16x24V



The LGF\_CountFallInDWord block is used, in this case, to count how often a falling edge occurs.

SIMATIC ET 200MP/S7-1500 Technology Module TM Timer DIDQ 16x24V (6ES7552-1AA00-0AB0)

<https://support.industry.siemens.com/cs/ww/en/view/95153313>

### Change log

Version & Date	Change description
<b>01.00.00</b> 16.01.2019	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 16.12.2019	<b>Simatic Systems Support</b> Code refactoring - minimize used code memory
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.3.4 LGF\_CountRisInDWord (FB / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

The function analyzes a variable of the type DWord and outputs how often a 0-1 sequence (rising edge) occurs in the variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
value	DWord	16#00000000	Input Double word in which the rising edges are counted

#### Output parameter

Identifier	Data type	Description
numberOfEdges	Int	Number of rising edges in the DWord

#### Functional description

In a variable of the data type DWORD, the block counts the rising edges (0-1 transitions) from left to right. The output `countRisInDWord` outputs the number of rising edges.

So that rising edges at the variable limit are also detected, the input `value` is copied to the static variable `statDWordPrevCycle` at the end of the evaluation and evaluated in the next cycle.

#### Example

The following example illustrates the block's functionality. In this case, it is assumed that a signal of unknown length is continuously sampled in the form of double words (DWORD) per cycle.

Within this signal, the 0-1 sequences (rising edges) must be counted and output continuously.

Table: Example

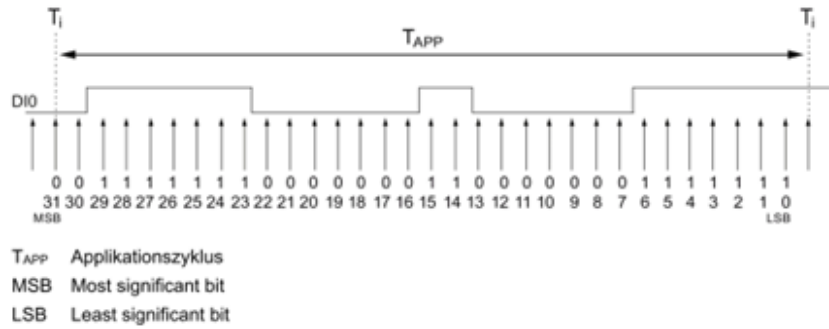
DWord previous cycle <code>statDWordPrevCycle</code>	DWord actual cycle <code>value</code>
1001_0000_0001_1010_1001_0000_0001_1010	1010_1010_0001_1111_0100_0011_1000_0101

Number of 0-1 sequences (rising edges): `Ret_Val` = 9

**Application example:**

Excerpt from the manual of the technology module TM Timer DIDQ 16x24V:  
 With the oversampling function, the technology module records the state of the respective digital input per application cycle (e.g. OB61) at 32 points in time with a uniform time interval. The 32 states are jointly returned as 32-bit values in the checkback interface.

Figure: Example of an oversampling of DI0 on TM Timer DIDQ 16x24V



The block LGF\_CountRisInDWordFB is used in this case to count how often a rising edge occurs.

SIMATIC ET 200MP/S7-1500 Technology Module TM Timer DIDQ 16x24V (6ES7552-1AA00-0AB0)

<https://support.industry.siemens.com/cs/ww/en/view/95153313>

**Change log**

Version & Date	Change description
<b>01.00.00</b> 16.01.2019	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 16.12.2019	<b>Simatic Systems Support</b> Code refactoring - minimize used code memory
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

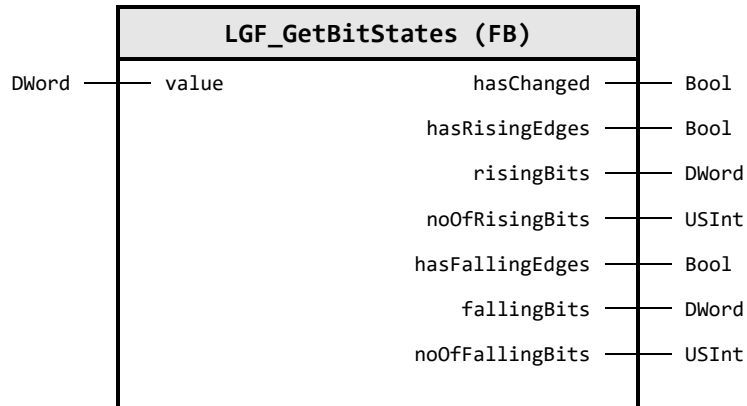
### 4.3.5 LGF\_GetBitStates (FB / V1.0.0)

Author: Siemens Simatic Systems Support

#### Short description

This function checks a DWord for falling as well as rising edges. It returns the number of edges, a DWord with the edge bits, and a boolean value if edge(s) are present.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
value	DWord	16#00000000	Check input value for changes and edges

#### Output parameter

Identifier	Data type	Description
hasChanged	Bool	Input value has changed (compared to the previous cycle)
hasRisingEdges	Bool	Input value has rising edges
risingBits	DWord	Bitstream with the rising edges
noOfRisingBits	USInt	Number of rising edges in the input value
hasFallingEdges	Bool	Input value has falling edges
fallingBits	DWord	Bitstream with the falling edges
noOfFallingBits	USInt	Number of falling edges in the input value

#### Change log

Version & Date	Change description
<b>01.00.00</b> 2021.01.28	<b>Simatic Systems Support</b> First released version

## 4.4 Comperator operations

### 4.4.1 LGF\_BinaryMaskCompare (FC / V1.0.0)

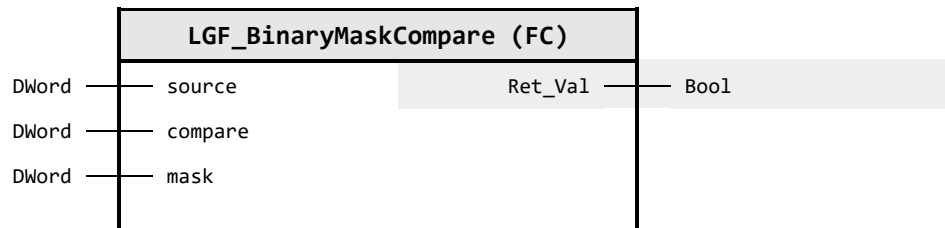
Author: Siemens Simatic Systems Support

#### Short description

This function compares two binary Values `source` and `compare` by a given `mask`. Both given values are masked (input AND `mask`), and the results is than compared and returned.

Can be used for Word and Byte as well, by convert the passed parameter using for e.g. `Byte_to_DWord(...)`.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
source	DWord	Source value to compare
compare	DWord	Value to compare against
mask	DWord	Mask the data - bits will pass if TRUE or block if FALSE

#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	Return TRUE if masked values are equal

#### Change log

Version & Date	Change description
01.00.00 19.01.2021	<b>Simatic Systems Support</b> First released version



#### 4.4.2 LGF\_CompareLReal (FC / V3.0.1)

Author: Siemens Digital Industry

##### Short description

This function checks floating point numbers for equality, by using an approximation formula and a fixed precision by constant 1.0E-12 (pico)

##### Block Interface



##### Input parameter

Identifier	Data type	Description
valueA	LReal	First LREAL number to be compared.
valueB	LReal	Second LREAL number to be compared.

##### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	FALSE: not equal TRUE: approximately the same

##### Functional description

The comparison of the LREAL numbers is based on a fixed accuracy of 1.0E-12. The difference between the two input values must be smaller than the **PRECISION** accuracy multiplied by one of the two input values.

Equation:

$$result := ABS(valueA - valueB) \leq (PRECISION * ABS(valueA))$$

##### Note

If your application requires a different accuracy when comparing the numbers, adapt the "PRECISION" constant in the function to your requirements.

Or you may use the FC LGF\_CompareLRealByPrecision.

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 13.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 03.06.2019	<b>Simatic Systems Support</b> Refactoring and performance improvement Delete Error and Status there is no need for, because of changed / adjusted algorithm add eno handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation

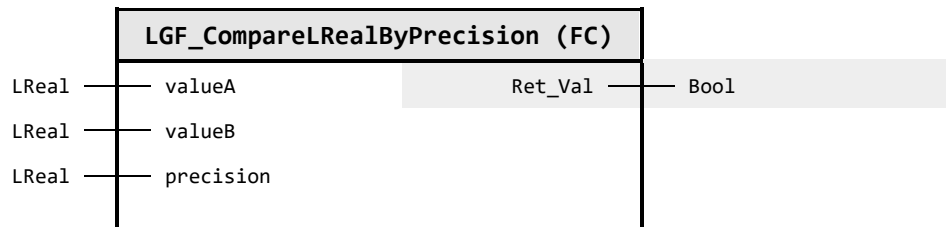
### 4.4.3 LGF\_CompareLRealByPrecision (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function checks floating point numbers for equality, by using an approximation formula and a fixed precision by constant 1.0E-12 (pico)

#### Block Interface



#### Input parameter

Identifier	Data type	Description
valueA	LReal	First LREAL number to be compared.
valueB	LReal	Second LREAL number to be compared.
precision	LReal	Accuracy with which the two values are compared.

#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	FALSE: not equal TRUE: approximately the same

#### Functional description

The comparison of the LREAL numbers is based on an given accuracy at the parameter `precision`. The difference between the two input values must be smaller than the `precision` accuracy value multiplied by one of the two input values.

Equation:

$$result := ABS(valueA - valueB) \leq (precision * ABS(valueA))$$

#### Change log

Version & Date	Change description
<b>01.00.00</b> 03.06.2019	<b>Simatic Systems Support</b> First released version function based on `LGF_CompareLReal`
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.4.4 LGF\_CompareVariant (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

The function compares two structured actual parameters (array, PLC data type) and outputs whether they are of the same type and have the same values.

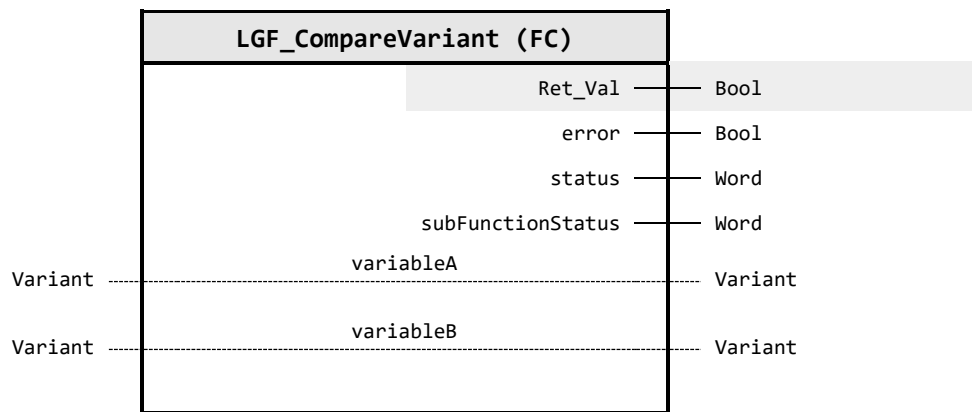
Compare arrays or plc datatypes and their values up to a max length of 200 Bytes of the connected variables. If at least one value of an element is not identical → set function result = false

Restrictions:

The attached structure must not include Strings

The attached structure can not exceed 200 bytes, because of the internal buffer size

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	FALSE: Values of comparison variables or PLC data types are different. TRUE: Values of the comparison variables are equal and PLC data types are identical.
error	Bool	FALSE: No error / TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB / 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### In/Out parameter

Identifier	Data type	Description
variableA	Variant	First comparison variable with any data type
variableB	Variant	Second comparison variable with any data type

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors

## 4 Program blocks

Code / Value	Identifier / Description
16#8201	ERR_INPUT_TYPES_MUST_MATCH Error, the input types must match, e.g. STRUCT
16#8202	ERR_INPUT_TYPES_LENGTH_NOT_EQUAL Error, the input types have different lengths after serialization, `subFunctionStatus` provides an indicator of the different size
16#8601	ERR_SERIALIZE_VARIABLE_A Error occurred while serialize variableA into Bytearray - see `subFunctionStatus` for detailed information
16#8602	ERR_SERIALIZE_VARIABLE_B Error occurred while serialize variableB into Bytearray - see `subFunctionStatus` for detailed information

### Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

This block compares two (structured) actual parameters and shows whether they equate to the same value.

**Note** The following differences cannot be detected with the comparison method (byte level):

- Variables of the data type `Struct` cannot be compared.
- For strings, there may be differences in the range between the actual length and the maximum length.
- With `REAL` numbers in the structure, a disparity can also be displayed for "same" variables.
- Variables of the type `ARRAY` of `BOOL` cannot be checked for equality with the function, because the command used, `CountOfElements`, also counts the filling elements (e.g. 8 is returned with an `ARRAY[0..1]` of `BOOL`).

## Change log

Version & Date	Change description
<b>01.00.00</b> 03.09.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 11.02.2015	<b>Siemens Industry Online Support</b> Bug fix
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.05</b> 03.06.2019	<b>Simatic Systems Support</b> Refactoring and performance improvement Change error handling to status and subFctStatus update serialize instruction add eno handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 19.01.2020	<b>Simatic Systems Support</b> Insert documentation

## 4.5 Math operations

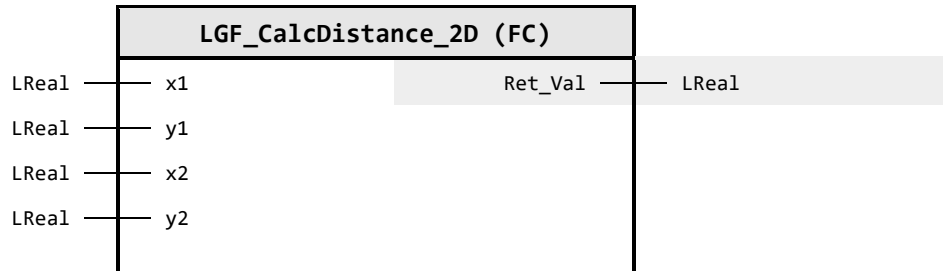
### 4.5.1 LGF\_CalcDistance\_2D (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

The function calculates the distance between two points in the plane.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
x1	LReal	X coordinate point 1
y1	LReal	Y coordinate point 1
x2	LReal	X coordinate point 2
y2	LReal	Y coordinate point 2

#### Output parameter

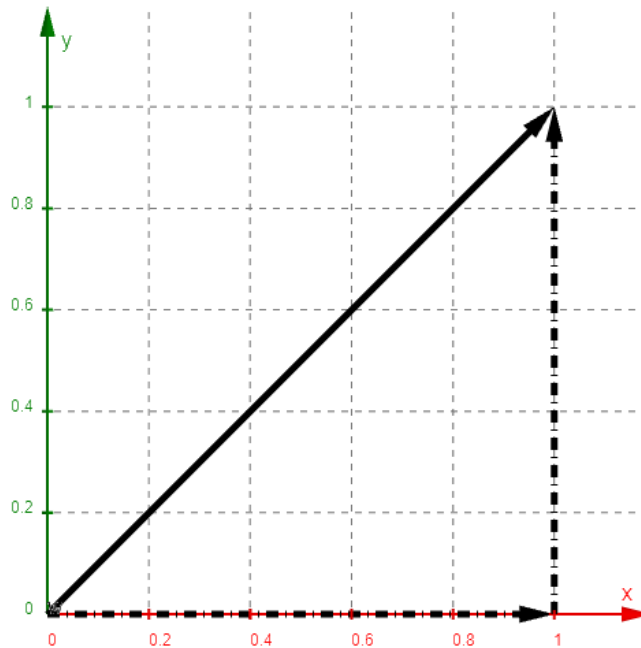
Identifier	Data type	Description
Ret_Val	LReal	Calculated distance between the Points

#### Functional description

The block calculates the distance between two points in a Cartesian coordinate system. The distance is calculated with the following formula:

$$result = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Figure: Graphical representation



### Change log

Version & Date	Change description
<b>01.00.00</b> 06.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.02.00</b> 04.10.2019	<b>Simatic Systems Support</b> renamed from "Distance" to "CalcDistance_2D" Data type changed to LREAL Data type changed to LREAL
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation



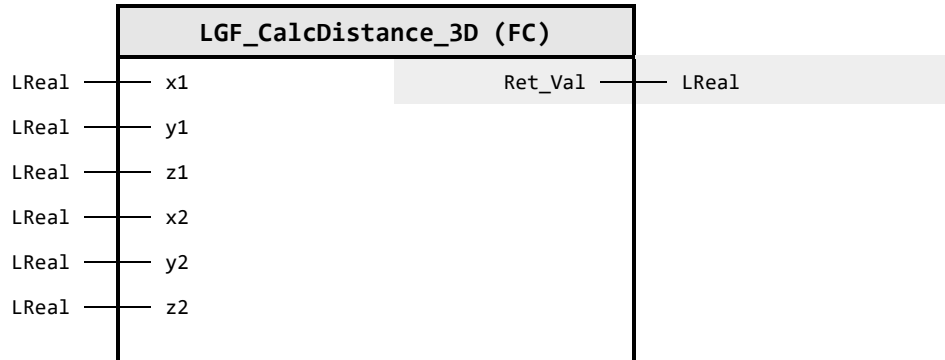
### 4.5.2 LGF\_CalcDistance\_3D (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

The function calculates the distance between two points in 3D space.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
x1	LReal	X coordinate point 1
y1	LReal	Y coordinate point 1
z1	LReal	Z coordinate point 1
x2	LReal	X coordinate point 2
y2	LReal	Y coordinate point 2
z2	LReal	Z coordinate point 2

#### Output parameter

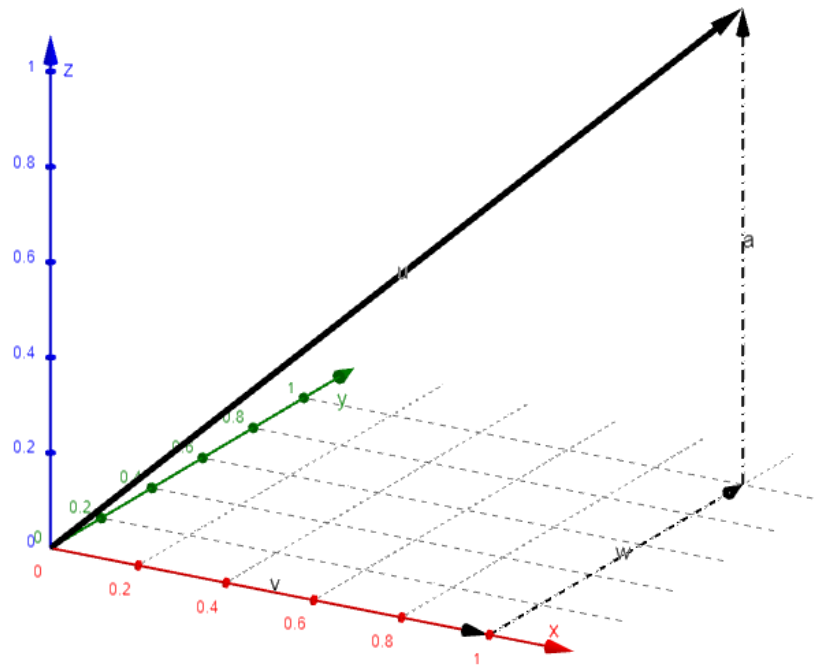
Identifier	Data type	Description
Ret_Val	LReal	Calculated distance between the Points

#### Functional description

The block calculates the distance between two points in a Cartesian coordinate system. The distance is calculated with the following formula:

$$result = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Figure: Graphical representation



**Change log**

Version & Date	Change description
<b>01.00.00</b> 04.10.2019	<b>Siemens Industry Presales Support</b> First released version derivate from "CalcDistance_2D" and extended to 3D
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.5.3 LGF\_GetFactorial (FC / V3.0.1)

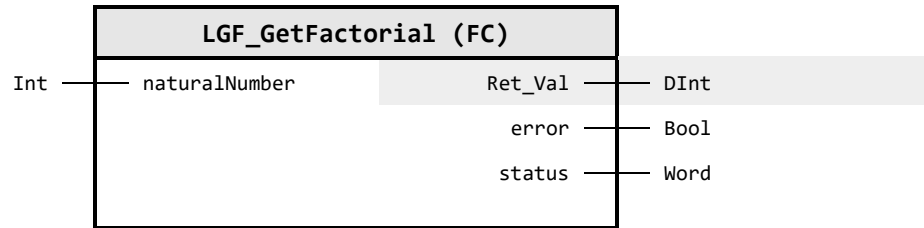
Author: Siemens Digital Industry

#### Short description

The function calculates the faculty of a natural number (N!) and returns the result.

The permissible value range of the input parameter `naturalNumber` is between 0 and 12, as 12 is the maximum factorial result fitting into a DInt type

#### Block Interface



#### Input parameter

Identifier	Data type	Description
naturalNumber	Int	Natural number (0..12)

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Calculated factorial
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8101	ERR_WRONG_VALUE_RANGE Error: Input value out of range Permissible value range is 0..12

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 23.09.2019	<b>Siemens Industry Online Support</b> Renamed from "Factorial" to "GetFactorial" Code refactoring, regions and more comments added Reworked to case of, MAGIC numbers are okay as they stay for the number/case itself
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

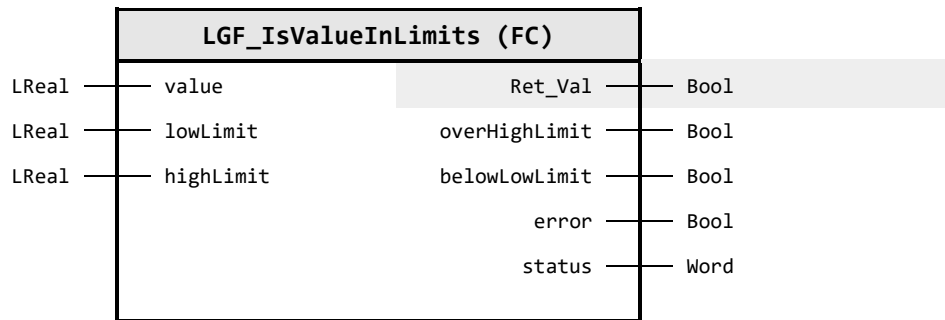
### 4.5.4 LGF\_IsValueInLimits (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

The function checks whether a value is within a defined value range. The value range is defined with a lower and an upper limit.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	LReal	Value to be checked to determine whether it is within the defined value range
lowLimit	LReal	Low limit where the value is checked against to be greater
highLimit	LReal	High limit where the value is checked against to to be less

#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	Return: TRUE if the “value” is in the value range (range of the set point)
overHighLimit	Bool	TRUE if the “value” is greater than the upper limit value
belowLowLimit	Bool	TRUE, if the “value” is less than the lower limit value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

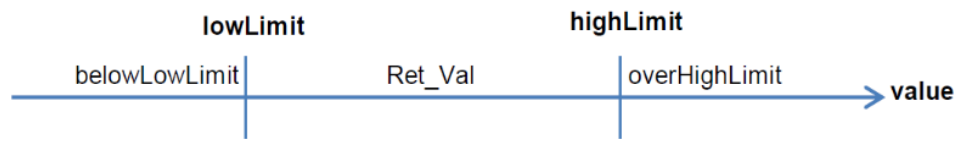
#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8401	ERR_RANGE_HIGH_BELOW_LOW_LIMIT Error:High limit less then low limit

#### Functional description

The variables `lowLimit` and `highLimit` define a value range. The function checks whether the `value` is below, in or above the value range. The outputs `belowLowLimit`, `Ret_Val`, or `overHighLimit` show where the `value` is located.

Figure: Principle of operation



**Change log**

Version & Date	Change description
<b>01.00.00</b> 10.12.2019	<b>Siemens Industry Support</b> First released version Copied from "IsValueInRange"
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

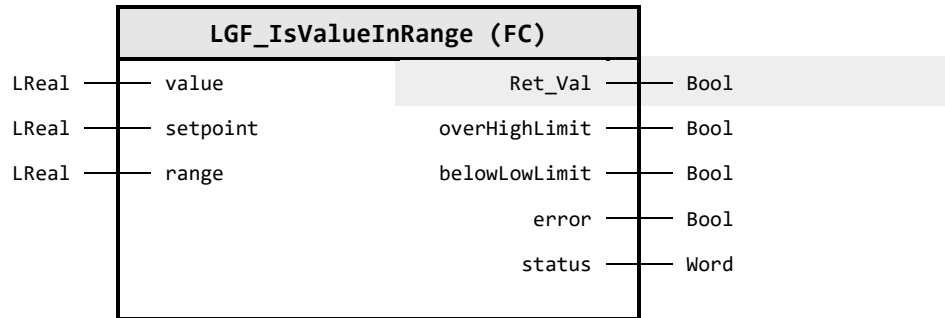
### 4.5.5 LGF\_IsValueInRange (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

The function checks whether a value is within a defined value range. The value range is defined with a set point and a range around this set point. The function calculates the low limit and high limit of the value range.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	LReal	Value to be checked to determine whether it is within the defined value range
setpoint	LReal	Set point
range	LReal	Area where the setpoint is in range

#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	Return: TRUE if the "value" is in the value range (range of the set point)
overHighLimit	Bool	TRUE if the "value" is greater than the upper limit value
belowLowLimit	Bool	TRUE, if the "value" is less than the lower limit value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

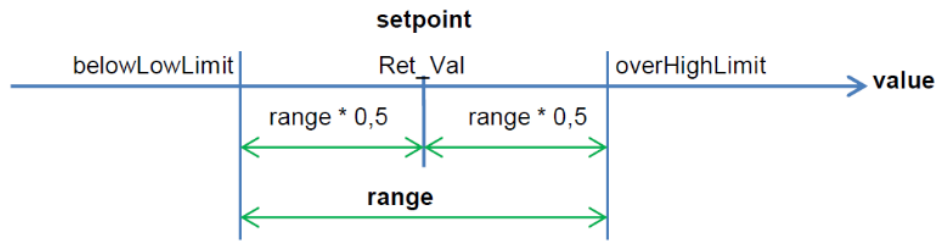
#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8401	ERR_RANGE_LIMIT_VALUES Error: During calculating of the limit values

#### Functional description

The `setpoint` and `range` variables define a range of values. The function checks whether the `value` is below, in or above the value range. The outputs `belowLowLimit`, `Ret_Val`, or `overHighLimit` show where the `value` is located.

Figure: Principle of operation



**Change log**

Version & Date	Change description
<b>01.00.00</b> 30.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 13.11.2019	<b>Simatic Systems Support</b> renamed from "LGF_HighLowLimit" to "LGF_IsValueInRange" Code refactoring error values changed, regions, comments and constant's are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation



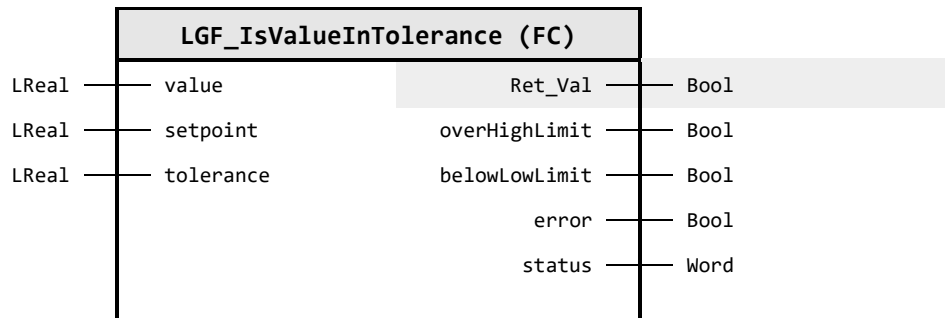
### 4.5.6 LGF\_IsValueInTolerance (FC / V3.0.2)

Author: Siemens Digital Industries

#### Short description

The function checks whether a value is within a defined value range. The value range is defined with a set point, as well as a tolerance range, around the set point in percent (%). The function calculates the low limit and high limit of the value range.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	LReal	Value to be checked to determine whether it is within the defined value range
setpoint	LReal	Set point
tolerance	LReal	Tolerance range around the set point in percent (%)

#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	Return: TRUE if the "value" is in the value range (range of the set point)
overHighLimit	Bool	TRUE if the "value" is greater than the upper limit value
belowLowLimit	Bool	TRUE, if the "value" is less than the lower limit value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: no error occurred
16#8401	ERR_RANGE_LIMIT_VALUES Error: wrong values during limit calculation for limit values

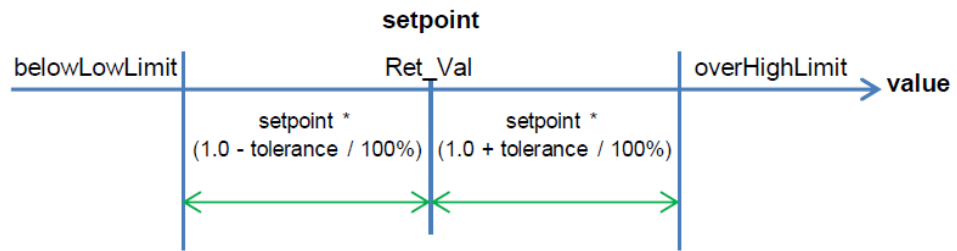
#### Functional description

The **setpoint** and **tolerance** percentage variables define a value range. The function checks whether the **value** is below, in or above the value range. The

## 4 Program blocks

outputs `belowLowLimit`, `Ret_Val`, or `overHighLimit` show where the `value` is located.

Figure: Principle of operation



### Change log

Version & Date	Change description
<b>01.00.00</b> 10.12.2019	<b>Siemens Industry Support</b> First released version Copied from "IsValueInRange"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.02</b> 12.11.2020	<b>Simatic Systems Support</b> Bug fix - negative setpoint verification Insert documentation

### 4.5.7 LGF\_NthRoot (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function extracts the n-th root of a given value.

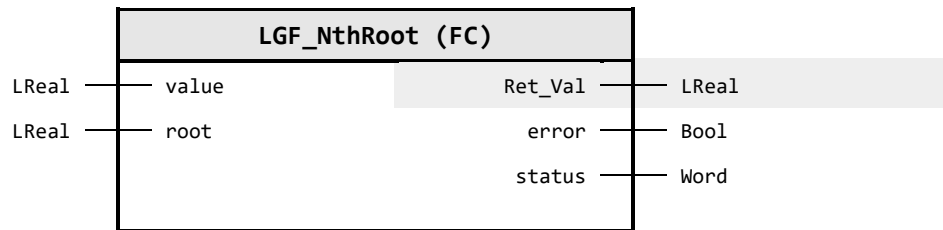
The root is defined as follows:

$$result = \sqrt[root]{value} = value^{\frac{1}{root}}$$

STEP 7 (TIA Portal) results in the following formula:

$$result = value ** (1/root)$$

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	LReal	Value from which the root should be calculated.
root	LReal	Exponent of root

#### Output parameter

Identifier	Data type	Description
Ret_Val	LReal	Returns the Nth root of a value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_NEG_VAR Error: Negative value for root exponend not permitted (Leads to complex numbers)

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.04</b> 17.01.2019	<b>Simatic Systems Support</b> Calculation changed
<b>01.00.09</b> 13.11.2019	<b>Simatic Systems Support</b> Renamed from "LGF_XRoot" to "LGF_NthRoot" Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

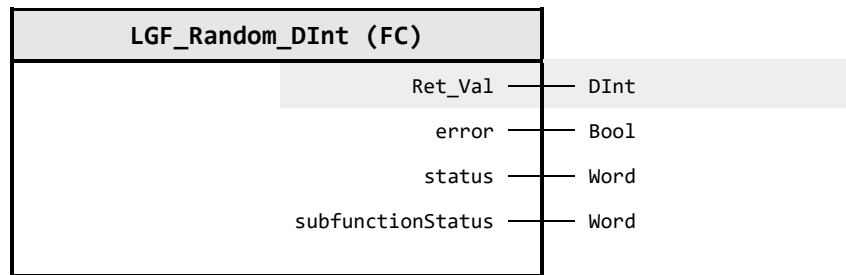
### 4.5.8 LGF\_Random\_DInt (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function generates a random value with each call. The random number has the data type DInt.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Random number in the DInt range
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Execution finished without errors
16#8600	ERR_RD_SYS_T Error in `RD_SYS_T` command - check `subFunctionStatus` code

#### Functional description

**Note** The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The function generates random values in the range:  
 $-2147483648 \leq ReturnVal \leq 2147483647$ .

The random value is formed from the nanoseconds of the current system time of the CPU. The byte order of this value is inverted and then converted to DInt.

### Change log

Version & Date	Change description
<b>01.00.00</b> 13.11.2019	<b>Siemens Industry Presales Support</b> First release copied from "LGF_Random_Real"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation

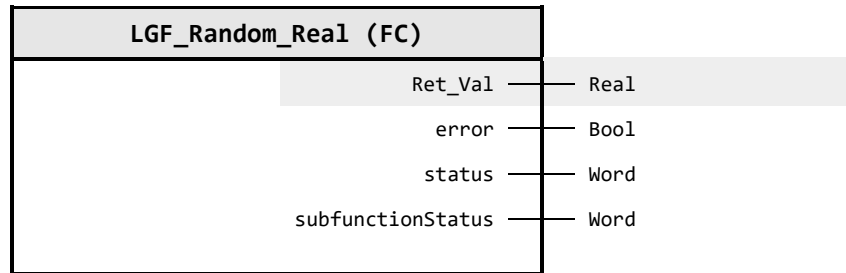
### 4.5.9 LGF\_Random\_Real (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function generates a random value with each call.  
The random number has the data type Real in the range from 0.0 to 1.0.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Random Real number between 0.0 and 1.0
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Execution finished without errors
16#8600	ERR_RD_SYS_T Error in `RD_SYS_T` command - check `subFunctionStatus` code

#### Functional description

**Note** The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The function generates random values in the range:  
 $0.0 \leq ReturnVal \leq 1.0$ .

The random value is formed from the nanoseconds of the current system time of the CPU. The byte order of this value is inverted and then converted to a floating point.

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 27.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 13.11.2019	<b>Simatic Systems Support</b> Renamed from "LGF_RandomBasic" to "LGF_Random_Real" Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation



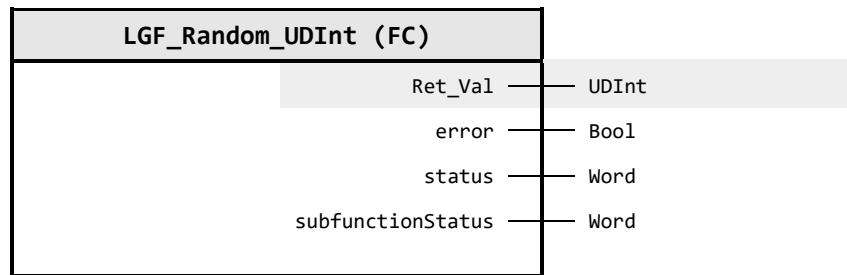
**4.5.10 LGF\_Random\_UDInt (FC / V3.0.1)**

Author: Siemens Digital Industries

**Short description**

This function generates a random value with each call.

The random number has the data type UDInt.

**Block Interface****Output parameter**

Identifier	Data type	Description
Ret_Val	UDInt	Random number in the UDInt range
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Execution finished without errors
16#8600	ERR_RD_SYS_T Error in `RD_SYS_T` command - check `subFunctionStatus` code

**Functional description**

**Note** The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The function generates random values in the range:

$$0 \leq ReturnVal \leq 4294967295.$$

The random value is formed from the nanoseconds of the current system time of the CPU. The byte order of this value is inverted and then converted to UDInt.

### Change log

Version & Date	Change description
<b>01.00.00</b> 11.12.2019	<b>Simatic Systems Support</b> First release copied from "LGF_Random_Real"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation

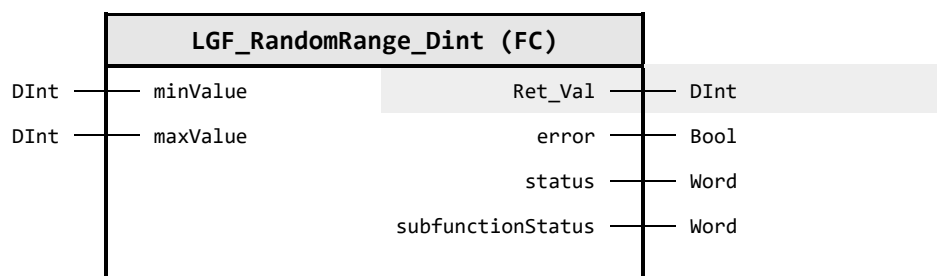
### 4.5.11 LGF\_RandomRange\_Dint (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function generates a random value in defined limits with each call. The random number has the data type DInt in the specified range.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
minValue	DInt	Minimum value of the range of the random number - lower border
maxValue	DInt	Maximum value of the range of the random number - upper border

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Random Real number in the predefined range
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Execution finished without errors
16#8200	ERR_MAX_LESS_MIN Error: The ranges specified are wrong: `minValue` is greater than `maxValue`
16#8600	ERR_RD_SYS_T Error in `RD_SYS_T` command - check `subFunctionStatus` code

#### Functional description

##### Note

The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

## 4 Program blocks

The block generates random values that are between the specified `minValue` and the `maxValue`. This random value is output via the `Ret_Val`.

The random value is formed from the nanoseconds of the current system time of the CPU. The byte order of this value is inverted and then converted to a `DInt`.

### Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 09.10.2019	<b>Simatic Systems Support</b> Renamed from "LGF_RandomInt" to "LGF_RandomRange_DInt" change random datatype from Int to DInt Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation

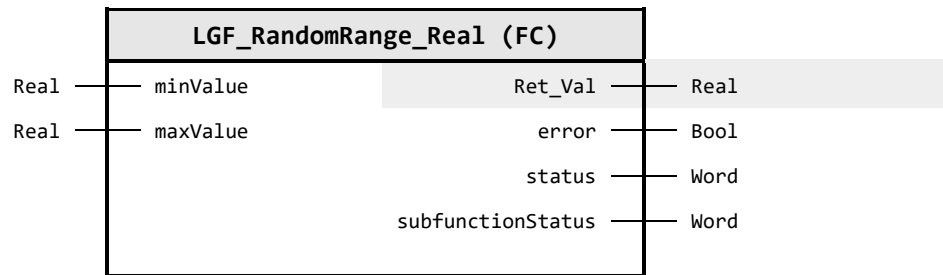
## 4.5.12 LGF\_RandomRange\_Real (FC / V3.0.1)

Author: Siemens Digital Industries

### Short description

This function generates a random value in defined limits with each call. The random number has the data type Real in the specified range.

### Block Interface



### Input parameter

Identifier	Data type	Description
minValue	Real	Minimum value of the range of the random number - lower border
maxValue	Real	Maximum value of the range of the random number - upper border

### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Random Real number in the predefined range
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Execution finished without errors
16#8200	ERR_MAX_LESS_MIN Error: The ranges specified are wrong: `minValue` is greater than `maxValue`
16#8600	ERR_RD_SYS_T Error in `RD_SYS_T` command - check `subFunctionStatus` code

### Functional description

#### Note

The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

## 4 Program blocks

The block generates random values that are between the specified `minValue` and the `maxValue`. This random value is output via the `Ret_Val`.

The random value is formed from the nanoseconds of the current system time of the CPU. The byte order of this value is inverted and then converted to a floating point.

### Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 02.03.2017	<b>Siemens Industry Online Support</b> Bugfix: FC number
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 13.11.2019	<b>Simatic Systems Support</b> Renamed from "LGF_RandomReal" to "LGF_RandomRange_Real" Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation

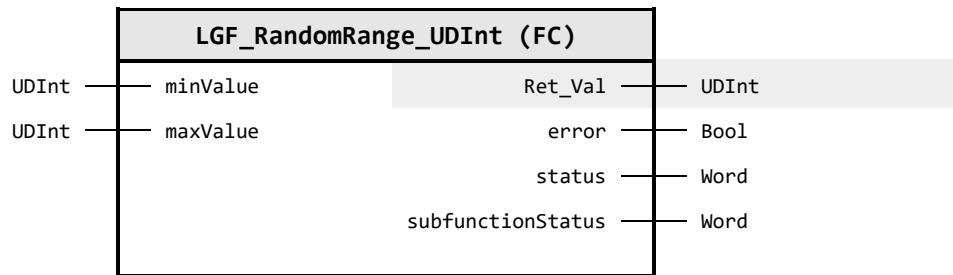
### 4.5.13 LGF\_RandomRange\_UDInt (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function generates a random value in defined limits with each call. The random number has the data type UDInt in the specified range.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
minValue	UDInt	Minimum value of the range of the random number - lower border
maxValue	UDInt	Maximum value of the range of the random number - upper border

#### Output parameter

Identifier	Data type	Description
Ret_Val	UDInt	Random UDInt number in the predefined range
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Execution finished without errors
16#8200	ERR_MAX_LESS_MIN Error: The ranges specified are wrong: `minValue` is greater than `maxValue`
16#8600	ERR_RD_SYS_T Error in `RD_SYS_T` command - check `subFunctionStatus` code

#### Functional description

##### Note

The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

## 4 Program blocks

---

The block generates random values that are between the specified `minValue` and the `maxValue`. This random value is output via the `Ret_Val`.

The random value is formed from the nanoseconds of the current system time of the CPU. The byte order of this value is inverted and then converted to a `UDInt`.

### Change log

Version & Date	Change description
<b>01.00.00</b> 11.12.2019	<b>Simatic Systems Support</b> First released version copied from "LGF_RandomRange_DInt"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation



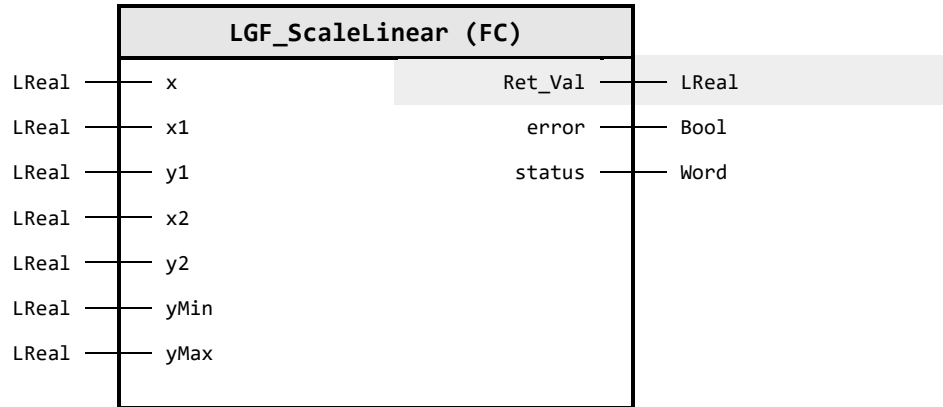
### 4.5.14 LGF\_ScaleLinear (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function scales an input variable (LReal) via a linear straight-line equation.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
x	LReal	Input value `x` to be scaled.
x1	LReal	Point 1 (P1) - `x` coordinate of the linear function.
y1	LReal	Point 1 (P1) - `y` coordinate of the linear function.
x2	LReal	Point 2 (P2) - `x` coordinate of the linear function.
y2	LReal	Point 2 (P2) - `y` coordinate of the linear function.
yMin	LReal	Lower limit value of the output.
yMax	LReal	High limit value of the output.

#### Output parameter

Identifier	Data type	Description
Ret_Val	LReal	Scaled output value `y`
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors.
16#6001	WARN_Y_LIMITED_TO_YMIN Warning: Output value limited to `yMin`
16#6002	WARN_Y_LIMITED_TO_YMAX Warning: Output value limited to `yMax`
16#8200	ERR_LOW_LIM_OVER_UP_LIM Error: Lower limit value `yMin` is greater than high limit value `yMax`.

### Functional description

The function linearly scales an input variable (e.g. an analog input value) to a specific output variable (e.g. level).

To determine the output variable, the following linear equation is used in the function:

$$x = \frac{y_2 - y_1}{x_2 - x_1} * (x - x_1) + y_1$$

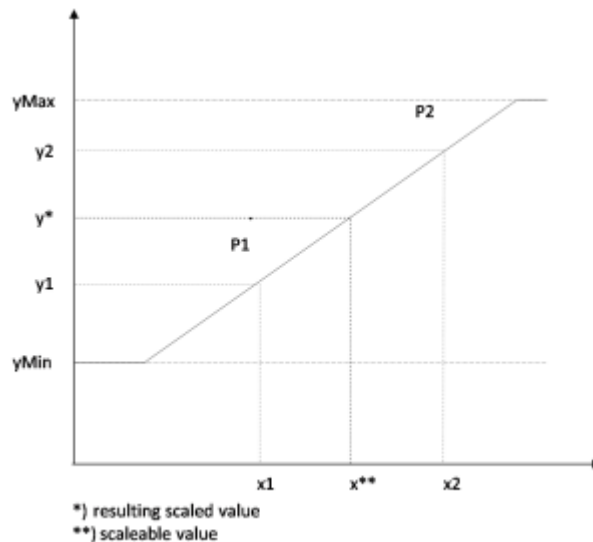
The straight line is described by the two points, P1 and P2. You specify the points as a Cartesian coordinate system using x and y coordinates.

#### Note

If the values of the parameters  $x_1$  and  $x_2$  are the same, the value of  $y_1$  is output on output  $y$ .

By specifying  $y_{Min}$  and  $y_{Max}$  you can restrict the calculated value of  $y$  to a range limited at top and bottom. Thus, you avoid override and underide ranges.

Figure: Graphical representation



### Example

A signal from 4 to 20mA is applied on an analog input module. This signal is converted to the CPU internal value from 0 to 27648 to measure a level. 0 corresponds to a level of 0.0m and 27648 to a level of 1.7m.

The block must then be parameterized as follows:

- P2:  $x_1 = 0$ ;  $y_1 = 0.0$
- P2:  $x_2 = +27648$ ;  $y_2 = 1.7$
- $y_{Min} = 0.0$ ;  $y_{Max} = 1.7$ ;

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 27.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 25.01.2019	<b>Simatic Systems Support</b> Data type changed from Variant to LReal
<b>02.00.01</b> 25.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update, status parameter added LReal value comparison added Result parameter changed to return value of FC for use in SCL Warning number changed to range of 16#6xxx refactor variable handling and extract returns in between the code add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Move to folder "Math operations"

### 4.5.15 LGF\_SearchMinMax (FC / V3.0.1)

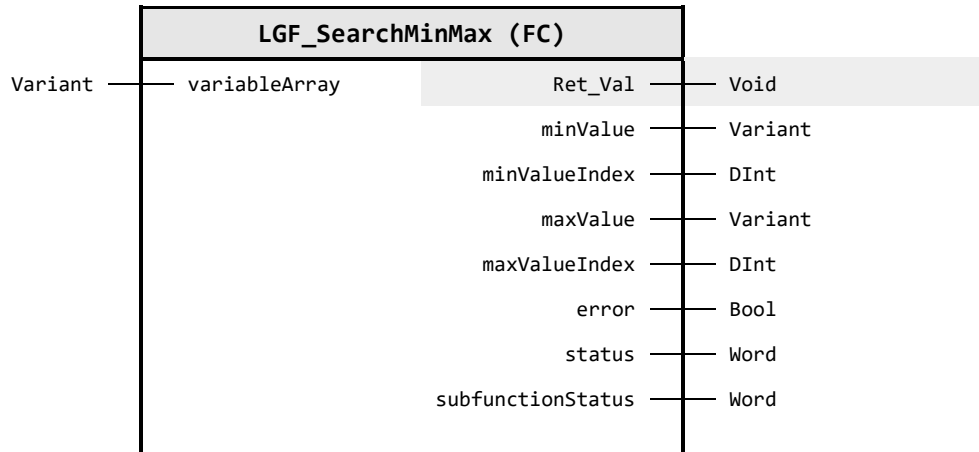
Author: Siemens Digital Industries

#### Short description

This function searches, in an array of the data type DInt, for the maximum and minimum value and the respective index in the array.

The following data types of the array elements are supported:  
Int, DInt, UInt, UDInt, USInt, SInt, and Real.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
variableArray	Variant	Array in whose fields the maximum and minimum are searched

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
minValue	Variant	Minimum value found in the array
minValueIndex	DInt	Index of the minimum found value in the array. The start index of the array plus `minValueIndex` results in the array index of the smallest value. The index starts with 0.
maxValue	Variant	Maximum value found in the array
maxValueIndex	DInt	Index of the maximum found value in the array. The start index of the array plus `maxValueIndex` results in the array index of the smallest value. The index starts with 0.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_NO_ARRAY Error: At input `variableArray` the actual parameter is not an array
16#8201	ERR_WRONG_TYPE Error: The data type of the elements in the array is not supported. Only the data types Int, UInt, DInt, UDInt, USInt, SInt and Real are supported.
16#8202	ERR_NOT_EQUAL_TYPES Error: The elements of the array do not have the same data type as the outputs `minValue` and `maxValue`
16#8203	ERR_MOVE_BLK_VARIANT Error: Subfunction `MOVE_BLK_VARIANT` executed with an error - check `subFunctionStatus` code

**Functional description**

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

An array of any size is connected via the `variableArray` input. After a data type query in the block, the elements are copied one after the other into a variable of the appropriate type and compared. The smallest and largest values, as well as their corresponding index are output to the array.

**Note** The following data types of the array elements are supported:  
Int, DInt, UInt, UDInt, USInt, SInt, and Real.

**Note** If there are several identical min. or max. values, the index of the first min. or max. value is output.

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 08.11.2019	<b>Siemens Industry Presales Support</b> Code refactoring, regions and more comments added
<b>03.00.00</b> 23.04.2020	<b>Siemens Industry Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 09.02.2021	<b>Simatic Systems Support</b> Rework constants and comments Insert documentation

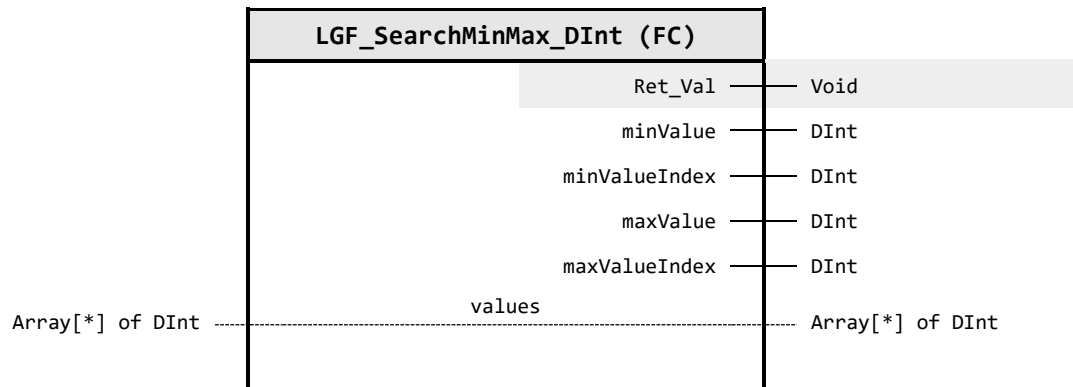
### 4.5.16 LGF\_SearchMinMax\_DInt (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function searches, in an array of the data type DInt, for the maximum and minimum value and the respective index in the array.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
minValue	DInt	Minimum value found in the array
minValueIndex	DInt	Index of the minimum found value in the array. The start index of the array plus `minValueIndex` results in the array index of the smallest value. The index starts with 0.
maxValue	DInt	Maximum value found in the array
maxValueIndex	DInt	Index of the maximum found value in the array. The start index of the array plus `maxValueIndex` results in the array index of the smallest value. The index starts with 0.

#### In/Out parameter

Identifier	Data type	Description
values	Array[*] of DInt	Array in whose fields the maximum and minimum are searched

#### Functional description

An array of any size is connected via the `values` input. The elements are then compared in turn. The smallest and largest values, as well as their corresponding index are output to the array.

#### Note

If there are several identical min. or max. values, the index of the first min. or max. value is output.

### Change log

Version & Date	Change description
<b>01.00.00</b> 11.11.2019	<b>Simatic Systems Support</b> First release copied from "LGF_SearchMinMax" and reworked to array[*]
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 09.02.2021	<b>Simatic Systems Support</b> Insert documentation



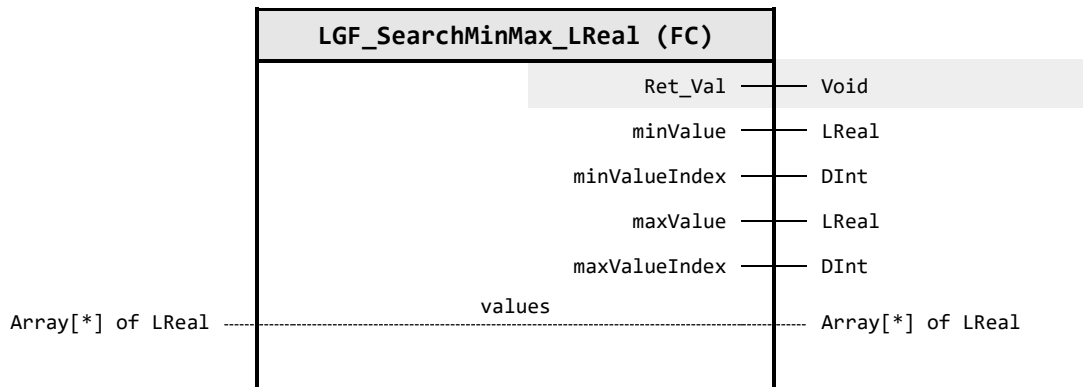
### 4.5.17 LGF\_SearchMinMax\_LReal (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function searches, in an array of the data type LReal, for the maximum and minimum value and the respective index in the array.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
minValue	LReal	Minimum value found in the array
minValueIndex	DInt	Index of the minimum found value in the array. The start index of the array plus `minValueIndex` results in the array index of the smallest value. The index starts with 0.
maxValue	LReal	Maximum value found in the array
maxValueIndex	DInt	Index of the maximum found value in the array. The start index of the array plus `maxValueIndex` results in the array index of the smallest value. The index starts with 0.

#### In/Out parameter

Identifier	Data type	Description
values	Array[*] of LReal	Array in whose fields the maximum and minimum are searched

#### Functional description

An array of any size is connected via the `values` input. The elements are then compared in turn. The smallest and largest values, as well as their corresponding index are output to the array.

**Note** If there are several identical min. or max. values, the index of the first min. or max. value is output.

### Change log

Version & Date	Change description
<b>01.00.00</b> 11.11.2019	<b>Simatic Systems Support</b> First release copied from "LGF_SearchMinMax" and reworked to array[*]
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 09.02.2021	<b>Simatic Systems Support</b> Insert documentation

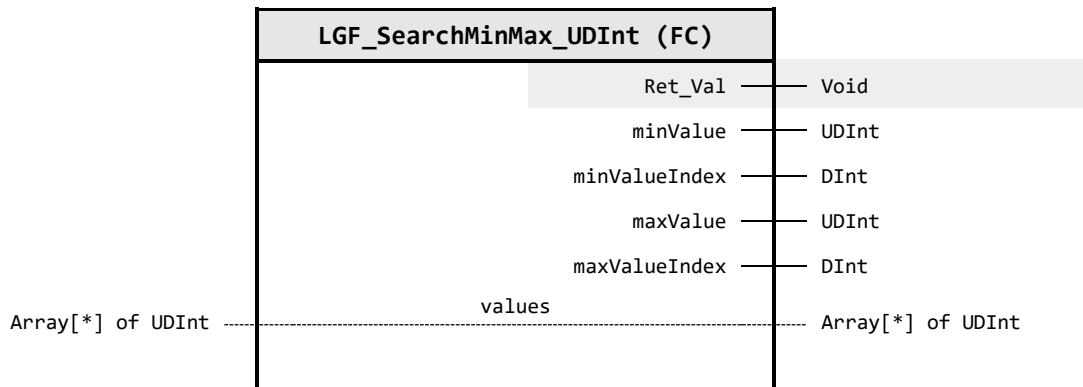
### 4.5.18 LGF\_SearchMinMax\_UDInt (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function searches, in an array of the data type UDInt, for the maximum and minimum value and the respective index in the array.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
minValue	UDInt	Minimum value found in the array
minValueIndex	DInt	Index of the minimum found value in the array. The start index of the array plus `minValueIndex` results in the array index of the smallest value. The index starts with 0.
maxValue	UDInt	Maximum value found in the array
maxValueIndex	DInt	Index of the maximum found value in the array. The start index of the array plus `maxValueIndex` results in the array index of the smallest value. The index starts with 0.

#### In/Out parameter

Identifier	Data type	Description
values	Array[*] of UDInt	Array in whose fields the maximum and minimum are searched

#### Functional description

An array of any size is connected via the `values` input. The elements are then compared in turn. The smallest and largest values, as well as their corresponding index are output to the array.

#### Note

If there are several identical min. or max. values, the index of the first min. or max. value is output.

### Change log

Version & Date	Change description
<b>01.00.00</b> 11.11.2019	<b>Simatic Systems Support</b> First release copied from "LGF_SearchMinMax" and reworked to array[*]
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 09.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.5.19 LGF\_Integration (FB / V3.0.2)

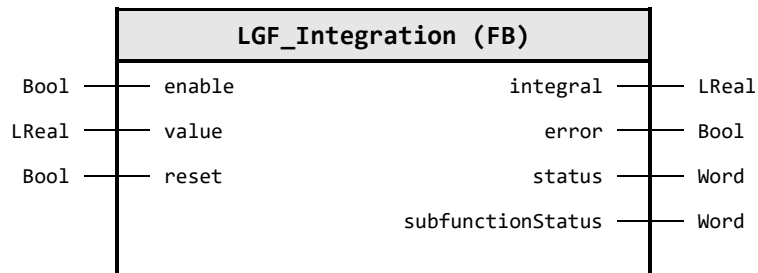
Author: Siemens Digital Industries

#### Short description

The function approximately calculates the area under a function curve. The function curve is transferred as an analog `value` (LReal) which varies over time. The integral value is output on `integral`.

The implementation is based on the trapezoidal rule and uses [ms] as time base.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
enable	Bool	FALSE	Enables integral calculation If this input is set to the value `FALSE`, the integral calculation is stopped and the `integral` output shows the last calculated value.
value	LReal	0.0	Analog value of the continuous function curve, based on [ms], (e.g. [volume flow/ms])
reset	Bool	FALSE	Sets the output "integral" to "0.0".

#### Output parameter

Identifier	Data type	Description
integral	LReal	Integral value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED_NO_ERROR Status: No error in the function during execution
16#8600	ERR_READ_SYS_TIME Error: System time FC `RD_SYS_T` returned an error when excuting - check `subFunctionStatus` code

### Functional description

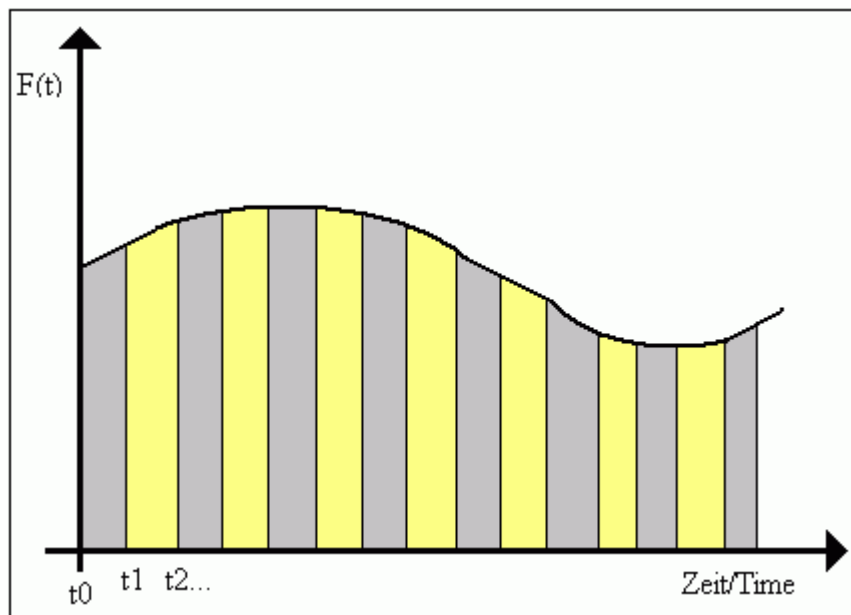
**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The integral calculation includes the summation of those trapezoidal areas that span between the last two function values on the “value” input and the time. The elapsed time is calculated via the system time of the CPU. This trapezoidal area is identical to the product of the mean value of the two process values and the time interval.

**Note** The calculation takes [ms] as time base. So the analog value has to use the same time base, e.g. [volume flow/ms].

$$A = \frac{1}{2} * (F_{t_1} + F_{t_0}) * (t_1 - t_0) + \frac{1}{2} * (F_{t_2} + F_{t_1}) * (t_2 - t_1) + \dots$$

Figure: Principle of operation



Start the integral calculation for the inputvalue at the parameter `value`:

- Set the parameter `enable` to the value `TRUE`
- Set the parameter `reset` to the value `FALSE`

If the parameter `enable` is set to the value `FALSE`, the integral calculation is stopped and the output `integral` outputs the last calculated value.

If the parameter `reset` is set to the value `TRUE`, the output `integral` is reset to `0.0`

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 17.02.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.00</b> 12.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added, code refactored
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 09.02.2021	<b>Simatic Systems Support</b> Insert documentation
<b>03.00.02</b> 07.06.2021	<b>Simatic Systems Support</b> Fix bug - incompatibility with S7-1200 and LTIME

**4.5.20 LGF\_StoreMinMax (FB / V3.0.1)**

Author: Siemens Digital Industries

**Short description**

This function reads-in a value of a variable at each call and outputs the maximum and minimum value that has been read in since the first call.  
The evaluation can be reset if necessary. The block supports the data type LReal.

**Block Interface****Input parameter**

Identifier	Data type	Default value	Description
value	LReal	0.0	Value to be compared with min/max
reset	Bool	false	TRUE: The min/max history is reset and the evaluation starts over again.

**Output parameter**

Identifier	Data type	Description
minValue	LReal	Minimum value since first call or `reset`.
maxValue	LReal	Maximum values since first call or `reset`.

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1 Code optimization
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.04</b> 09.10.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 04.02.2021	<b>Simatic Systems Support</b> Insert documentation



## 4.6 Math operations / Matrix

### 4.6.1 LGF\_MatrixAddition (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This block adds two matrices of equal size of the data type `ARRAY[*,*]` of `LREAL`.

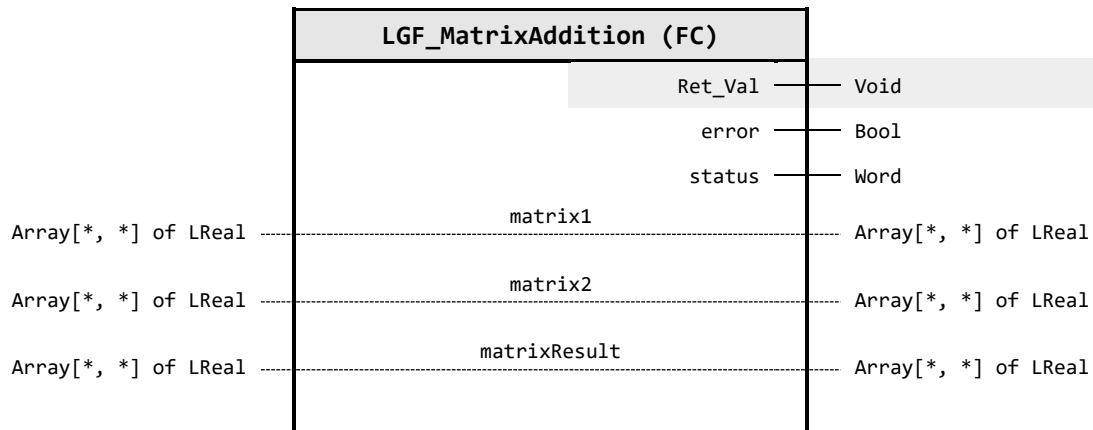
The individual fields of the two incoming matrices are read, added and then output in the matrix `matrixResult`.

$$result = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

#### Note

Note that all input and output matrices must have the same low and high limits and therefore the same number of columns and rows.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrix1	Array[*,*] of LReal	First summand (matrix)
matrix2	Array[*,*] of LReal	Second summand (matrix)
matrixResult	Array[*,*] of LReal	Sum of the matrices

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_MATR1_LOWBOUND_ROWS_MATR2_LOWBOUND_ROWS Error: Matrix1 lower bound rows(Dim1) size is different with Matrix2 lower bound rows(Dim1)
16#8201	ERR_MATR1_LOWBOUND_ROWS_RESMATR_LOWBOUND_ROWS Error: Matrix1 lower bound rows(Dim1) size is different with Result Matrix lower bound rows(Dim1)
16#8202	ERR_MATR1_LOWBOUND_COLUMNS_MATR2_LOWBOUND_COLUMNS Error: Matrix1 lower bound columns(Dim2) size is different with Matrix 2 lower bound columns(Dim2)
16#8203	ERR_MATR1_LOWBOUND_COLUMNS_RESMATR_LOWBOUND_COLUMNS Error: Matrix1 lower bound columns(Dim2) size is different with Result Matrix lower bound columns(Dim2)
16#8204	ERR_MATR1_UPPBOUND_ROWS_MATR2_UPPBOUND_ROWS Error: Matrix1 upper bound rows(Dim1) size is different with Matrix2 upper bound rows(Dim1)
16#8205	ERR_MATR1_UPPBOUND_ROWS_RESMATR_UPPBOUND_ROWS Error: Matrix1 upper bound row(Dim1)s size is different with Result Matrix upper bound rows(Dim1)
16#8206	ERR_MATR1_UPPBOUND_COLUMNS_MATR2_UPPBOUND_COLUMNS Error: Matrix1 upper bound columns(Dim2) size is different with Matrix2 upper bound columns(Dim2)
16#8207	ERR_MATR1_UPPBOUND_COLUMNS_RESMATR_UPPBOUND_COLUMNS Error: Matrix1 upper bound columns(Dim2) size is different with Result Matrix upper bound columns(Dim2)

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>02.00.00</b> 06.02.2017	<b>Siemens Industry Online Support</b> Functionality using Array[*,*]
<b>02.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>02.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.08</b> 13.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added Moved matrices to IO field.
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.6.2 LGF\_MatrixCompare (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

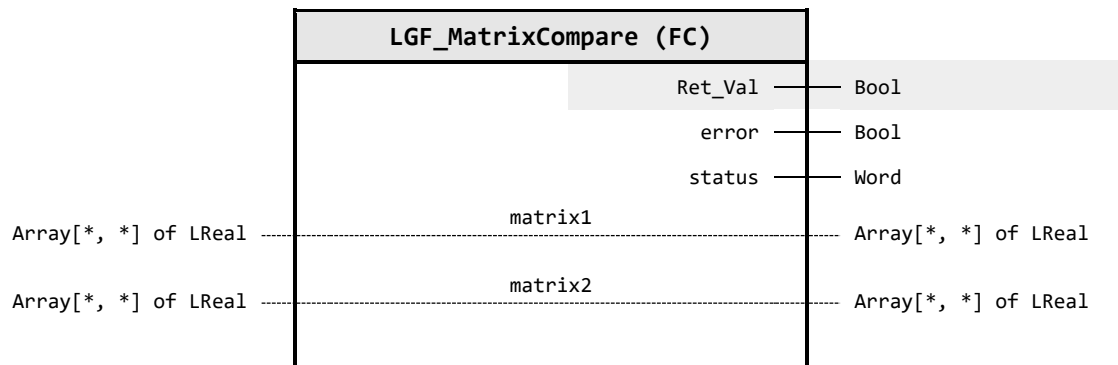
This function compares two matrices of the data type `ARRAY[*,*]` of `LREAL` of equal size.

If both matrices are identical, the return value of the function is set to `TRUE`.

#### Note

Note that all input matrices must have the same lower and upper limit, and, therefore, the same number of columns and rows.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Bool	TRUE: Both matrices are identical.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrix1	Array[*,*] of LReal	First Matrix
matrix2	Array[*,*] of LReal	Second Matrix

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_MATR1_LOWBOUND_ROWS_MATR2_LOWBOUND_ROWS Error: Matrix1 lower bound rows(Dim1) size is different with Matrix2 lower bound rows(Dim1)
16#8201	ERR_MATR1_LOWBOUND_COLUMNS_MATR2_LOWBOUND_COLUMNS Error: Matrix1 lower bound columns(Dim2) size is different with Matrix 2 lower bound columns(Dim2)

#### 4 Program blocks

Code / Value	Identifier / Description
16#8202	ERR_MATR1_UPPBOUND_ROWS_MATR2_UPPBOUND_ROWS Error: Matrix1 upper bound rows(Dim1) size is different with Matrix2 upper bound rows(Dim1)
16#8203	ERR_MATR1_UPPBOUND_COLUMNS_MATR2_UPPBOUND_COLUMNS Error: Matrix1 upper bound columns(Dim2) size is different with Matrix2 upper bound columns(Dim2)

#### Change log

Version & Date	Change description
<b>01.00.00</b> 13.11.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.6.3 LGF\_MatrixInverse (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

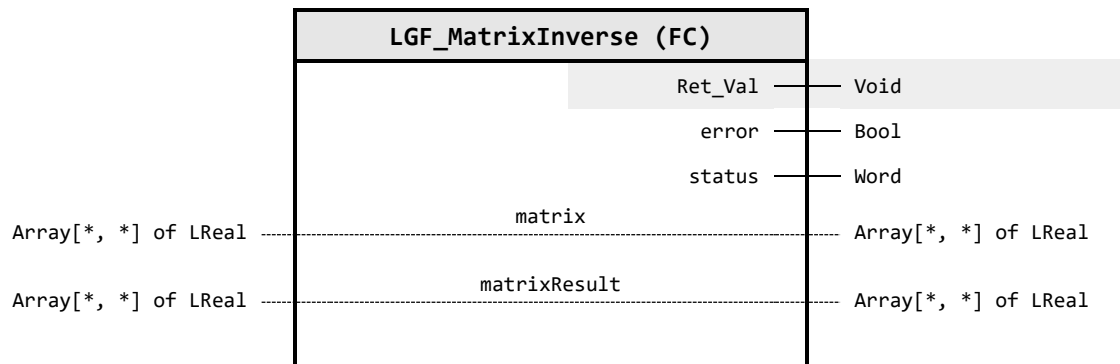
This function inverts a square matrix of the data type ARRAY[\*,\*] of LREAL. The square matrix of any size will be inverted according to the Shipley-Coleman method.

$$matrixResult = matrix^{-1}$$

#### Note

Note that the input matrix must be square. This means that the number of rows must be equal to the number of columns. The output matrix must be the same size and have the same array boundaries as the input matrix.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrix	Array[*,*] of LReal	Square input matrix that will be inversed (Array[0..x, 0..x] of REAL)
matrixResult	Array[*,*] of LReal	Inverted matrix

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_NOT_SQUARE_MATRIX Error: Matrix is not square (number of rows equals number of colums)
16#8201	ERR_ALGORITHM_NOT_POSSIBLE Matrix determinant is zero. Inversion is not possible for this matrix

#### 4 Program blocks

Code / Value	Identifier / Description
16#8202	ERR_MATR1_LOWBOUND_ROWS_RESMATR_LOWBOUND_ROWS Error: Matrix1 lower bound rows(Dim1) size is different with Result Matrix lower bound rows(Dim1)
16#8203	ERR_MATR1_LOWBOUND_COLUMNS_RESMATR_LOWBOUND_COLUMNS Error: Matrix1 lower bound columns(Dim2) size is different with Result Matrix lower bound columns(Dim2)
16#8204	ERR_MATR1_UPPBOUND_ROWS_RESMATR_UPPBOUND_ROWS Error: Matrix1 upper bound rows(Dim1) size is different with Result Matrix upper bound rows(Dim1)
16#8205	ERR_MATR1_UPPBOUND_COLUMNS_RESMATR_UPPBOUND_COLUMNS Error: Matrix1 upper bound columns(Dim2) size is different with Result Matrix upper bound columns(Dim2)

#### Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>02.00.00</b> 06.02.2017	<b>Siemens Industry Online Support</b> Functionality using Array[*,*]
<b>02.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>02.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.05</b> 13.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added Moved matrices to IO field.
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.6.4 LGF\_MatrixMultiplication (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function multiplies two matrices of the data type `ARRAY[*,*]` of `LREAL`.

Example for 2x2 matrix:

$$result = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} * b_{11} + a_{12} * b_{21} & a_{11} * b_{12} + a_{12} * b_{22} \\ a_{21} * b_{11} + a_{22} * b_{21} & a_{21} * b_{12} + a_{22} * b_{22} \end{bmatrix}$$

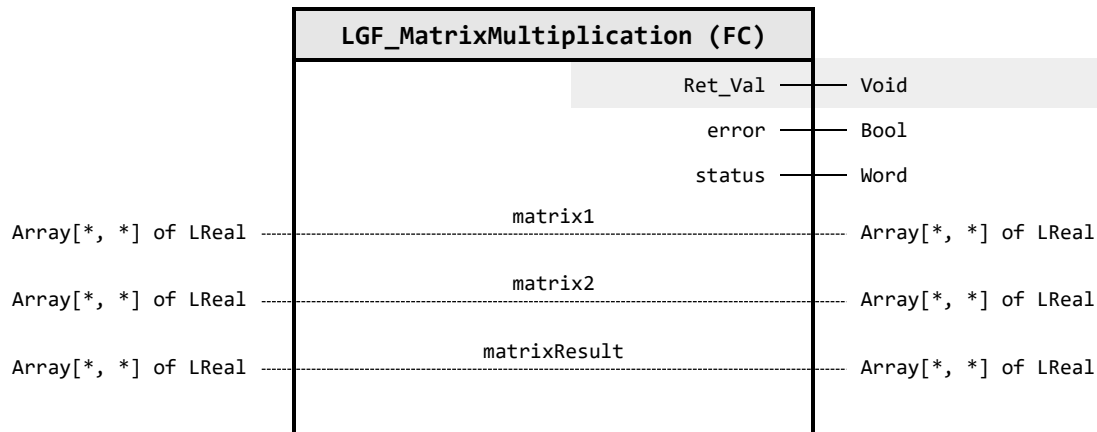
The block multiplies two matrices of variable size. The individual elements of the two incoming matrices are read, multiplied, and then output in the `matrixResult` matrix.

#### Note

Note that the number of columns in the first matrix must be equal to the number of rows in the second matrix.

The size of the initial matrix (`m * n`) results from the number of rows (`m`) of `matrix1` and the number of columns (`n`) of `matrix2`.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrix1	Array[*,*] of LReal	First factor: Matrix to multiply
matrix2	Array[*,*] of LReal	Second factor: Matrix to multiply
matrixResult	Array[*,*] of LReal	Product: The resulting matrix

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_MATR1_LOWBOUND_COLUMNS_MATR2_LOWBOUND_ROWS Error: Matrix 1 lower bound columns(Dim2) size is different with Matrix2 lower bound rows(Dim1)
16#8201	ERR_MATR1_UPPBOUND_COLUMNS_MATR2_UPPBOUND_ROWS Error: Matrix 1 upper bound columns(Dim2) size is different with Matrix 2 upper bound rows(Dim1)
16#8202	ERR_MATR1_LOWBOUND_ROWS_RESMATR_LOWBOUND_ROWS Error: Matrix 1 lower bound rows(Dim1) size is different with Result Matrix lower bound rows(Dim1)
16#8203	ERR_MATR2_LOWBOUND_COLUMNS_RESMATR_LOWBOUND_COLUMNS Error: Matrix 2 lower bound columns(Dim2) size is different with Result Matrix lower bound columns(Dim2)
16#8204	ERR_MATR1_UPPBOUND_ROWS_RESMATR_UPPBOUND_ROWS Error: Matrix 1 upper bound rows(Dim1) size is different with Result Matrix upper bound rows(Dim1)
16#8205	ERR_MATR2_UPPBOUND_COLUMNS_RESMATR_UPPBOUND_COLUMNS Error: Matrix 2 upper bound columns(Dim2) size is different with Result Matrix upper bound columns(Dim2)

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>02.00.00</b> 06.02.2017	<b>Siemens Industry Online Support</b> Functionality using Array[*,*]
<b>02.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>02.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.06</b> 13.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added Moved matrices to IO field.
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation



### 4.6.5 LGF\_MatrixScalarMultiplication (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function block multiplies a matrix of the data type `ARRAY[*,*]` of `LREAL` with a scalar.

Example for 2x2 matrix:

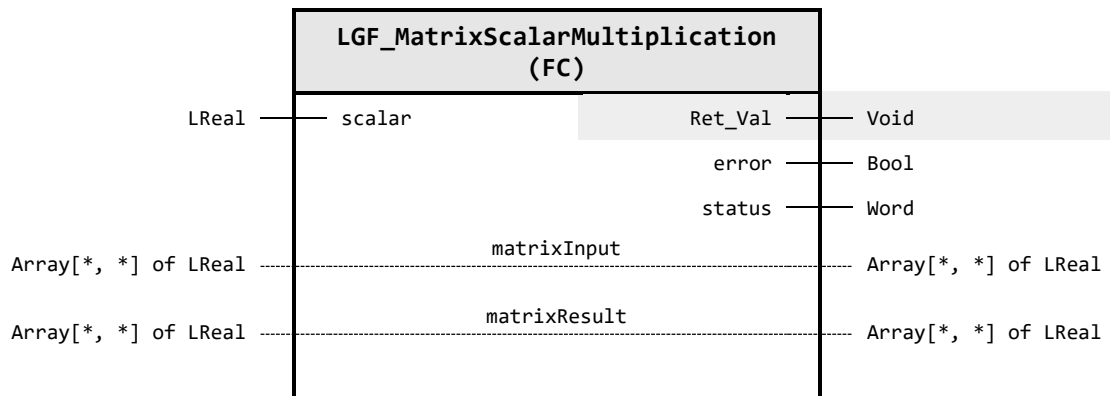
$$B = b * A = \begin{bmatrix} b * a_{11} & b * a_{12} \\ b * a_{21} & b * a_{22} \end{bmatrix}$$

A matrix is multiplied by a scalar, thereby multiplying each matrix element by the scalar. The result is output in the `matrixResult` matrix.

#### Note

Note that the input and output matrix must have the same number of columns and rows.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
scalar	LReal	Scalar value where the matrix is multiplied

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrixInput	Array[*,*] of LReal	Matrix to multiply
matrixResult	Array[*,*] of LReal	The result matrix of the multiplication

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8201	ERR_MATRICES_LOWER_BOUND_ROWS_DONT_MATCH Error: Matrices lower bound rows(Dim1) do not match
16#8202	ERR_MATRICES_UPPER_BOUND_ROWS_DONT_MATCH Error: Matrices upper bound rows(Dim1) do not match
16#8203	ERR_MATRICES_LOWER_BOUND_COLUMNS_DONT_MATCH Error: Matrices lower bound columns(Dim2) do not match
16#8204	ERR_MATRICES_UPPER_BOUND_COLUMNS_DONT_MATCH Error: Matrices upper bound columns(Dim2) do not match

**Change log**

Version & Date	Change description
<b>01.00.00</b> 11.12.2019	<b>Simatic Systems Support</b> First released version based on "LGF_MatrixMultiplication"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.6.6 LGF\_MatrixSubtraction (FC / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function subtracts a matrix of the data type `ARRAY[*,*]` of `LREAL` from another one.

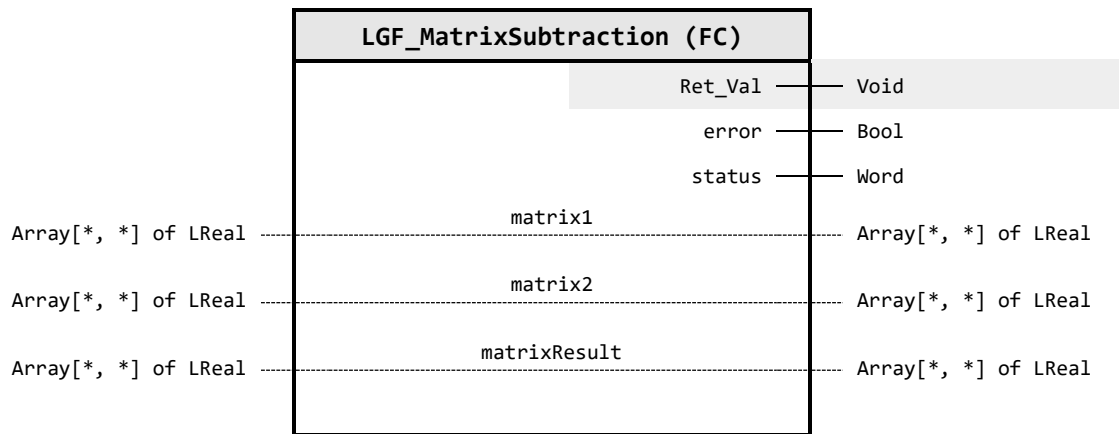
The individual fields of the two matrices are read, subtracted and then output in the matrix `matrixResult`.

$$result = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} - \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} - b_{11} & \dots & a_{1n} - b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} - b_{m1} & \dots & a_{mn} - b_{mn} \end{bmatrix}$$

#### Note

Note that all input and output matrices must have the same number of columns and rows.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrix1	Array[*,*] of LReal	First matrix - minuend
matrix2	Array[*,*] of LReal	Second matrix - subtrahend
matrixResult	Array[*,*] of LReal	Sum of the matrices

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_MATR1_LOWBOUND_ROWS_MATR2_LOWBOUND_ROWS Error: Matrix1 lower bound rows(Dim1) size is different with Matrix2 lower bound rows(Dim1)
16#8201	ERR_MATR1_LOWBOUND_ROWS_RESMATR_LOWBOUND_ROWS Error: Matrix1 lower bound rows(Dim1) size is different with Result Matrix lower bound rows(Dim1)
16#8202	ERR_MATR1_LOWBOUND_COLUMNS_MATR2_LOWBOUND_COLUMNS Error: Matrix1 lower bound columns(Dim2) size is different with Matrix 2 lower bound columns(Dim2)
16#8203	ERR_MATR1_LOWBOUND_COLUMNS_RESMATR_LOWBOUND_COLUMNS Error: Matrix1 lower bound columns(Dim2) size is different with Result Matrix lower bound columns(Dim2)
16#8204	ERR_MATR1_UPPBOUND_ROWS_MATR2_UPPBOUND_ROWS Error: Matrix1 upper bound rows(Dim1) size is different with Matrix2 upper bound rows(Dim1)
16#8205	ERR_MATR1_UPPBOUND_ROWS_RESMATR_UPPBOUND_ROWS Error: Matrix1 upper bound rows(Dim1)s size is different with Result Matrix upper bound rows(Dim1)
16#8206	ERR_MATR1_UPPBOUND_COLUMNS_MATR2_UPPBOUND_COLUMNS Error: Matrix1 upper bound columns(Dim2) size is different with Matrix2 upper bound columns(Dim2)
16#8207	ERR_MATR1_UPPBOUND_COLUMNS_RESMATR_UPPBOUND_COLUMNS Error: Matrix1 upper bound columns(Dim2) size is different with Result Matrix upper bound columns(Dim2)

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>02.00.00</b> 06.02.2017	<b>Siemens Industry Online Support</b> Functionality using Array[*,*]
<b>02.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>02.00.06</b> 07.10.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added Moved matrices to IO field.
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.6.7 LGF\_MatrixTranspose (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function transposes a matrix of the data type ARRAY[\*,\*] of LREAL.

Condition: Input matrix (m x n) = output matrix (n x m).

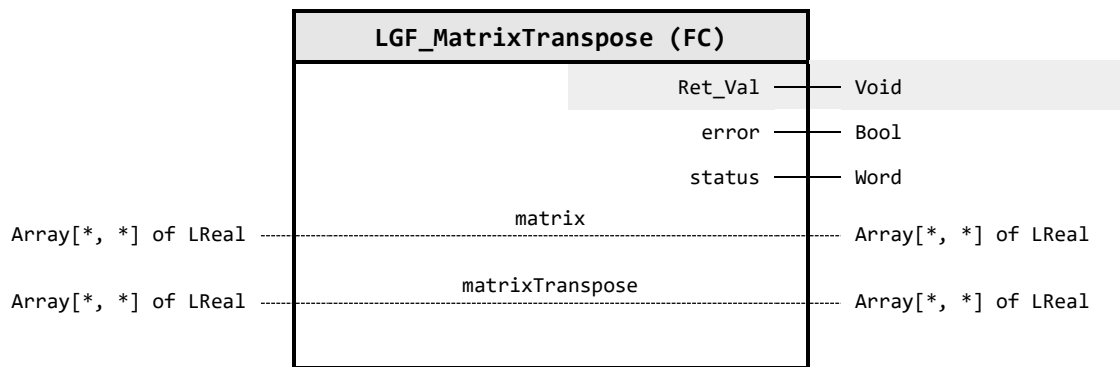
A matrix is transposed by making columns out of the rows.

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}; A^T = \begin{bmatrix} a_{11} & \dots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \dots & a_{mn} \end{bmatrix}$$

#### Note

Note that the number of rows of the input matrix must be equal to the number of columns of the output matrix. Also, the number of columns of the input matrix must be equal to the number of rows of the output matrix.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
matrix	Array[*,*] of LReal	Matrix to be transposed
matrixTranspose	Array[*,*] of LReal	Transposed matrix

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Execution finished without errors
16#8200	ERR_MATR1_LOWBOUND_ROWS_RESMATR_LOWBOUND_COLUMNS Error: Matrix1 lower bound rows(Dim1) size is different with Result Matrix lower bound columns(Dim2)

#### 4 Program blocks

Code / Value	Identifier / Description
16#8201	ERR_MATR1_LOWBOUND_COLUMNS_RESMATR_LOWBOUND_ROWS Error: Matrix1 lower bound columns(Dim2) size is different with Result Matrix lower bound rows(Dim1)
16#8202	ERR_MATR1_UPPBOUND_ROWS_RESMATR_UPPBOUND_COLUMNS Error: Matrix1 upper bound rows(Dim1) size is different with Result Matrix upper bound columns(Dim2)
16#8203	ERR_MATR1_UPPBOUND_COLUMNS_RESMATR_UPPBOUND_ROWS Error: Matrix1 upper bound columns(Dim2) size is different with Result Matrix upperbound rows(Dim1)

#### Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>02.00.00</b> 06.02.2017	<b>Siemens Industry Online Support</b> Functionality using Array[*,*]
<b>02.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>02.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.07</b> 13.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added Moved matrices to IO field.
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 02.02.2020	<b>Simatic Systems Support</b> Insert documentation

## 4.7 Data handling

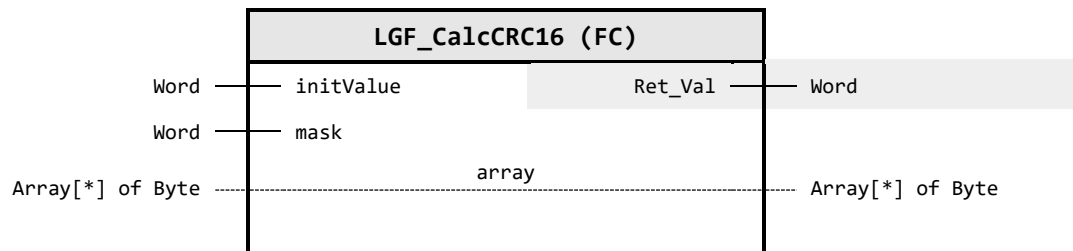
### 4.7.1 LGF\_CalcCRC16 (FC / V3.0.1)

Author: Siemens Industry Support

#### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent. The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC16` uses 16 bits as the generator polynomial (mask).

#### Block Interface



#### Input parameter

Identifier	Data type	Description
initValue	Word	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	Word	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)

#### Output parameter

Identifier	Data type	Description
Ret_Val	Word	Calculated CRC value (return value of the function).

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of Byte	Data stream for which the CRC value will be calculated.

#### Functional description

The function calculates the CRC value from a data stream of any size. The data stream is composed of the individual elements of the array at the input/output parameter `array`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

#### Note

Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):

[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 09.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Assign default start values to optional inputs - `initValue`, `mask`



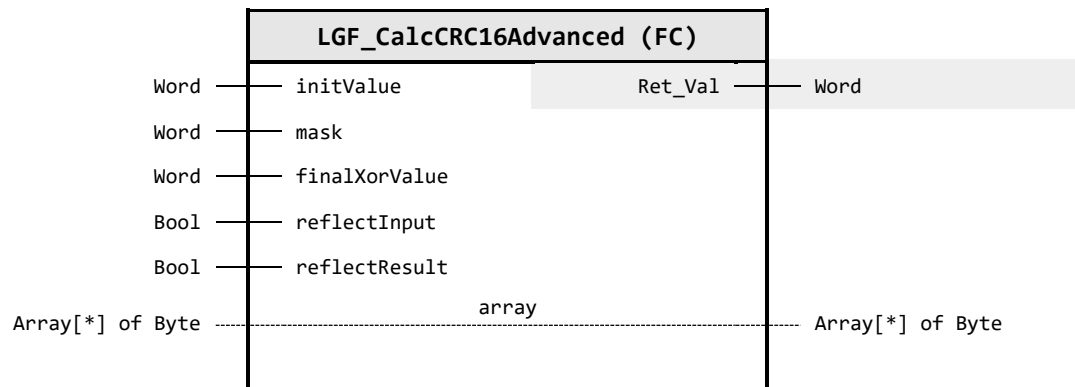
## 4.7.2 LGF\_CalcCRC16Advanced (FC / V3.0.1)

Author: Siemens Industry Support

### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent. The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC16Advanced` uses 16 bits as the generator polynomial (mask) and the parameters `finalXorValue`, `reflectInput`, and `reflectResult`.

### Block Interface



### Input parameter

Identifier	Data type	Description
initValue	Word	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	Word	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)
finalXorValue	Word	Value with which another XOR operation is performed at the end
reflectInput	Bool	TRUE: the sequence of the bits within the input byte is mirrored. The sequence 0...7 becomes 7...0.
reflectResult	Bool	TRUE: the order of the bits within the result is mirrored. The sequence 0...7 becomes 7...0.

### Output parameter

Identifier	Data type	Description
Ret_Val	Word	Calculated CRC value (return value of the function).

### In/Out parameter

Identifier	Data type	Description
array	Array[*] of Byte	Data stream for which the CRC value will be calculated.

### Functional description

The function calculates the CRC value from a data stream of any size. The data stream is composed of the individual elements of the array at the input/output parameter `array`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

Via the Boolean input parameters `reflectInput` and `reflectResult`, you may optionally mirror the bits of the input data or the CRC value. An XOR operation is also performed with the CRC value at the end and the value `finalXorValue`.

**Note**

Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):

[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

**Change log**

Version & Date	Change description
<b>01.00.00</b> 16.12.2019	<b>Simatic Systems Support</b> first release, copied from "LGF_CalcCRC32Advanced"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Assign default start values to optional inputs - <code>`initValue`</code> , <code>`mask`</code> , <code>`finalXorValue`</code> , <code>`reflectInput`</code> , <code>`reflectResult`</code>

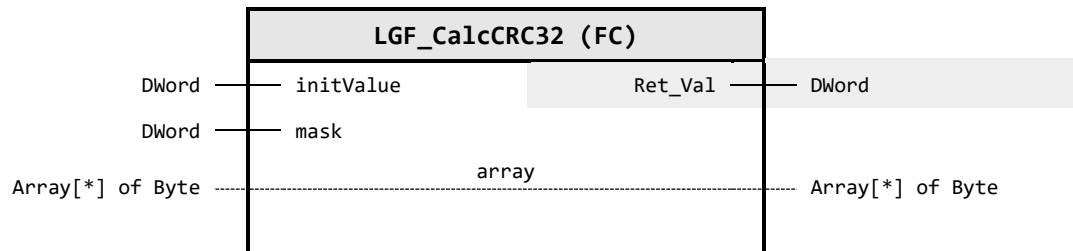
### 4.7.3 LGF\_CalcCRC32 (FC / V3.0.1)

Author: Siemens Industry Support

#### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent. The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC32` uses 32 bits as the generator polynomial (mask).

#### Block Interface



#### Input parameter

Identifier	Data type	Description
initValue	DWord	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	DWord	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Calculated CRC value (return value of the function).

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of Byte	Data stream for which the CRC value will be calculated.

#### Functional description

The function calculates the CRC value from a data stream of any size. The data stream is composed of the individual elements of the array at the input/output parameter `array`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

#### Note

Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):

[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 09.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Assign default start values to optional inputs - `initValue`, `mask`

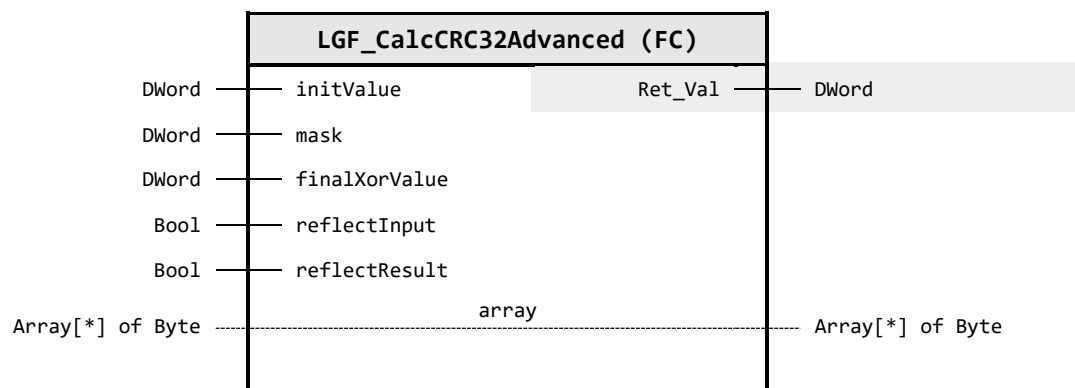
#### 4.7.4 LGF\_CalcCRC32Advanced (FC / V3.0.1)

Author: Siemens Industry Support

##### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent. The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC32Advanced` uses 32 bits as the generator polynomial (mask) and the parameters `finalXorValue`, `reflectInput`, and `reflectResult`.

##### Block Interface



##### Input parameter

Identifier	Data type	Description
initValue	DWord	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	DWord	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)
finalXorValue	DWord	Value with which another XOR operation is performed at the end
reflectInput	Bool	TRUE: the sequence of the bits within the input byte is mirrored. The sequence 0...7 becomes 7...0.
reflectResult	Bool	TRUE: the order of the bits within the result is mirrored. The sequence 0...7 becomes 7...0.

##### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Calculated CRC value (return value of the function).

##### In/Out parameter

Identifier	Data type	Description
array	Array[*] of Byte	Data stream for which the CRC value will be calculated.

##### Functional description

The function calculates the CRC value from a data stream of any size. The data stream is composed of the individual elements of the array at the input/output parameter `array`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

Via the Boolean input parameters `reflectInput` and `reflectResult`, you may optionally mirror the bits of the input data or the CRC value. An XOR operation is also performed with the CRC value at the end and the value `finalXorValue`.

**Note** Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):  
[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

### Change log

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 09.10.2019	<b>Siematic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Siematic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Siematic Systems Support</b> Insert documentation Assign default start values to optional inputs - <code>`initValue`</code> , <code>`mask`</code> , <code>`finalXorValue`</code> , <code>`reflectInput`</code> , <code>`reflectResult`</code>

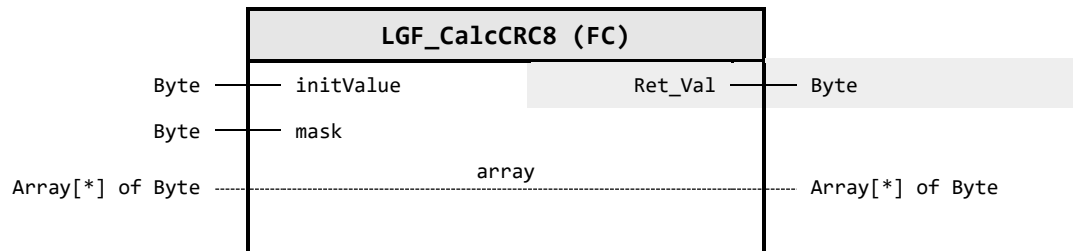
### 4.7.5 LGF\_CalcCRC8 (FC / V3.0.1)

Author: Siemens Industry Support

#### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent. The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC8` uses 8 bits as the generator polynomial (mask).

#### Block Interface



#### Input parameter

Identifier	Data type	Description
initValue	Byte	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	Byte	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)

#### Output parameter

Identifier	Data type	Description
Ret_Val	Byte	Calculated CRC value (return value of the function).

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of Byte	Data stream for which the CRC value will be calculated.

#### Functional description

The function calculates the CRC value from a data stream of any size. The data stream is composed of the individual elements of the array at the input/output parameter `array`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

#### Note

Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):

[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 09.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Assign default start values to optional inputs - `initValue`, `mask`



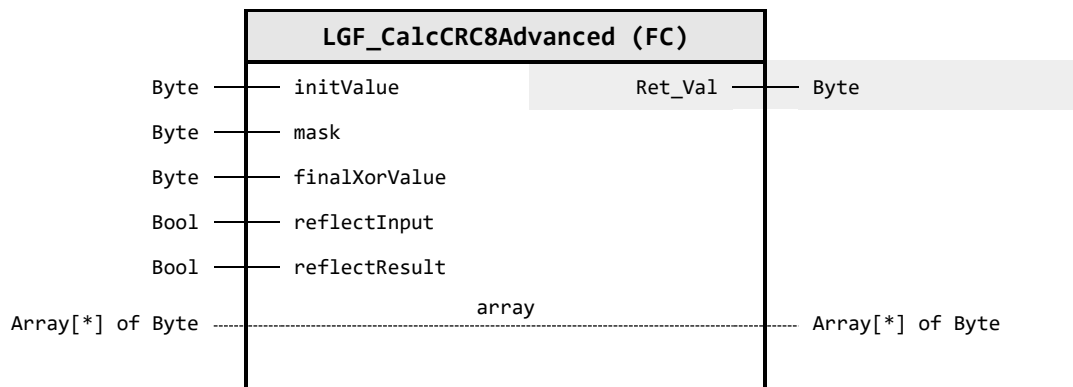
### 4.7.6 LGF\_CalcCRC8Advanced (FC / V3.0.1)

Author: Siemens Industry Support

#### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent. The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC8Advanced` uses 8 bits as the generator polynomial (mask) and the parameters `finalXorValue`, `reflectInput`, and `reflectResult`.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
initValue	Byte	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	Byte	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)
finalXorValue	Byte	Value with which another XOR operation is performed at the end
reflectInput	Bool	TRUE: the sequence of the bits within the input byte is mirrored. The sequence 0...7 becomes 7...0.
reflectResult	Bool	TRUE: the order of the bits within the result is mirrored. The sequence 0...7 becomes 7...0.

#### Output parameter

Identifier	Data type	Description
Ret_Val	Byte	Calculated CRC value (return value of the function).

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of Byte	Data stream for which the CRC value will be calculated.

#### Functional description

The function calculates the CRC value from a data stream of any size. The data stream is composed of the individual elements of the array at the input/output parameter `array`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

Via the Boolean input parameters `reflectInput` and `reflectResult`, you may optionally mirror the bits of the input data or the CRC value. An XOR operation is also performed with the CRC value at the end and the value `finalXorValue`.

**Note**

Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):

[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

### Change log

Version & Date	Change description
<b>01.00.00</b> 16.12.2019	<b>Simatic Systems Support</b> first release, copied from "LGF_CalcCRC32Advanced"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Assign default start values to optional inputs - <code>`initValue`</code> , <code>`mask`</code> , <code>`finalXorValue`</code> , <code>`reflectInput`</code> , <code>`reflectResult`</code>

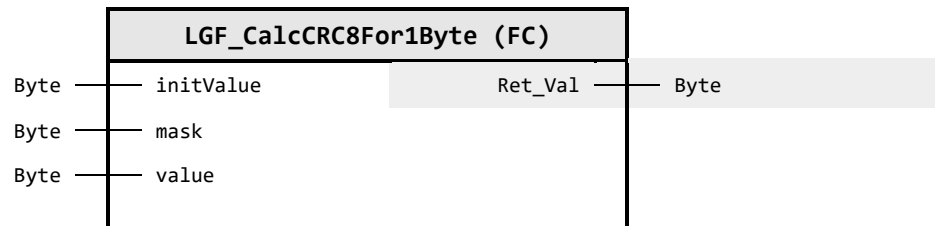
### 4.7.7 LGF\_CalcCRC8For1Byte (FC / V3.0.1)

Author: Siemens Industry Support

#### Short description

The CRC calculation is used for error detection at data transmission. The result of a calculation returns a CRC value via the data sent (Byte). The receiver detects a faulty transmission due to the unequal CRC value. The function `LGF_CalcCRC8For1Byte` uses 8 bits as the generator polynomial (mask).

#### Block Interface



#### Input parameter

Identifier	Data type	Description
initValue	Byte	Start value with which the calculation is executed. If there is no need for start value - assign 16#00
mask	Byte	Generator polynomial with which the calculation is executed. (Mask / CRC polynomial)
value	Byte	Data byte for which the CRC value will be calculated.

#### Output parameter

Identifier	Data type	Description
Ret_Val	Byte	Calculated CRC value (return value of the function).

#### Functional description

The function calculates the CRC value from a data byte `value`. The start value `initValue` and the generator polynomial `mask` can be freely selected.

#### Note

Various online tools are available for calculating the CRC values. The function of the block was tested with the following online tool, since it supports the input parameters `mask` (Polynomial) and `initValue` (Initial Value):

[http://www.sunshine2k.de/coding/javascript/crc/crc\\_js.html](http://www.sunshine2k.de/coding/javascript/crc/crc_js.html)

**Change log**

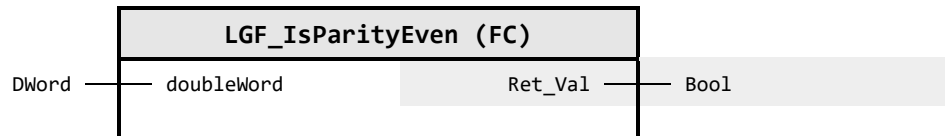
<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 09.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation Assign default start values to optional inputs - `initValue`, `mask`

**4.7.8 LGF\_IsParityEven (FC / V3.0.1)**

Author: Siemens SIMATIC Systems Support

**Short description**

The function checks whether the parity of the input variable of type DWord is even. If the number of bits that are assigned `TRUE` in the sequence is even, the return value is set to `TRUE`.

**Block Interface****Input parameter**

Identifier	Data type	Description
doubleWord	DWord	Variable for which the parity is to be determined.

**Output parameter**

Identifier	Data type	Description
Ret_Val	Bool	TRUE: When the number of bits that are assigned `TRUE` is even

**Change log**

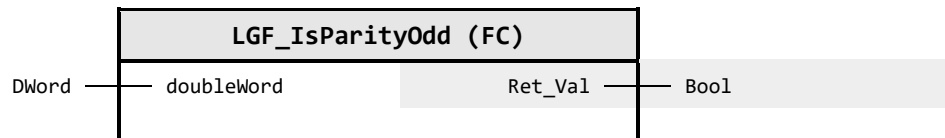
Version & Date	Change description
<b>01.00.00</b> 2019.11.28	<b>Simatic Systems Support</b> First released version
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

**4.7.9 LGF\_IsParityOdd (FC / V3.0.1)**

Author: Siemens SIMATIC Systems Support

**Short description**

The function checks whether the parity of the input variable of type DWord is odd. The return value is set to **TRUE** if the number of bits that are assigned **TRUE** in the sequence is odd.

**Block Interface****Input parameter**

Identifier	Data type	Description
doubleWord	DWord	Variable for which the parity is to be determined.

**Output parameter**

Identifier	Data type	Description
Ret_Val	Bool	TRUE: When the number of bits that are assigned `TRUE` is odd

**Change log**

Version & Date	Change description
<b>01.00.00</b> 2019.11.28	<b>Simatic Systems Support</b> First released version
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation

### 4.7.10 LGF\_FIFO (FB / V3.0.1)

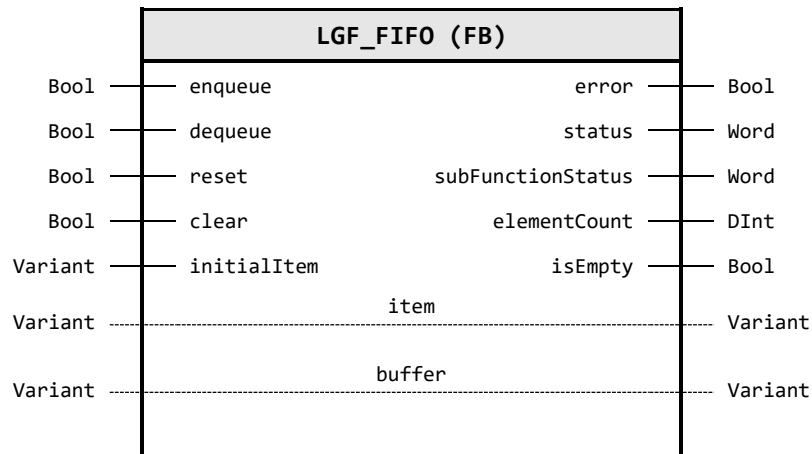
Author: Siemens Industry Support

#### Short description

FIFO (First-In First-Out / Queue / ring buffer memory)

This function stores incoming data and outputs the oldest unprocessed data.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
enqueue	Bool	false	Enqueue item to the buffer
dequeue	Bool	false	Dequeue item from the buffer and return it on `item`
reset	Bool	FALSE	Initializing the buffer (reset the index and the counter)
clear	Bool	FALSE	Clearing the buffer and initialize with the initial value `initialItem` (Reset index and counter).
initialItem	Variant	---	Value with which the ARRAY of the buffer is initialized (usually: `0` / default value)

#### Output parameter

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
elementCount	DInt	Number of elements in the buffer
isEmpty	Bool	TRUE: Buffer is empty

#### In/Out parameter

Identifier	Data type	Description
item	Variant	The entry that is either returned from the ring buffer or written into the buffer

## 4 Program blocks

Identifier	Data type	Description
buffer	VARIANT	The ARRAY that is used as the ring buffer. (Array of...)

### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#7000	STATUS_NO_CURRENT_JOBS Status: No current jobs, initial state
16#8001	ERR_BUFFER_EMPTY Error: The buffer is empty
16#8002	ERR_BUFFER_FULL Error: The buffer is full
16#8200	ERR_NO_ARRAY Error: No array is present at the input `buffer`
16#8201	ERR_WRONG_TYPE_ITEM Error: The data type of the InOut parameter `item` does not correspond to the data type of the array elements of the input `buffer`
16#8202	ERR_WRONG_TYPE_INITIAL_ITEM Error: The data type of the input `initialValue` does not correspond to the data type of the InOut parameter `item`
16#8601	ERR_INDEX_IN_ARRAY_LIMITS_1 Error: The tag `statNextEmptyItemIndex` is not within the array limits
16#8602	ERR_INDEX_IN_ARRAY_LIMITS_2 Error: The tag `statFirstItemIndex` is not within the array limits
16#8610	ERR_CLEAR_BUFFER Error: While clearing buffer in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8611	ERR_RETURN_FIRST_ENTRY Error: While return first entry of buffer in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8612	ERR_REPLACE_ITEM_BY_INIT_VALUE Error: While replace item by initial value in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8613	ERR_WRITE_ENTRY Error: While write entry to buffer in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code

### Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

**Note** In computer science, the queue is also based on the FIFO principle.

With the `enqueue` input, a new item is stored from the InOut parameter `item` in the next free position in the buffer. The output `elementCount` is incremented by one.



## 4 Program blocks

With the `dequeue` input, the next element to be processed is output to the InOut parameter `item`, and this field in the buffer is replaced by the value in the parameter `initialItem`. The output `elementCount` decremented by one.

With the `reset` input, the buffer is initialized and the index and counter are reset. The `elementCount` output is set to zero and the `isEmpty` output is set to TRUE.

With the `clear` input, the buffer is emptied and initialized with the initial value `initialItem`. Index and counter are reset. The `elementCount` output is set to zero and the `isEmpty` output is set to TRUE.

### Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 16.11.2015	<b>Siemens Industry Online Support</b> Bug fix <code>resetBuffer</code>
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 29.01.2019	<b>Siemens Industry Online Support</b> Output "done" removed (not necessary, because block works synchronous)
<b>03.00.00</b> 22.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added Interface change ( <code>enqueue</code> , <code>dequeue</code> etc.) Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.7.11 LGF\_LIFO (FB / V3.0.1)

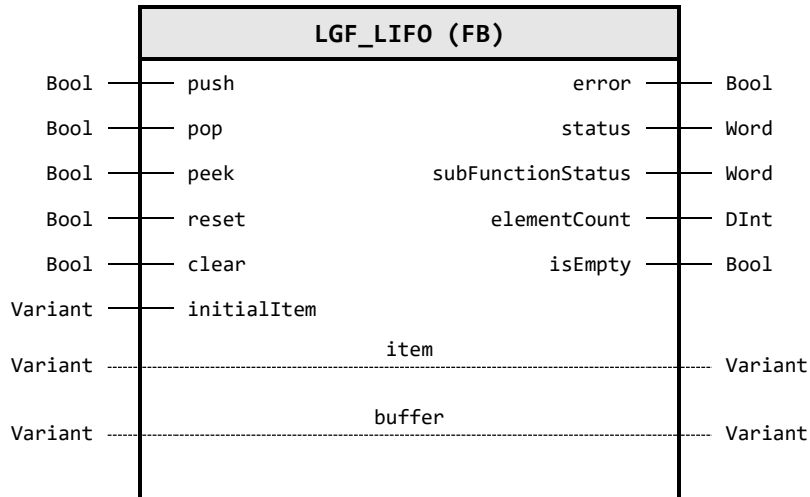
Author: Siemens Industry Support

#### Short description

LIFO (Last-In First-Out / Stack buffer memory)

This function stores incoming data and outputs the latest/most recent not-yet-processed data.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
push	Bool	false	Push item to the buffer
pop	Bool	false	Pop item from the buffer
peek	Bool	false	Peek item from the buffer (buffer not changed/modified)
reset	Bool	FALSE	Initializing the buffer (reset the index and the counter)
clear	Bool	FALSE	Clearing the buffer and initialize with the initial value `initialItem` (Reset index and counter).
initialItem	Variant	---	Value with which the ARRAY of the buffer is initialized (usually: `0` / default value)

#### Output parameter

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
elementCount	DInt	Number of elements in the buffer
isEmpty	Bool	TRUE: Buffer is empty

**In/Out parameter**

Identifier	Data type	Description
item	Variant	The entry that is either returned from the ring buffer or written into the buffer
buffer	Variant	The ARRAY that is used as the ring buffer. (Array of...)

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#7000	STATUS_NO_CURRENT_JOBS Status: No current jobs, initial state
16#8001	ERR_BUFFER_EMPTY Error: The buffer is empty
16#8002	ERR_BUFFER_FULL Error: The buffer is full
16#8200	ERR_NO_ARRAY Error: No array is present at the input `buffer`
16#8201	ERR_WRONG_TYPE_ITEM Error: The data type of the InOut parameter `item` does not correspond to the data type of the array elements of the input `buffer`
16#8202	ERR_WRONG_TYPE_INITIAL_ITEM Error: The data type of the input `initialValue` does not correspond to the data type of the InOut parameter `item`
16#8610	ERR_CLEAR_BUFFER Error: While clearing buffer in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8611	ERR_RETURN_LAST_ENTRY Error: While return first entry of buffer in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code (POP & PEEK)
16#8612	ERR_POP_REPLACE_ITEM_BY_INIT_VALUE Error: While replace item by initial value in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code (POP)
16#8613	ERR_WRITE_ENTRY Error: While write entry to buffer in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code (PUSH)

**Functional description**

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

**Note** In computer science the stack is also based on the LIFO principle.

With the `push` input, a new item is stored from the InOut parameter `item` in the next free position in the buffer. The output `elementCount` is incremented by one.

With the `pop` input, the latest / most recent item is output to the `InOut` parameter `item`, and this field in the buffer is replaced by the value at the parameter `initialItem`. The output `elementCount` is decremented by one.

The `peek` input allows the last entry in the buffer to be read out. The buffer is not changed.

With the `reset` input, the buffer is initialized and the index and counter are reset. The `elementCount` output is set to zero and the `isEmpty` output is set to TRUE.

With the `clear` input, the buffer is emptied and initialized with the initial value `initialItem`. Index and counter are reset. The `elementCount` output is set to zero and the `isEmpty` output is set to TRUE.

### Change log

Version & Date	Change description
<b>01.00.00</b> 10.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 29.01.2019	<b>Siemens Industry Online Support</b> Output "done" removed (not necessary, because block works synchron)
<b>03.00.00</b> 22.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added Interface change (push, pop, peek etc.) Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

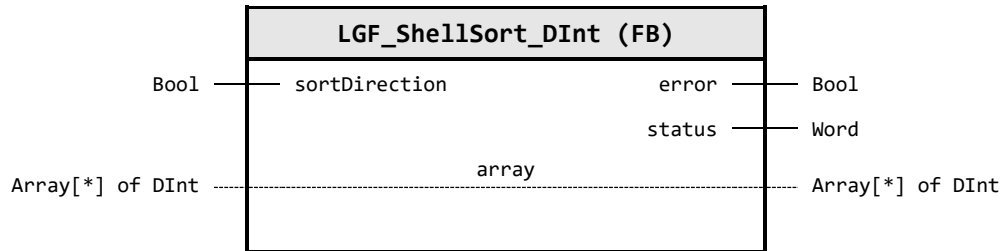
### 4.7.12 LGF\_ShellSort\_DInt (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This block sorts an array of type `DInt` with any number of elements (max. 1000) in ascending or descending order and returns the sorted version of the array in the same variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
sortDirection	Bool	FALSE	FALSE: Sort ascending; TRUE: Sort descending

#### Output parameter

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of DInt	Array to be sorted

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#8200	ERR_NO_ARRAY Error: Actual parameter at the `array` input has only one element.
16#8201	ERR_TOO_MANY_ELEMENTS Error: Actual parameter at the `array` input has too many elements (maximum is 1000).

#### Functional description

The block sorts according to the shell sort procedure. Note that the execution time of the block depends significantly on how many elements the array to be sorted has. The overview below shows several measured values of the block depending on the number of array elements.

Average steps needed for execution:  $\mathcal{O}(n \cdot \log(n)^2)$

## 4 Program blocks

Table: Execution times of the block LGF\_ShellSort...

Number of array elements	S7-1212C DC/DC/DC	S7-1516-3 PN/DP
100	approx. 11-16 ms	approx. 1-2 ms
1000	approx. 185-205 ms	approx. 10-12 ms

### Note

The block is executed synchronously and is not split over several PLC cycles. Thus the execution time has a direct effect on the PLC cycle time. Note this behavior for your project of the controller used and adjust the monitoring time of the controller if necessary.

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.01.00</b> 19.05.2016	<b>Siemens Industry Online Support</b> New function: reverse sort
<b>01.01.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.01.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 29.01.2019	<b>Siemens Industry Online Support</b> Output "done" removed (not necessary, because only one cycle)
<b>02.00.01</b> 15.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added, change data type from Int to DInt
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

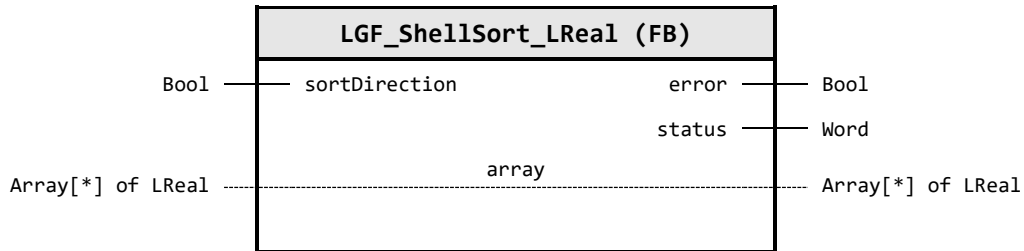
### 4.7.13 LGF\_ShellSort\_LReal (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This block sorts an array of type `LReal` with any number of elements (max. 1000) in ascending or descending order and returns the sorted version of the array in the same variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
sortDirection	Bool	FALSE	FALSE: Sort ascending; TRUE: Sort descending

#### Output parameter

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of LReal	Array to be sorted

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#8200	ERR_NO_ARRAY Error: Actual parameter at the `array` input has only one element.
16#8201	ERR_TOO_MANY_ELEMENTS Error: Actual parameter at the `array` input has too many elements (maximum is 1000).

#### Functional description

The block sorts according to the shell sort procedure. Note that the execution time of the block depends significantly on how many elements the array to be sorted has. The overview below shows several measured values of the block depending on the number of array elements.

## 4 Program blocks

Average steps needed for execution:  $\mathcal{O}(n \cdot \log(n)^2)$

Table: Execution times of the block LGF\_ShellSort...

Number of array elements	S7-1212C DC/DC/DC	S7-1516-3 PN/DP
100	approx. 11-16 ms	approx. 1-2 ms
1000	approx. 185-205 ms	approx. 10-12 ms

### Note

The block is executed synchronously and is not split over several PLC cycles. Thus the execution time has a direct effect on the PLC cycle time. Note this behavior for your project of the controller used and adjust the monitoring time of the controller if necessary.

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.01.00</b> 19.05.2016	<b>Siemens Industry Online Support</b> New function: reverse sort
<b>01.01.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.01.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 29.01.2019	<b>Siemens Industry Online Support</b> Output "done" removed (not necessary, because only one cycle)
<b>02.00.01</b> 15.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added, change data type from Real to LReal
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation



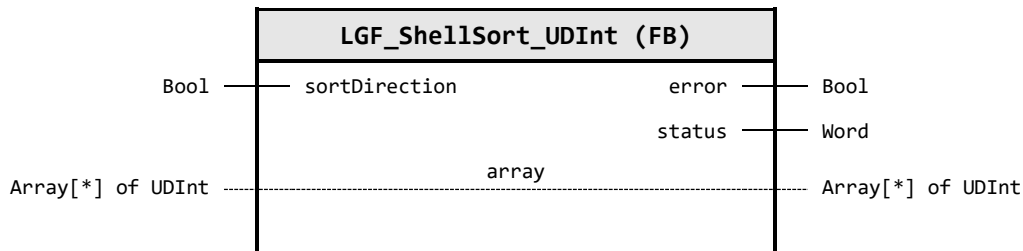
### 4.7.14 LGF\_ShellSort\_UDInt (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This block sorts an array of type UDInt with any number of elements (max. 1000) in ascending or descending order and returns the sorted version of the array in the same variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
sortDirection	Bool	FALSE	FALSE: Sort ascending; TRUE: Sort descending

#### Output parameter

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
array	Array[*] of UDInt	Array to be sorted

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#8200	ERR_NO_ARRAY Error: Actual parameter at the `array` input has only one element.
16#8201	ERR_TOO_MANY_ELEMENTS Error: Actual parameter at the `array` input has too many elements (maximum is 1000).

#### Functional description

The block sorts according to the shell sort procedure. Note that the execution time of the block depends significantly on how many elements the array to be sorted has. The overview below shows several measured values of the block depending on the number of array elements.

## 4 Program blocks

Average steps needed for execution:  $\mathcal{O}(n \cdot \log(n)^2)$

Table: Execution times of the block LGF\_She11Sort...

Number of array elements	S7-1212C DC/DC/DC	S7-1516-3 PN/DP
100	approx. 11-16 ms	approx. 1-2 ms
1000	approx. 185-205 ms	approx. 10-12 ms

### Note

The block is executed synchronously and is not split over several PLC cycles. Thus the execution time has a direct effect on the PLC cycle time. Note this behavior for your project of the controller used and adjust the monitoring time of the controller if necessary.

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.01.00</b> 19.05.2016	<b>Siemens Industry Online Support</b> New function: reverse sort
<b>01.01.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.01.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 29.01.2019	<b>Siemens Industry Online Support</b> Output "done" removed (not necessary, because only one cycle)
<b>02.00.01</b> 15.10.2019	<b>Simatic Systems Support</b> Code refactoring, comments added, change data type from UInt to UDIInt
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.7.15 LGF\_ShiftRegister (FB / V1.0.3)

Author: Siemens Digital Industry Support

#### Short description

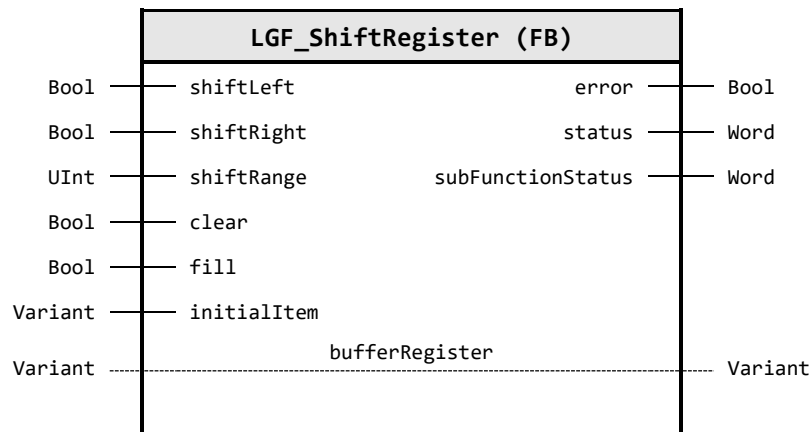
The Function represents a shift register for any kind of Datatype (using `variant`). It is possible to shift the elements in the array at `bufferRegister` to the left (index `array[n]:=array[n+1]`) or right (index `array[n]:=array[n-1]`). It could be used for material tracking trough a machine or a process, e.g. for a rotary indexing table.

#### Note

As this is a real shift operation, it may cause some runtime effects while using big array sizes to move at the input `bufferRegister`.

Please consider that a FIFO or LIFO storage, based on indexes, could be used as well for most applications.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
shiftLeft	Bool	FALSE	Rising edge: Elements in the array <code>bufferRegister</code> shifted left. Elements moved from index <code>N</code> to <code>N - 1</code> . The element at index <code>N = 0</code> is overwritten
shiftRight	Bool	FALSE	Rising edge: Elements in the array <code>bufferRegister</code> shifted right. Elements moved from index <code>N</code> to <code>N + 1</code> . The element at index <code>N = lastIndex</code> is overwritten
shiftRange	UInt	1	Number of places to be shifted in the <code>bufferRegister</code> input array
clear	Bool	FALSE	Clear buffer elements in <code>bufferRegister</code> with <code>initialItem</code>
fill	Bool	FALSE	Overwrite buffer elements after shift operation. * <code>shiftLeft</code> - the most right element/s * <code>shiftRight</code> - the most left element/s overwritten by <code>initialItem</code> .
initialItem	Variant	---	Value with which the array at input <code>bufferRegister</code> is initialized (usually the default value)

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

**In/Out parameter**

Identifier	Data type	Description
bufferRegister	VARIANT	Buffer / Register memory as ARRAY, which keeps the data. The data in the register is shifted left or right depending on the command.

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#7000	STATUS_NO_CURRENT_JOBS Status: No current jobs, initial state
16#8200	ERR_NO_ARRAY Error: No array is present at the input `bufferRegister`
16#8201	ERR_CLEARING_WITHOUT_INITIAL_ITEM Error: Clearing `bufferRegister` without an `initialItem` is not possible.
16#8202	ERR_FILL_WITHOUT_INITIAL_ITEM Error: Option `fill` the buffer after shift operation without an `initialItem` is not possible.
16#8203	ERR_WRONG_TYPE_INITIAL_ITEM Error: The data type of the input `initialItem` does not correspond to the data type of the array at the InOut parameter `bufferRegister`.
16#8401	ERR_MORE_THAN_ONE_COMMAND Error: More than one command present at the same time. Only one of the inputs `shiftLeft`, `shiftRight` or `clear` is allowed.
16#8402	ERR_IN_SHIFT_RANGE Error: The value at `shiftRange` must not exceed the maximum size of the Array at `bufferRegister`.
16#8610	ERR_CLEAR_BUFFER Error: While clearing buffer in block `MOVE_BLK_VARIANT`. Check `subFunctionStatus` code.
16#8611	ERR_SHIFT_BUFFER_LEFT Error: While shift buffer left in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8612	ERR_SHIFT_BUFFER_LEFT_FILL Error: While fill buffer after shift left in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8622	ERR_SHIFT_BUFFER_RIGHT Error: While shift buffer right in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code
16#8622	ERR_SHIFT_BUFFER_RIGHT_FILL Error: While fill buffer after shift right in block `MOVE_BLK_VARIANT` - check `subFunctionStatus` code

### Change log

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>03.00.00</b> 09.04.2021	<b>Simatic Systems Support</b> Refactoring and alignment to Datatype Variant Insert documentation

## 4.8 Converter operations

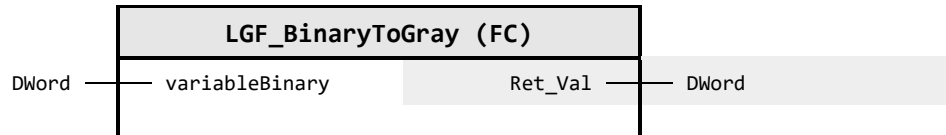
### 4.8.1 LGF\_BinaryToGray (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a binary coded value into a Gray-coded value.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
variableBinary	DWord	Binary coded value to convert to Gray code

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Gray-coded value

#### Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 28.10.2015	<b>Siemens Industry Online Support</b> Name changed
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.05</b> 11.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

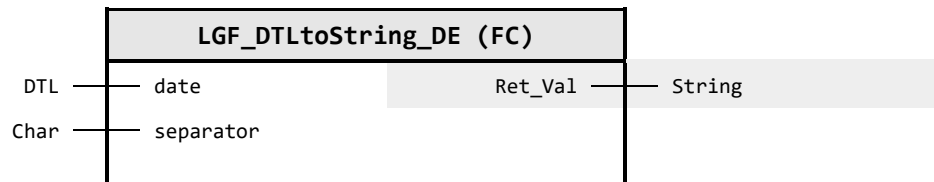
### 4.8.2 LGF\_DTLtoString\_DE (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a date of data type DTL into a character string of data type STRING in the traditional format (DD MM YYYY...).

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	DTL	Date to convert as DTL tag
separator	Char	Separator between the components of the output date.

#### Output parameter

Identifier	Data type	Description
Ret_Val	String	Output string according to the traditional format. Example: `22-01-2019 14:07:57.696417000`.

#### Functional description

The block reads a date of data type DTL and converts the individual components of the date (year, month, day, hour...) into a character string and outputs it in traditional format (DE). The separator between the components of the date is variable.

#### Traditional format (DE):

	Format																													
outString	D	D	-	M	M	-	Y	Y	Y	Y		H	H	:	M	M	:	S	S	.	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	

#### Separator:

At the input parameter `separatorDate`, you specify the separator between the components of the calendar date.

#### Example:

- `separatorDate = /` - `outString = 16/03/2016...`
- `separatorDate = -` - `outString = 16-03-2016...`

### Change log

Version & Date	Change description
<b>01.00.00</b> 18.07.2019	<b>Simatic Systems Support</b> First released version Split from "LGF_DTLtoString"
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation



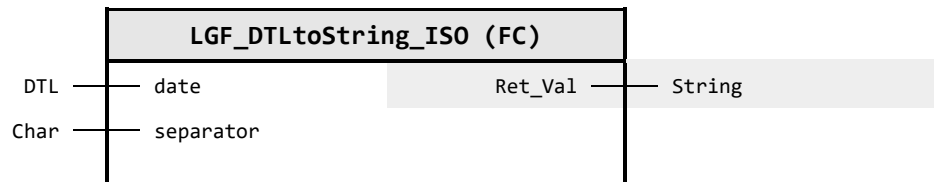
### 4.8.3 LGF\_DTLtoString\_ISO (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a date of data type DTL into a character string of data type STRING in international format (YYYY MM DD...).

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	DTL	Date to convert as DTL tag
separator	Char	Separator between the components of the output date.

#### Output parameter

Identifier	Data type	Description
Ret_Val	String	Output string in accordance with the ISO 8601 format. Example: `2019-01-22 14:06:51.524621000`.

#### Functional description

The block reads a date of data type DTL and converts the individual components of the date (year, month, day, hour...) into a character string and outputs it in international format. The separator between the components of the date is variable.

#### International format (ISO 8601):

	Format																														
outString	Y	Y	Y	Y	-	M	M	-	D	D		H	H	:	M	M	:	S	S	.	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		

#### Separator:

At the input parameter `separatorDate`, you specify the separator between the components of the calendar date.

#### Example:

- `separatorDate = /` - `outString = 2016/03/16...`
- `separatorDate = -` - `outString = 2016-03-16...`

## Change log

Version & Date	Change description
<b>01.00.00</b> 15.06.2016	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.04</b> 17.07.2019	<b>Simatic Systems Support</b> Bugfix - nanosecond precision and '0' filling
<b>01.00.05</b> 18.07.2019	<b>Simatic Systems Support</b> Renamed from "LGF_DTLtoString" to "LGF_DTLtoString_ISO" Split into two blocks, removed "format" input
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.8.4 LGF\_DTLToUnixTime (FC / V3.0.1)

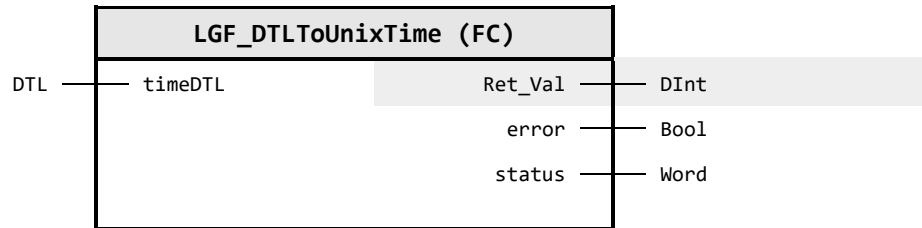
Author: Siemens Digital Industry

#### Short description

This function converts the date and time of data type DTL to the UNIX time of data type DInt. The timestamp is calculated in UTC. This means that the time zone is not considered.

Only times after 01/01/1990 are permitted.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
timeDTL	DTL	Date and time as DTL to convert to UNIX time

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Converted UNIX time
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED_NO_ERROR Execution finished without errors
16#8000	ERR_TIME_BEFORE_1990 Error: Input time is before 01/01/1990. The function does not support this conversion, because of internal used datatype

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 16.10.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 20.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update, status parameter added
<b>01.00.02</b> 10.07.2019	<b>Simatic Systems Support</b> Comments added and code refactoring Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.8.5 LGF\_GpsDDToGps (FC / V3.0.2)

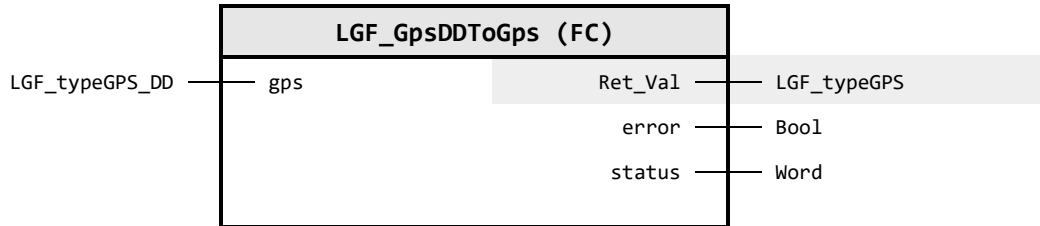
Author: Siemens Digital Industry

#### Short description

This function converts a given GPS-DD data type (decimal degrees) into a GPS data type (direction, degrees, minutes, and seconds).

GPS decimal degree to GPS “native”.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
gps	LGF_typeGPS_D	GPS-Data to be converted (decimal degrees), e.g. 52.520817 13.40945

#### Output parameter

Identifier	Data type	Description
Ret_Val	LGF_typeGPS	Converted GPS-Data (direction, degrees, minutes, and seconds), e.g. N52° 31' 14.941" E13° 24' 34.020"
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED_NO_ERROR Execution finished without errors
16#8201	ERR_LATITUDE_VALUE Error: Latitude Value
16#8203	ERR_LONGITUDE_VALUE Error: Longitude Value

**User defined datatype(s)****LGF\_typeGPS\_DD (UDT)**

Datatype for GPS Coordinates in decimal degrees.  
For latitude and longitude.  
Datatype for a whole GPS Data set.

Identifier	Data type	Default value	Description
latitude	Real	0.0	Degrees latitude with decimal places (Unit: degree decimal), North = positive; South = negative) valid value range [-90.00000..90.00000]
longitude	Real	0.0	Degrees longitude in degrees with decimal places (Unit: degree decimal), East = positive; West = negative) valid range [-180.0000..180.0000]

**LGF\_typeGPS (UDT)**

Datatype for GPS Coordinates Latitude and Longitude.  
Child Datatypes in Degree, Minutes, Seconds and the Direction.  
Datatype for a whole GPS Data set.

Identifier	Data type	Default value	Description
latitude	LGF_typeGPS_DM S	---	Latitude child element
longitude	LGF_typeGPS_DM S	---	Longitude child element

**Change log**

Version & Date	Change description
<b>01.00.00</b> 2019.09.11	<b>SIMATIC Systems Support</b> First released version
<b>03.00.00</b> 2019.04.23	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.02</b> 20.01.2021	<b>Simatic Systems Support</b> Fix `tempStatus` initialization Insert documentation

### 4.8.6 LGF\_GpsToGpsDD (FC / V3.0.2)

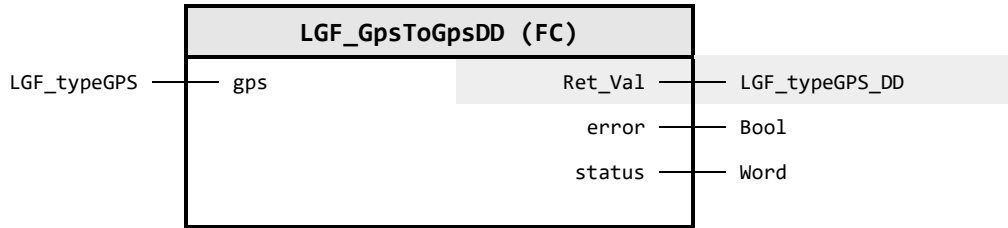
Author: Siemens Digital Industry

#### Short description

This function converts a given GPS data type (format direction, degrees, minutes, and seconds) into a GPS-DD data type (decimal degrees).

GPS "native" to GPS decimal Degree.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
gps	LGF_typeGPS	GPS-Data to be converted (direction, degrees, minutes, and seconds), e.g. N52° 31' 14.941" E13° 24' 34.020"

#### Output parameter

Identifier	Data type	Description
Ret_Val	LGF_typeGPS_D D	Converted GPS-Data (decimal degrees), e.g. 52.520817 13.40945
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED_NO_ERROR Execution finished without errors
16#8200	ERR_LATITUDE_DIRECTION Error: Latitude Direction
16#8201	ERR_LATITUDE_VALUE Error: Latitude Value
16#8202	ERR_LONGITUDE_DIRECTION Error: Longitude Direction
16#8203	ERR_LONGITUDE_VALUE Error: Longitude Value

**User defined datatype(s)****LGF\_typeGPS (UDT)**

Datatype for GPS Coordinates Latitude and Longitude.  
 Child Datatypes in Degree, Minutes, Seconds and the Direction.  
 Datatype for a whole GPS Data set.

Identifier	Data type	Default value	Description
latitude	LGF_typeGPS_DM S	---	Latitude child element
longitude	LGF_typeGPS_DM S	---	Longitude child element

**LGF\_typeGPS\_DD (UDT)**

Datatype for GPS Coordinates in decimal degrees.  
 For latitude and longitude.  
 Datatype for a whole GPS Data set.

Identifier	Data type	Default value	Description
latitude	Real	0.0	Degrees latitude with decimal places (Unit: degree decimal), North = positive; South = negative) valid value range [-90.00000..90.00000]
longitude	Real	0.0	Degrees longitude in degrees with decimal places (Unit: degree decimal), East = positive; West = negative) valid range [-180.0000..180.0000]

**Change log**

Version & Date	Change description
<b>01.00.00</b> 2019.09.11	<b>SIMATIC Systems Support</b> First released version
<b>03.00.00</b> 2019.04.23	<b>Siemens Industry Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.02</b> 20.01.2021	<b>Simatic Systems Support</b> Fix `tempStatus` initialization Insert documentation

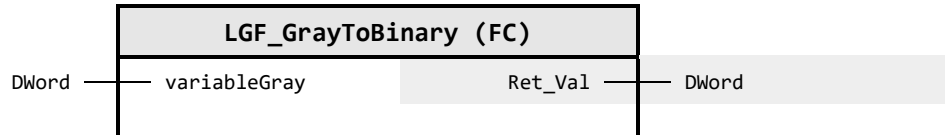


**4.8.7 LGF\_GrayToBinary (FC / V3.0.1)**

Author: Siemens Digital Industry

**Short description**

This function converts a gray coded value into a binary coded value.

**Block Interface****Input parameter**

Identifier	Data type	Description
variableGray	DWord	Gray coded value to convert to binary value

**Output parameter**

Identifier	Data type	Description
Ret_Val	DWord	Binary value

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 28.10.2015	<b>Siemens Industry Online Support</b> Name changed
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.05</b> 11.06.2019	<b>Simatic Systems Support</b> Standard header, block parameters update and performance update Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

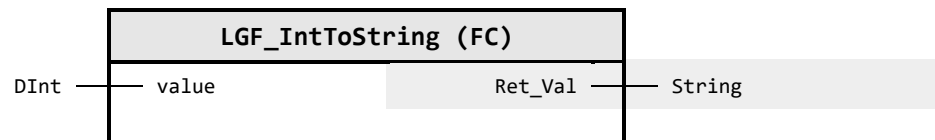
### 4.8.8 LGF\_IntToString (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a variable of the data type `DInt` into a variable of the data type `String`.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	DInt	Double-Integer value to convert

#### Output parameter

Identifier	Data type	Description
Ret_Val	String	Converted value as string. Example: '+16927'

#### Change log

Version & Date	Change description
<b>01.00.00</b> 04.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.05</b> 07.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update Program changed to VAL_STRG wrapper
<b>01.00.06</b> 30.07.2019	<b>Simatic Systems Support</b> Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

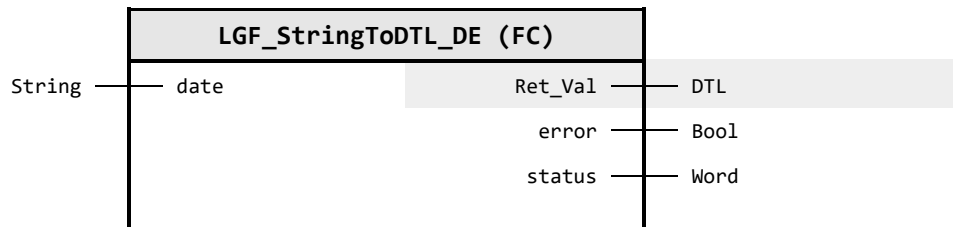
### 4.8.9 LGF\_StringToDTL\_DE (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a character string in the traditional format (DE) with date components into the data type DTL.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	String	Date as a character string according to the format. Example: `22-01-2019 14:07:57.696417000`.

#### Output parameter

Identifier	Data type	Description
Ret_Val	DTL	The converted date and time in the format DTL
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#7000	STATUS_NO_CURRENT_JOBS Status: No current job processed
16#8201	ERR_FORMAT_YEAR Error: YEAR out of range of DTL - YEAR value does not correspond to the format or specification (outside the value range of DTL)
16#8202	ERR_FORMAT_MONTH Error: MONTH out of range of DTL - MONTH value does not correspond to the format or specification (outside the value range of DTL)
16#8203	ERR_FORMAT_DAY Error: DAY out of range of DTL - DAY value does not correspond to the format or specification (outside the value range of DTL)
16#8204	ERR_FORMAT_HOUR Error: HOUR out of range of DTL - HOUR value does not correspond to the format or specification (outside the value range of DTL)
16#8205	ERR_FORMAT_MINUTE Error: MINUTE out of range of DTL - MINUTE value does not correspond to the format or specification (outside the value range of DTL)

## 4 Program blocks

Code / Value	Identifier / Description
16#8206	ERR_FORMAT_SECOND Error: SECOND out of range of DTL - SECOND value does not correspond to the format or specification (outside the value range of DTL)
16#8207	ERR_FORMAT_NANOSECOND Error: NANOSECOND out of range of DTL - NANOSECOND value does not correspond to the format or specification (outside the value range of DTL)

### Functional description

The block reads a date as a character string and converts it to the data type DTL. The individual date components in the character string are separated according to the traditional format (DE). The separator between the components in the character string is irrelevant.

### Traditional format (DE):

	Format																														
outString	D	D	-	M	M	-	Y	Y	Y	Y		H	H	:	M	M	:	S	S	.	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		

### Change log

Version & Date	Change description
<b>01.00.00</b> 22.07.2019	<b>Simatic Systems Support</b> First released version Split from "LGF_StringToDTL" Correction of the weekday of DTL, comments added Add ENO handling, adjust comments in interface
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

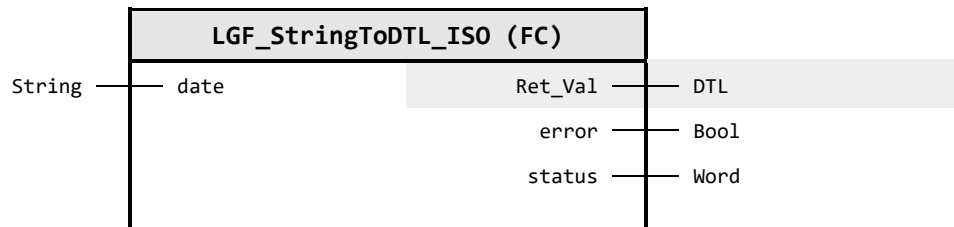
### 4.8.10 LGF\_StringToDTL\_ISO (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a character string in international format with date components into the data type DTL.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
date	String	Date as a character string according to the format. Example: `22-01-2019 14:07:57.696417000`.

#### Output parameter

Identifier	Data type	Description
Ret_Val	DTL	The converted date and time in the format DTL
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#7000	STATUS_NO_JOB Status: No current job processed
16#8201	ERR_FORMAT_YEAR Error: YEAR out of range of DTL - YEAR value does not correspond to the format or specification (outside the value range of DTL)
16#8202	ERR_FORMAT_MONTH Error: MONTH out of range of DTL - MONTH value does not correspond to the format or specification (outside the value range of DTL)
16#8203	ERR_FORMAT_DAY Error: DAY out of range of DTL - DAY value does not correspond to the format or specification (outside the value range of DTL)
16#8204	ERR_FORMAT_HOUR Error: HOUR out of range of DTL - HOUR value does not correspond to the format or specification (outside the value range of DTL)
16#8205	ERR_FORMAT_MINUTE Error: MINUTE out of range of DTL - MINUTE value does not correspond to the format or specification (outside the value range of DTL)

#### 4 Program blocks

Code / Value	Identifier / Description
16#8206	ERR_FORMAT_SECOND Error: SECOND out of range of DTL - SECOND value does not correspond to the format or specification (outside the value range of DTL)
16#8207	ERR_FORMAT_NANOSECOND Error: NANOSECOND out of range of DTL - NANOSECOND value does not correspond to the format or specification (outside the value range of DTL)

#### Functional description

The block reads a date as a character string and converts it to the data type DTL. The individual date components in the character string are separated according to the international format. The separator between the components in the character string is irrelevant.

#### International format (ISO 8601):

	Format																														
outString	Y	Y	Y	Y	-	M	M	-	D	D		H	H	:	M	M	:	S	S	.	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		

#### Change log

Version & Date	Change description
<b>01.00.00</b> 15.06.2016	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.04</b> 17.07.2019	<b>Simatic Systems Support</b> Reworked from "LGF_StringToDTL" to "LGF_StringToDTL_ISO" Removed format and split into two blocks Bugfix - set weekday correctly Correction of the weekday of DTL, comments added Add ENO handling, adjust comments in interface
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

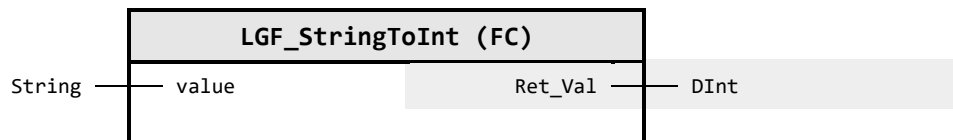
### 4.8.11 LGF\_StringToInt (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a variable of data type `String` into a variable of data type `DInt`.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	String	String value to be converted to Double-Integer. Example: '+16927'

#### Output parameter

Identifier	Data type	Description
Ret_Val	DInt	Converted Double-Integer value

#### Change log

Version & Date	Change description
<b>01.00.01</b> 12.06.2019	<b>Simatic Systems Support</b> First released version
<b>01.00.03</b> 30.07.2019	<b>Simatic Systems Support</b> Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation ENO handling done by STRG_VAL system function

### 4.8.12 LGF\_StringToTaddr (FC / V3.0.1)

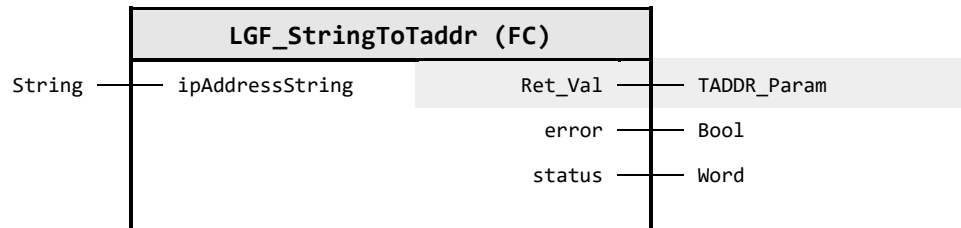
Author: Siemens Digital Industry

#### Short description

The system data type `TADDR_Param` contains address information consisting of an IPV4 address and the port number.

The `LGF_StringToTaddr` function converts a variable of data type `String` to a `TADDR_Param` system data type variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
ipAddressString	String	IPV4 address string in the format of `192.168.1.200:55047` [Port number including colon `:` is optional]

#### Output parameter

Identifier	Data type	Description
Ret_Val	TADDR_Param	IP-Address and Port number as `TADDR_Param` data type
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8110	ERR_OCTET_WRONG_NUMBER_OF_CHAR Error: Wrong number / too many characters in the X'th octet of the IP address
16#8120	ERR_OCTET_STRING_IS_EMPTY Error: No number/ character in the X'th octet of the IP address is given - String is empty
16#8130	ERR_OCTET_EXCEEDS_MAX_IP_ADDRESS Error: Maximum possible number of IP address octet exceeded (255)
16#8150	ERR_PORT_WRONG_NUMBER_OF_CHAR Error: Wrong number / too many characters in string port conversion
16#8151	ERR_PORT_STRING_IS_EMPTY Error: No number/ character in the Port string is given - String is empty
16#8152	ERR_PORT_EXCEEDS_MAX_PORT Error: Maximum number of Port exceeded (65535)



### Functional description

The function converts the IPV4 address with or without port number from data type `String` to `TADDR_Param`.

The string must be in the following form:

- without port number: `[0..255].[0..255].[0..255].[0..255]`
- with port number: `[0..255].[0..255].[0..255].[0..255]:[0..65535]`

### Example:

- The standard string format for an IPV4 address without port number:  
`192.168.11.11`
- The standard string format for an IPV4 address with port number:  
`192.168.11.11:3294`

### Note

If you do not specify a port number in the `ipAddressString` parameter, the `Ret_Val.REM_PORT_NR` parameter returns 0.

### Change log

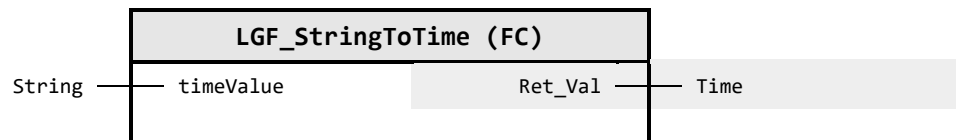
Version & Date	Change description
<b>01.00.00</b> 30.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 10.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update
<b>01.00.04</b> 10.07.2019	<b>Simatic Systems Support</b> Code refactoring and performance improvements
<b>01.00.06</b> 14.11.2019	<b>Simatic Systems Support</b> Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

**4.8.13 LGF\_StringToTime (FC / V3.0.1)**

Author: Siemens Digital Industry

**Short description**

The function converts a variable of the data type `String` into a variable of the system data type `Time`.

**Block Interface****Input parameter**

Identifier	Data type	Description
timeValue	String	Time to be converted as string Example: `1D3H45M6S0MS`

**Output parameter**

Identifier	Data type	Description
Ret_Val	Time	Converted time value Example: `T#1D_3H_45M_6S`

**Change log**

Version & Date	Change description
<b>01.00.00</b> 02.07.2019	<b>Simatic Systems Support</b> First released version
<b>01.00.01</b> 09.07.2019	<b>Simatic Systems Support</b> Further improvements and code optimization
<b>01.00.02</b> 30.07.2019	<b>Simatic Systems Support</b> Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

#### 4.8.14 LGF\_TaddrToString (FC / V3.0.1)

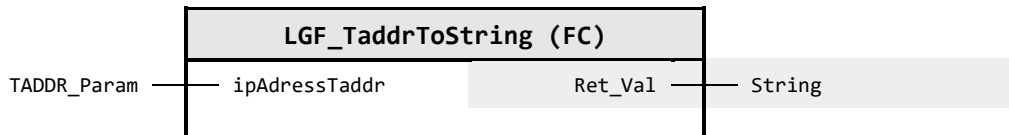
Author: Siemens Digital Industry

##### Short description

The system data type `TADDR_Param` contains address information consisting of an IPV4 address and the port number.

The `LGF_TaddrToString` function converts a `TADDR_Param` system data type variable to a `String` data type variable.

##### Block Interface



##### Input parameter

Identifier	Data type	Description
ipAdressTaddr	TADDR_Param	IP-Address and Port number to convert into string

##### Output parameter

Identifier	Data type	Description
Ret_Val	String	IP-Address and Port number as string

##### Functional description

The function converts the IPV4 address with or without port number. The system data type `TADDR_Param` is a structured data type. This structure contains the variable `REM_PORT_NR`. If this variable is 0, no port is written to the parameter `Ret_Val`.

##### Example Result at `Ret_Val`:

- The standard string format for an IPV4 address without port number:  
`192.168.11.11`
- The standard string format for an IPV4 address with port number:  
`192.168.11.11:3294`

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 19.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 17.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update
<b>01.00.04</b> 10.07.2019	<b>Simatic Systems Support</b> Refactoring of While to Do/While and constants inserted
<b>01.00.05</b> 30.07.2019	<b>Simatic Systems Support</b> Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

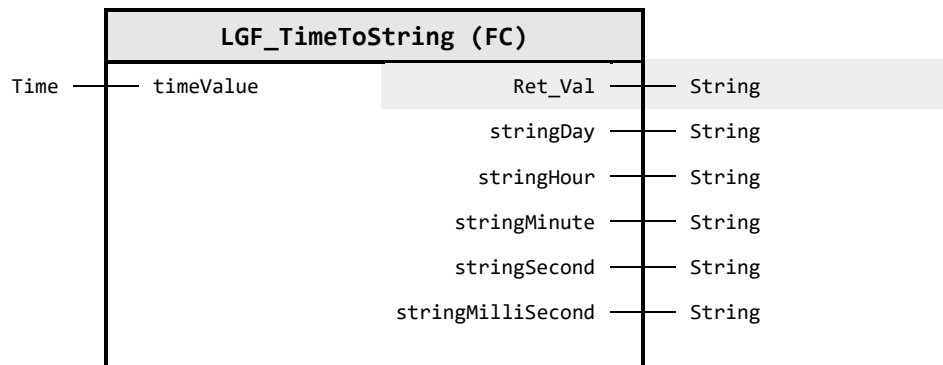
### 4.8.15 LGF\_TimeToString (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function converts a variable of the system data type `Time` into a variable of the data type `String`.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
timeValue	Time	Time value to convert Example: `T#1D_3H_45M_6S`

#### Output parameter

Identifier	Data type	Description
Ret_Val	String	Converted time as string. Example: `1D3H45M6S0MS`
stringDay	String	Converted day as string
stringHour	String	Converted hour as string
stringMinute	String	Converted minute as string
stringSecond	String	Converted second as string
stringMilliSecond	String	Converted millisecond as string

## Change log

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 02.07.2019	<b>Simatic Systems Support</b> Standard header and block parameters update, status parameter added
<b>01.00.05</b> 09.07.2019	<b>Simatic Systems Support</b> Further improvements minimization and comments added
<b>01.00.06</b> 30.07.2019	<b>Simatic Systems Support</b> Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.8.16 LGF\_UnixTimeToDTL (FC / V3.0.1)

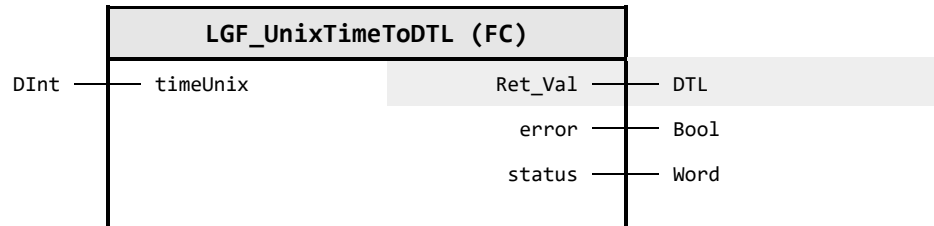
Author: Siemens Digital Industry

#### Short description

This function converts the Unix time of data type DInt to a date and time of data type DTL. The timestamp is calculated in UTC. This means that the time zone is not considered.

Only times after 01/01/1990 are permitted.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
timeUnix	DInt	UNIX time to convert

#### Output parameter

Identifier	Data type	Description
Ret_Val	DTL	Converted time (Date and time). In case of Error: 0 (error = true)
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_NO_ERROR Status: Execution finished without errors
16#6001	WARN_CONVERSION_LIMIT Warning: UNIX time (timeUnix) is exactly at the lower limit of 01.01.1990.
16#8000	ERR_TIME_BEFORE_1990 Error: UNIX time (timeUnix) is before 01/01/1990. The function does not support this conversion.

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 16.10.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 18.06.2019	<b>Simatic Systems Support</b> Standard header and block parameters update, status parameter added
<b>01.00.02</b> 10.07.2019	<b>Simatic Systems Support</b> Comments added and code intention adjusted Add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 Harmonize the version of the whole library
<b>03.00.01</b> 23.02.2021	<b>Simatic Systems Support</b> Insert documentation



## 4.9 Converter operations / Binary types - Split and Merge

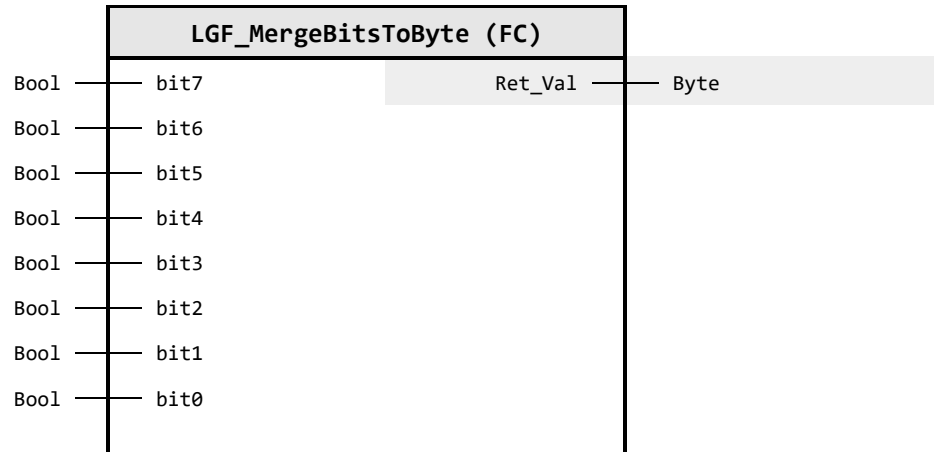
### 4.9.1 LGF\_MergeBitsToByte (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function merge 8 Bits / 8 Boolean variables into one Byte variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
bit7	Bool	Input Bit 7 - MSB
bit6	Bool	Input Bit 6
bit5	Bool	Input Bit 5
bit4	Bool	Input Bit 4
bit3	Bool	Input Bit 3
bit2	Bool	Input Bit 2
bit1	Bool	Input Bit 1
bit0	Bool	Input Bit 0 - LSB

#### Output parameter

Identifier	Data type	Description
Ret_Val	Byte	Composite Bit sequence stored as Byte variable

### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

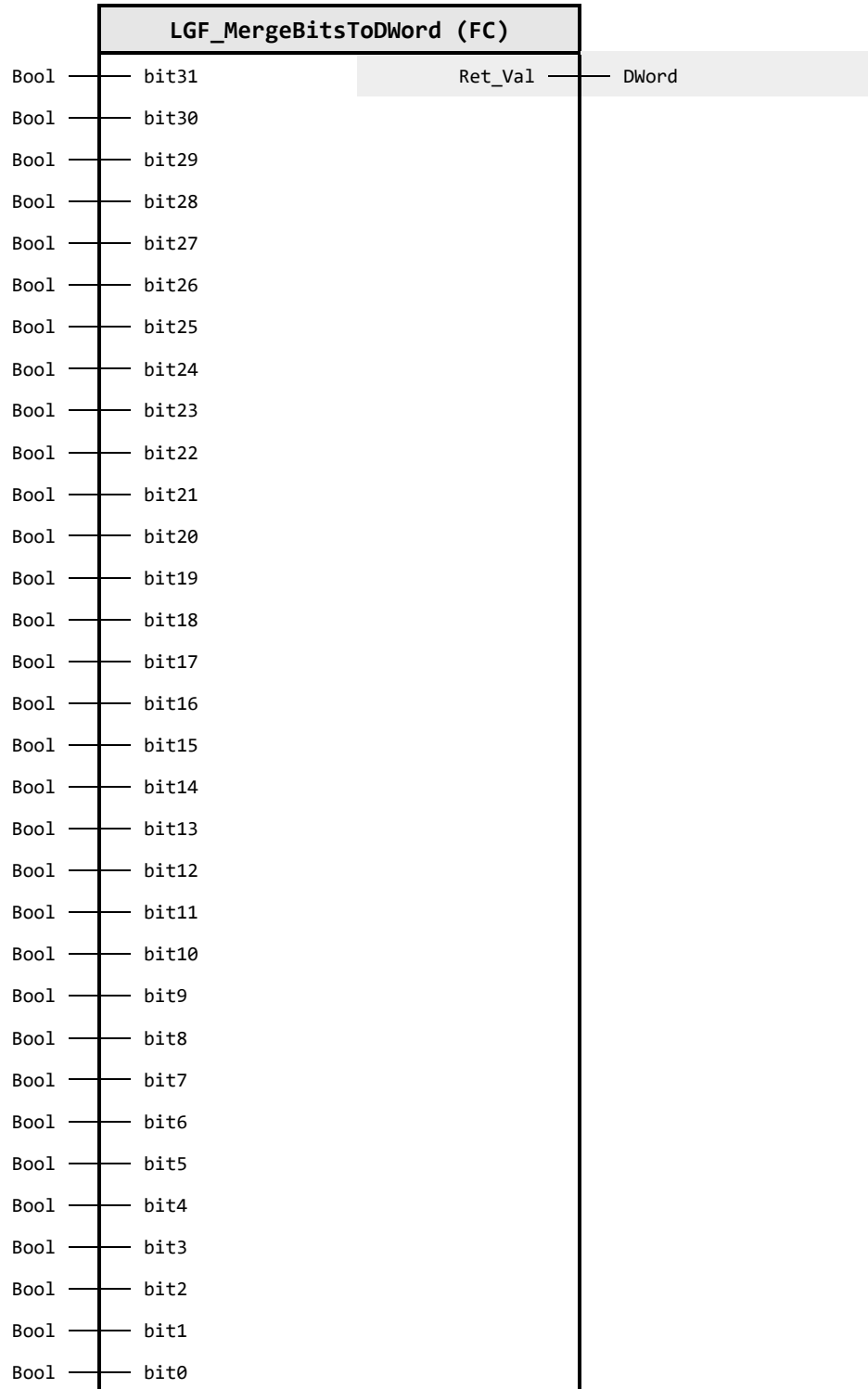
### 4.9.2 LGF\_MergeBitsToDWord (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function merge 32 Bits / 32 Boolean variables into one DWord variable.

#### Block Interface



**Input parameter**

Identifier	Data type	Description
bit31	Bool	Input Bit 31 - MSB
bit30	Bool	Input Bit 30
bit29	Bool	Input Bit 29
bit28	Bool	Input Bit 28
bit27	Bool	Input Bit 27
bit26	Bool	Input Bit 26
bit25	Bool	Input Bit 25
bit24	Bool	Input Bit 24
bit23	Bool	Input Bit 23
bit22	Bool	Input Bit 22
bit21	Bool	Input Bit 21
bit20	Bool	Input Bit 20
bit19	Bool	Input Bit 19
bit18	Bool	Input Bit 18
bit17	Bool	Input Bit 17
bit16	Bool	Input Bit 16
bit15	Bool	Input Bit 15
bit14	Bool	Input Bit 14
bit13	Bool	Input Bit 13
bit12	Bool	Input Bit 12
bit11	Bool	Input Bit 11
bit10	Bool	Input Bit 10
bit9	Bool	Input Bit 9
bit8	Bool	Input Bit 8
bit7	Bool	Input Bit 7
bit6	Bool	Input Bit 6
bit5	Bool	Input Bit 5
bit4	Bool	Input Bit 4
bit3	Bool	Input Bit 3
bit2	Bool	Input Bit 2
bit1	Bool	Input Bit 1
bit0	Bool	Input Bit 0 - LSB

**Output parameter**

Identifier	Data type	Description
Ret_Val	DWord	Composite Bit sequence stored as DWord variable

### Change log

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

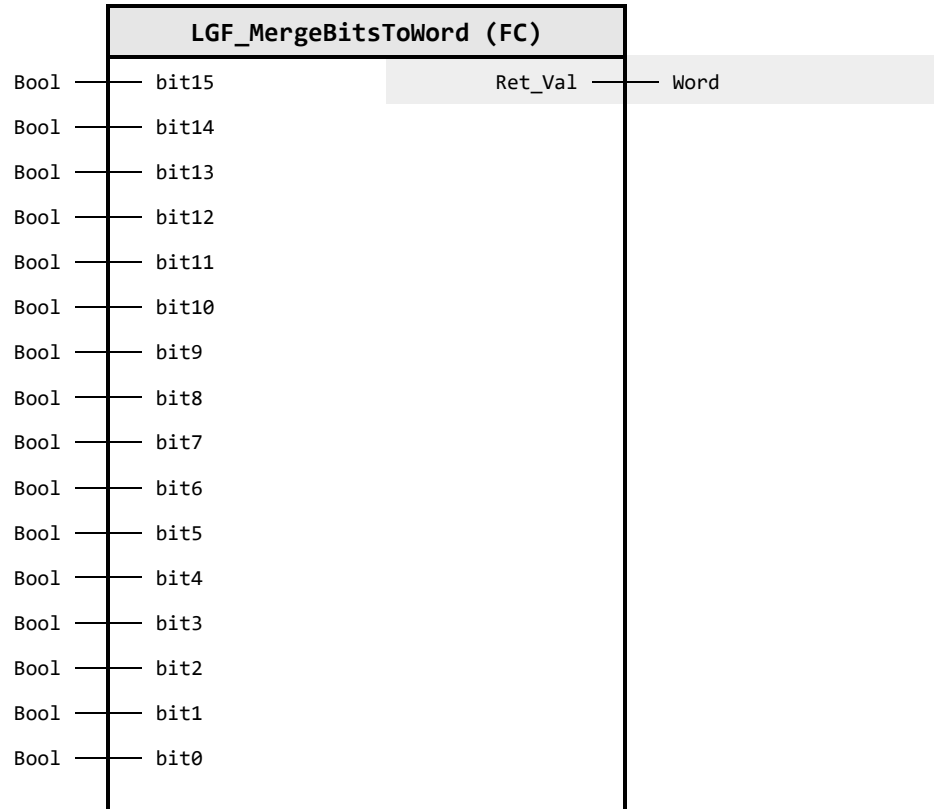
### 4.9.3 LGF\_MergeBitsToWord (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function merge 16 Bits / 16 Boolean variables into one Word variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
bit15	Bool	Input Bit 15 - MSB
bit14	Bool	Input Bit 14
bit13	Bool	Input Bit 13
bit12	Bool	Input Bit 12
bit11	Bool	Input Bit 11
bit10	Bool	Input Bit 10
bit9	Bool	Input Bit 9
bit8	Bool	Input Bit 8
bit7	Bool	Input Bit 7
bit6	Bool	Input Bit 6
bit5	Bool	Input Bit 5
bit4	Bool	Input Bit 4
bit3	Bool	Input Bit 3
bit2	Bool	Input Bit 2

#### 4 Program blocks

Identifier	Data type	Description
bit1	Bool	Input Bit 1
bit0	Bool	Input Bit 0 - LSB

#### Output parameter

Identifier	Data type	Description
Ret_Val	Word	Composite Bit sequence stored as Word variable

#### Change log

Version & Date	Change description
<b>01.00.00</b> 09.02.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 26.07.2019	<b>Simatic Systems Support</b> Standard header, style guide
<b>01.00.04</b> 30.07.2019	<b>Simatic Systems Support</b> add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

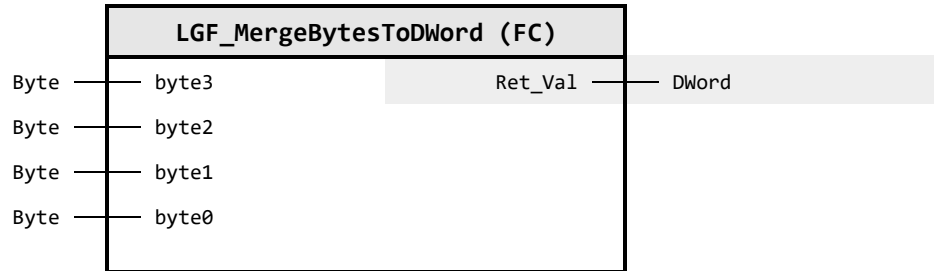
### 4.9.4 LGF\_MergeBytesToDWord (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function merge 4 Byte variables into one DWord variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
byte3	Byte	Input Byte 3 - MSB
byte2	Byte	Input Byte 2
byte1	Byte	Input Byte 1
byte0	Byte	Input Byte 0 - LSB

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Composite Byte sequence stored as DWord variable

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation



### 4.9.5 LGF\_MergeBytesToWorld (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function merge 2 Byte variables into one Word variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
byte1	Byte	Input Byte 1 - MSB
byte0	Byte	Input Byte 0 - LSB

#### Output parameter

Identifier	Data type	Description
Ret_Val	Word	Composite Byte sequence stored as Word variable

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

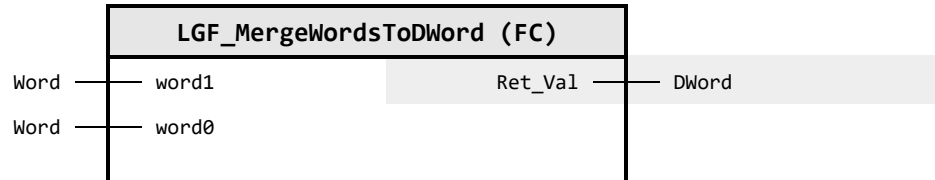
### 4.9.6 LGF\_MergeWordsToDWord (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function merge 2 Word variables into one DWord variable.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
word1	Word	Input Word 1 - MSB
word0	Word	Input Word 0 - LSB

#### Output parameter

Identifier	Data type	Description
Ret_Val	DWord	Composite Word sequence stored as DWord variable

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

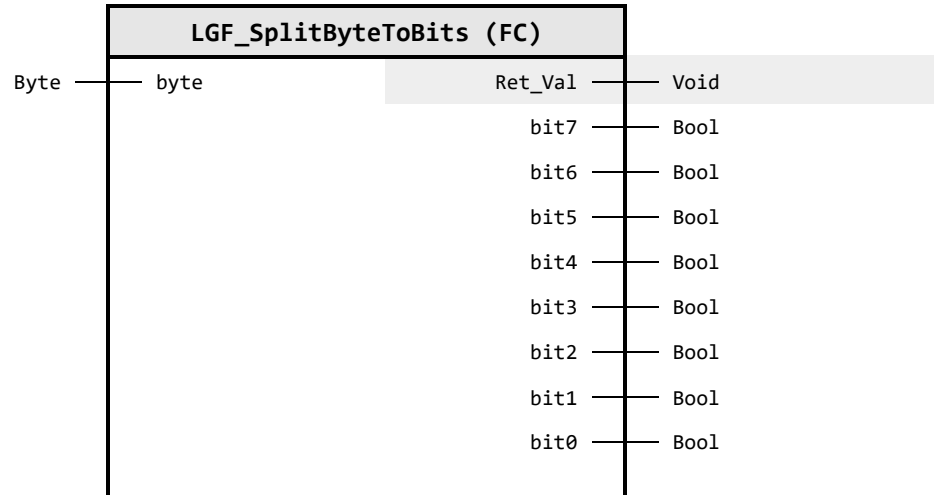
### 4.9.7 LGF\_SplitByteToBits (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function splits a Byte variable into 8 Boolean / 8 Bit variables.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
byte	Byte	Bit sequence to be split

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
bit7	Bool	Output Bit 7 - MSB
bit6	Bool	Output Bit 6
bit5	Bool	Output Bit 5
bit4	Bool	Output Bit 4
bit3	Bool	Output Bit 3
bit2	Bool	Output Bit 2
bit1	Bool	Output Bit 1
bit0	Bool	Output Bit 0 - LSB

### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

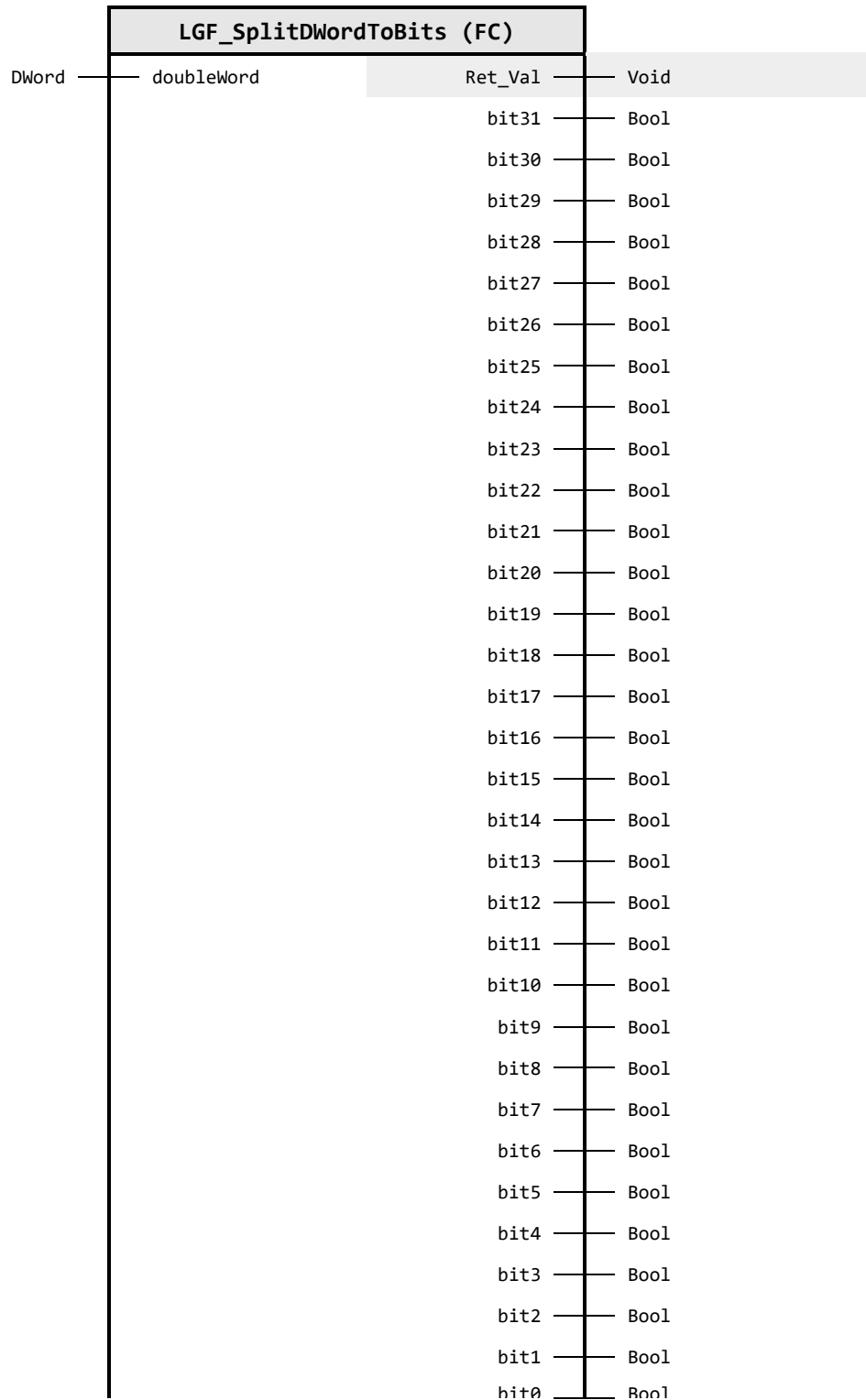
### 4.9.8 LGF\_SplitDWordToBits (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function splits a DWord variable into 32 Boolean / 32 Bit variables.

#### Block Interface





**Input parameter**

Identifier	Data type	Description
doubleWord	DWord	Bit sequence to be split

**Output parameter**

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
bit31	Bool	Output Bit 31 - MSB
bit30	Bool	Output Bit 30
bit29	Bool	Output Bit 29
bit28	Bool	Output Bit 28
bit27	Bool	Output Bit 27
bit26	Bool	Output Bit 26
bit25	Bool	Output Bit 25
bit24	Bool	Output Bit 24
bit23	Bool	Output Bit 23
bit22	Bool	Output Bit 22
bit21	Bool	Output Bit 21
bit20	Bool	Output Bit 20
bit19	Bool	Output Bit 19
bit18	Bool	Output Bit 18
bit17	Bool	Output Bit 17
bit16	Bool	Output Bit 16
bit15	Bool	Output Bit 15
bit14	Bool	Output Bit 14
bit13	Bool	Output Bit 13
bit12	Bool	Output Bit 12
bit11	Bool	Output Bit 11
bit10	Bool	Output Bit 10
bit9	Bool	Output Bit 9
bit8	Bool	Output Bit 8
bit7	Bool	Output Bit 7
bit6	Bool	Output Bit 6
bit5	Bool	Output Bit 5
bit4	Bool	Output Bit 4
bit3	Bool	Output Bit 3
bit2	Bool	Output Bit 2
bit1	Bool	Output Bit 1
bit0	Bool	Output Bit 0 - LSB

### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

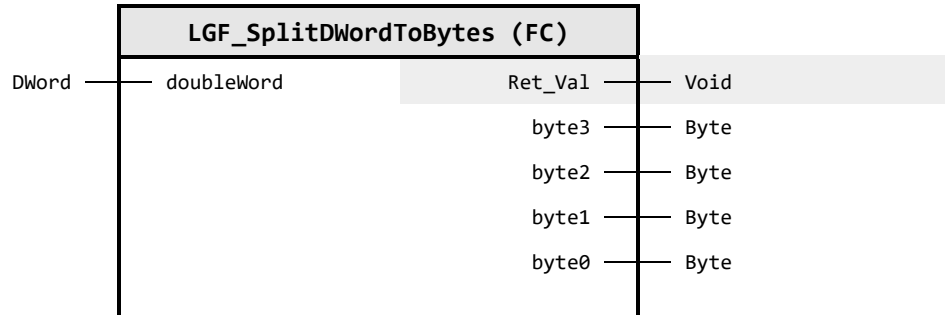
### 4.9.9 LGF\_SplitDWordToBytes (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function splits a DWord variable into 4 Byte variables.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
doubleWord	DWord	Bit sequence to be split

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
byte3	Byte	Output Byte 3 - MSB
byte2	Byte	Output Byte 2
byte1	Byte	Output Byte 1
byte0	Byte	Output Byte 0 - LSB

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation



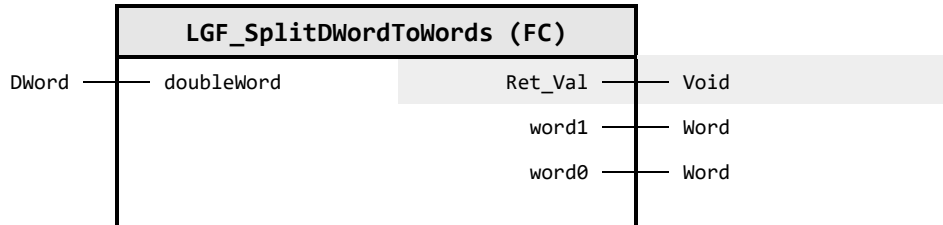
### 4.9.10 LGF\_SplitDWordToWords (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function splits a DWord variable into 2 Word variables.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
doubleWord	DWord	Bit sequence to be split

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
word1	Word	Output Word 1 - MSW
word0	Word	Output Word 0 - LSW

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

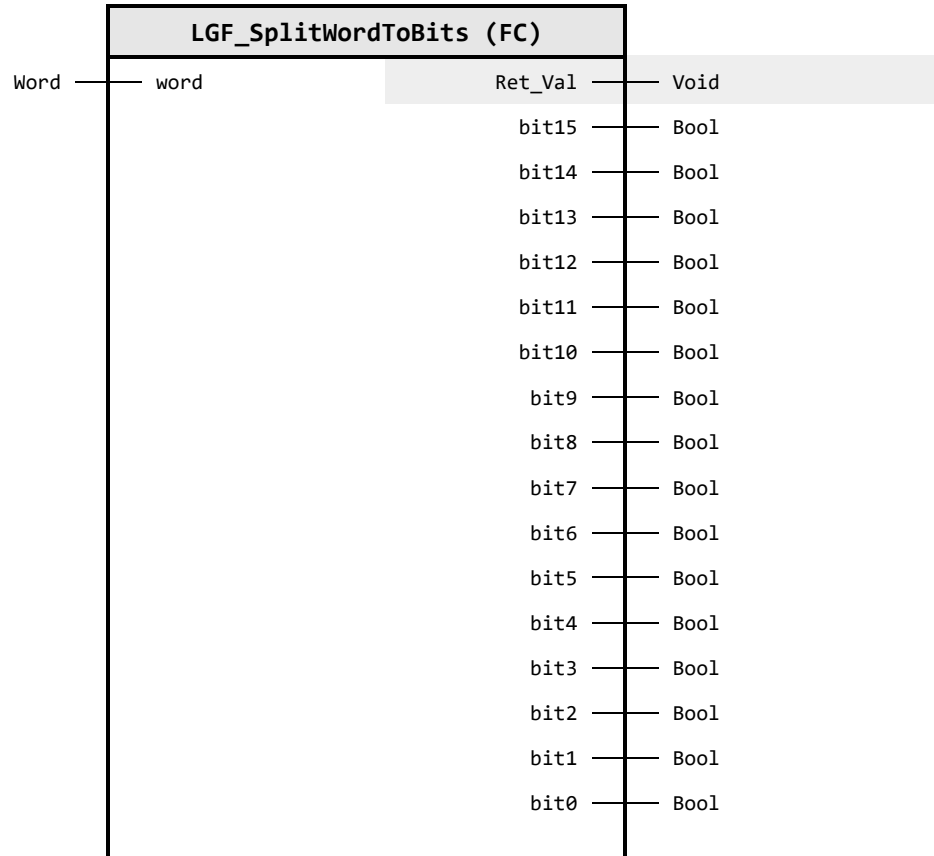
### 4.9.11 LGF\_SplitWordToBits (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function splits a Word variable into 16 Boolean / 16 Bit variables.

#### Block Interface



© Siemens AG 2021. All rights reserved

#### Input parameter

Identifier	Data type	Description
word	Word	Bit sequence to be split

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
bit15	Bool	Output Bit 15 - MSB
bit14	Bool	Output Bit 14
bit13	Bool	Output Bit 13
bit12	Bool	Output Bit 12
bit11	Bool	Output Bit 11
bit10	Bool	Output Bit 10
bit9	Bool	Output Bit 9
bit8	Bool	Output Bit 8

#### 4 Program blocks

Identifier	Data type	Description
bit7	Bool	Output Bit 7
bit6	Bool	Output Bit 6
bit5	Bool	Output Bit 5
bit4	Bool	Output Bit 4
bit3	Bool	Output Bit 3
bit2	Bool	Output Bit 2
bit1	Bool	Output Bit 1
bit0	Bool	Output Bit 0 - LSB

#### Change log

Version & Date	Change description
<b>01.00.00</b> 09.02.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 26.07.2019	<b>Simatic Systems Support</b> Standard header, style guide
<b>01.00.04</b> 30.07.2019	<b>Simatic Systems Support</b> add ENO handling
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

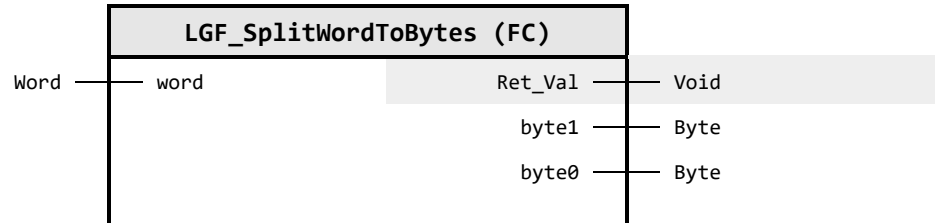
### 4.9.12 LGF\_SplitWordToBytes (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function splits a Word variable into 2 Byte variables.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
word	Word	Bit sequence to be split

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
byte1	Byte	Output Byte 1 - MSB
byte0	Byte	Output Byte 0 - LSB

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

## 4.10 Converter operations / Temperature

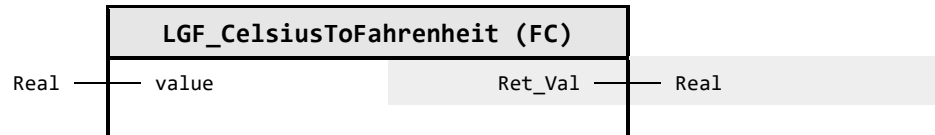
### 4.10.1 LGF\_CelsiusToFahrenheit (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Celsius to °Fahrenheit.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Celsius

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Fahrenheit

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.10.2 LGF\_CelsiusToKelvin (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Celsius to °Kelvin.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Celsius

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Kelvin

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

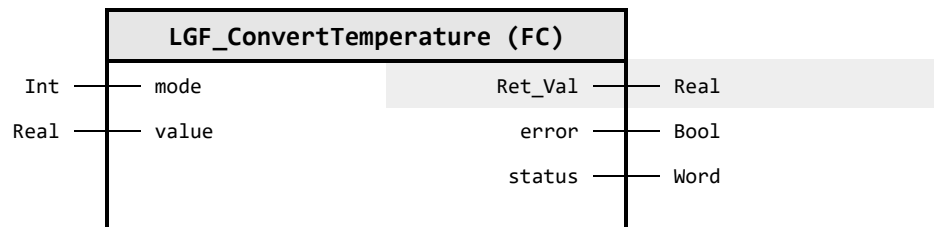
### 4.10.3 LGF\_ConvertTemperature (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value from one into another unit by using an appropriate given `mode` parameter.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
mode	Int	1: Celsius to Fahrenheit, 2: Fahrenheit to Celsius, 3: Celsius to Kelvin, 4: Kelvin to Celsius, 5: Fahrenheit to Kelvin, 6: Kelvin in Fahrenheit, 7: Rankine to Kelvin, 8: Kelvin to Rankine
value	Real	Temperature value to be converted

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature result
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#8200	ERR_WRONG_MODE Error: Incorrect mode on input "mode", see description of the input parameters

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 16.12.2018	<b>Simatic Systems Support</b> Rename from "LGF_TemperatureConvert" to "LGF_ConvertTemperature" to start with the verb include the Rankine conversion Code refactoring, regions, comments and constants
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation



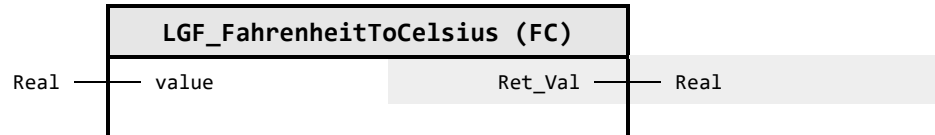
### 4.10.4 LGF\_FahrenheitToCelsius (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Fahrenheit to °Celsius.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Fahrenheit

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Celsius

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.10.5 LGF\_FahrenheitToKelvin (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Fahrenheit to °Kelvin.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Fahrenheit

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Kelvin

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

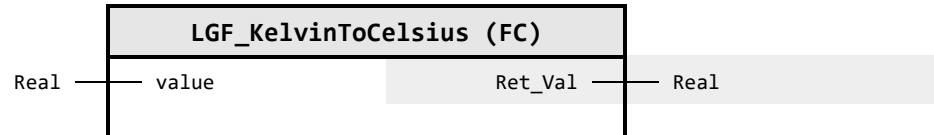
### 4.10.6 LGF\_KelvinToCelsius (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Kelvin to °Celsius.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Kelvin

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Celsius

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

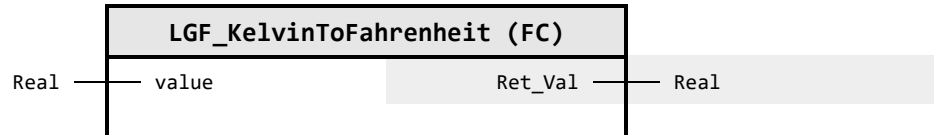
### 4.10.7 LGF\_KelvinToFahrenheit (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Kelvin to °Fahrenheit.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Kelvin

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Fahrenheit

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

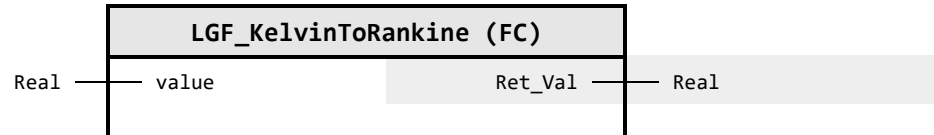
### 4.10.8 LGF\_KelvinToRankine (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Kelvin to °Rankine.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Kelvin

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Rankine

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.10.9 LGF\_RankineToKelvin (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

This function converts a temperature value - from °Rankine to °Kelvin.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
value	Real	Temperature to be converted in °Rankine

#### Output parameter

Identifier	Data type	Description
Ret_Val	Real	Converted temperature in °Kelvin

#### Change log

Version & Date	Change description
<b>01.00.00</b> 20.06.2019	<b>Simatic Systems Support</b> First release
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0 harmonize the version of the whole library
<b>03.00.01</b> 22.02.2021	<b>Simatic Systems Support</b> Insert documentation

## 4.11 Signal generators

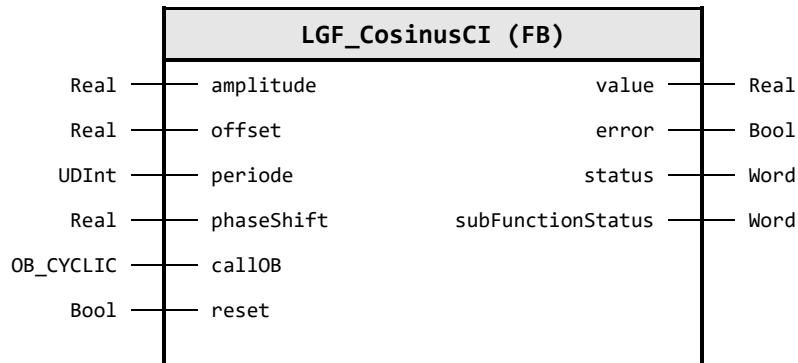
### 4.11.1 LGF\_CosinusCI (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates a cosinusoidal signal profile. For this it uses the time interval of the calling Cyclic Interrupt OB.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
amplitude	Real	1.0	Amplitude of the signal profile.
offset	Real	0.0	Offset of the signal profile in the Y-direction.
periode	UDInt	1000	Period duration of the signal profile in [ms]
phaseShift	Real	0.0	Phase offset in [ms]
callOB	OB_CYCLIC	---	Calling cyclic interrupt OB
reset	Bool	FALSE	Reset of the signal profile.

#### Output parameter

Identifier	Data type	Description
value	Real	Current value of the cosinusoidal signal.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8600	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.

Code / Value	Identifier / Description
16#8601	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

**Functional description**

**Note** The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The block calculates the values for a cosinusoidal signal profile, which is output to the output parameter value.

The amplitude, the offset in the Y-direction, the period, and the phase shift can be set at the input parameters.

The input parameter reset resets the signal profile. At the value output parameter, the value 0 is output as long as reset is set to TRUE.

The block must be called in a cyclic interrupt OB. The time interval of the calling cyclic interrupt OB is determined in the FB with the command QRY\_CINT. For this, the constant name of the calling cyclic interrupt OB must be interconnected at the input parameter callOB.



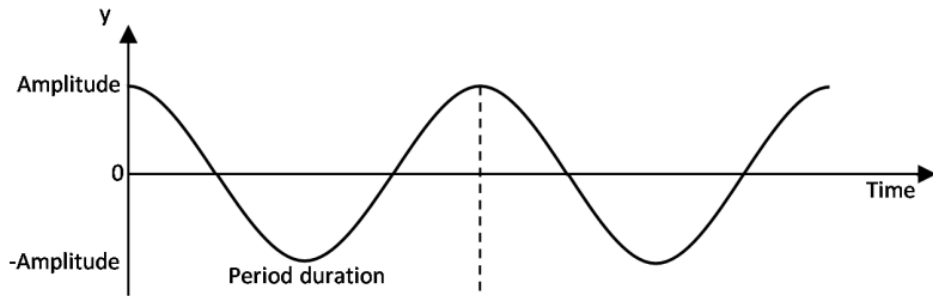
The number of calculated values of the signal profile per period duration is calculated as follows:

$$QuantityValues = \frac{Periodduration}{TimeintervakCyclicinterruptOB}$$

**Note** To obtain a continuous signal profile of the curve, the time interval of the cyclic interrupt OB should not be selected too large depending on the period duration.



The Figure below shows the signal profile of the calculated values.



**Change log**

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 26.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added phase shift availability added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

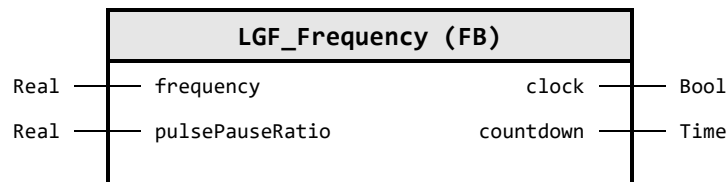
### 4.11.2 LGF\_Frequency (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates a signal that changes between the values `FALSE` and `TRUE` depending on a defined frequency and a pulse pause ratio.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
frequency	Real	0.0	Clock frequency in Hz.
pulsePauseRatio	Real	1.0	Pulse pause ratio (standard: 1.0 corresponds to 1:1).

#### Output parameter

Identifier	Data type	Description
clock	Bool	Output changes with defined frequency.
countdown	Time	Remaining time of the current `clock` state.

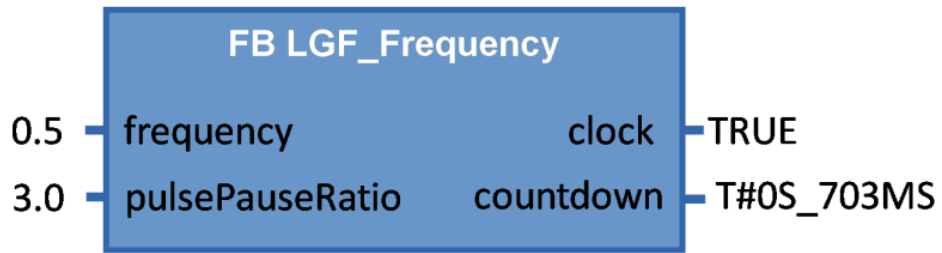
#### Functional description

The `clock` output is a Boolean value that toggles at the desired frequency. The `pulsePauseRatio` input is used to set the pulse pause ratio.

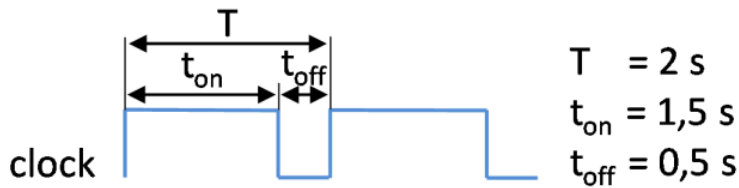
The output `countdown` outputs the remaining time of the current state of `clock`.

If the desired frequency or pulse pause ratio is less than or equal to `0.0`, the output `clock` = `FALSE` and `countdown` = `0s`.

Example



$$pulsePauseRatio = \frac{t_{on}}{t_{off}} = \frac{3}{1}$$



**Note** The clock of the FB LGF\_Frequency depends on the cycle time of the OB Main. To increase the accuracy, the FB can also be called in a cyclic interrupt OB with a low time interval.

Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.01.00</b> 25.05.2016	<b>Siemens Industry Online Support</b> New function: pulse pause ratio
<b>01.01.01</b> 26.05.2016	<b>Siemens Industry Online Support</b> Add comments
<b>01.01.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.01.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.07</b> 20.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

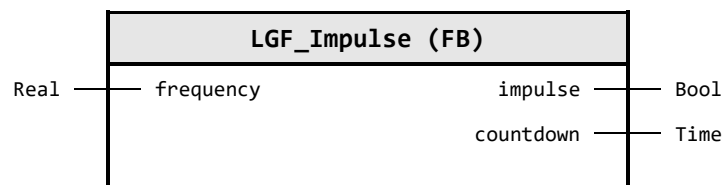
### 4.11.3 LGF\_Impulse (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates pulses at a given frequency. The pulse is always present for one (control) cycle.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
frequency	Real	0.0	Clock frequency in Hz

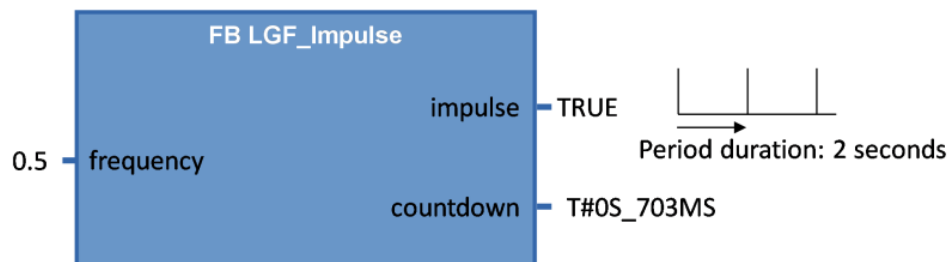
#### Output parameter

Identifier	Data type	Description
impulse	Bool	Impulse signal output
countdown	Time	Time until next pulse

#### Functional description

The function generates pulses at the output `impulse` with the frequency `frequency`. The block always begins with a pulse and sets the next pulse after the period that has elapsed.

#### Example



## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.01.01</b> 16.06.2015	<b>Siemens Industry Online Support</b> LGF_Impulse calls new LGF_Frequency V1.1.1
<b>01.01.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.02.00</b> 02.02.2017	<b>Siemens Industry Online Support</b> Code optimization: no call of LGF_Frequency Fix at output "countdown"
<b>01.02.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.02.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.02.04</b> 20.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

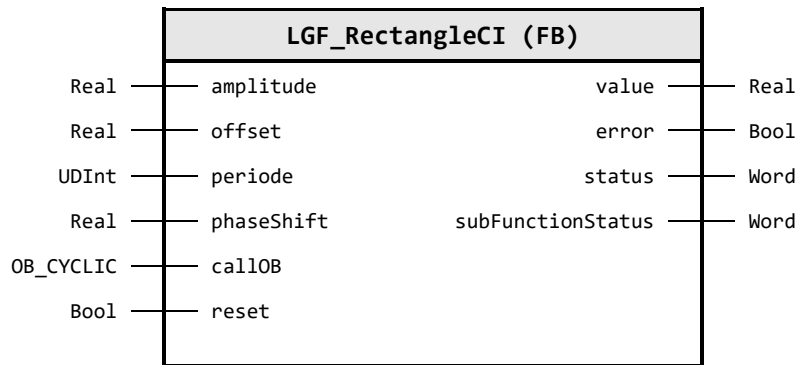
### 4.11.4 LGF\_RectangleCI (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates a rectangular signal profile. For this it uses the time interval of the calling Cyclic Interrupt OB.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
amplitude	Real	1.0	Amplitude of the signal profile.
offset	Real	0.0	Offset of the signal profile in the Y-direction.
periode	UDInt	1000	Period duration of the signal profile in [ms]
phaseShift	Real	0.0	Phase offset in [ms]
callOB	OB_CYCLIC	---	Calling cyclic interrupt OB
reset	Bool	FALSE	Reset of the signal profile.

#### Output parameter

Identifier	Data type	Description
value	Real	Current value of the rectangular signal.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8600	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.
16#8601	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

## Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The block calculates the values for a rectangular signal profile, which is output to the output parameter `value`.

The `amplitude`, the `offset` in the Y-direction, the `period`, and the `phase shift` can be set at the input parameters.

The input parameter `reset` resets the signal profile. At the `value` output parameter, the value 0 is output as long as `reset` is set to `TRUE`.

The block must be called in a cyclic interrupt OB. The time interval of the calling cyclic interrupt OB is determined in the FB with the command `QRY_CINT`. For this, the constant name of the calling cyclic interrupt OB must be interconnected at the input parameter `callOB`.

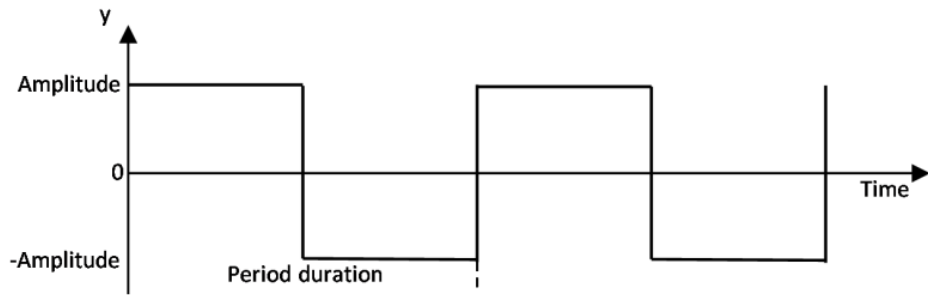


The number of calculated values of the signal profile per period duration is calculated as follows:

$$QuantityValues = \frac{Periodduration}{TimeintervakCyclicinterruptOB}$$

**Note** To obtain a continuous signal profile of the curve, the time interval of the cyclic interrupt OB should not be selected too large depending on the period duration.

The Figure below shows the signal profile of the calculated values.



**Change log**

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 26.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added phase shift availability added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation



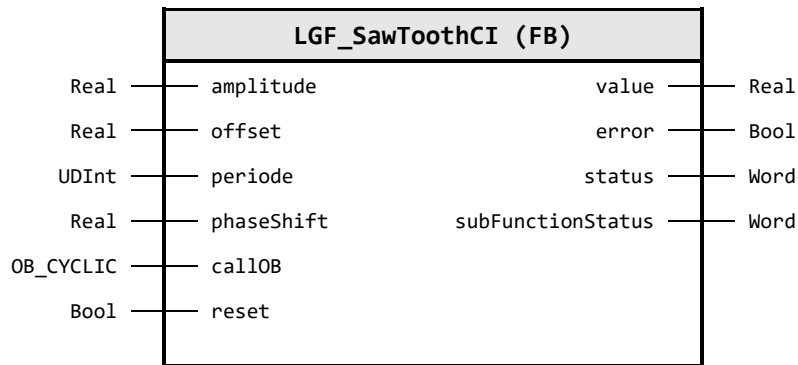
### 4.11.5 LGF\_SawToothCI (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates a sawtooth-shaped signal profile. For this it uses the time interval of the calling Cyclic Interrupt OB.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
amplitude	Real	1.0	Amplitude of the signal profile.
offset	Real	0.0	Offset of the signal profile in the Y-direction.
periode	UDInt	1000	Period duration of the signal profile in [ms]
phaseShift	Real	0.0	Phase offset in [ms]
callOB	OB_CYCLIC	---	Calling cyclic interrupt OB
reset	Bool	FALSE	Reset of the signal profile.

#### Output parameter

Identifier	Data type	Description
value	Real	Current value of the sawtooth signal.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8600	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.
16#8601	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

## Functional description

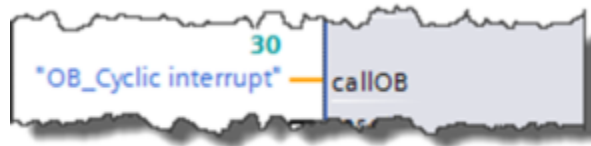
**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The block calculates the values for a sawtooth-shaped signal profile, which is output to the output parameter `value`.

The `amplitude`, the `offset` in the Y-direction, the `period`, and the `phase shift` can be set at the input parameters.

The input parameter `reset` resets the signal profile. At the `value` output parameter, the value 0 is output as long as `reset` is set to `TRUE`.

The block must be called in a cyclic interrupt OB. The time interval of the calling cyclic interrupt OB is determined in the FB with the command `QRY_CINT`. For this, the constant name of the calling cyclic interrupt OB must be interconnected at the input parameter `callOB`.

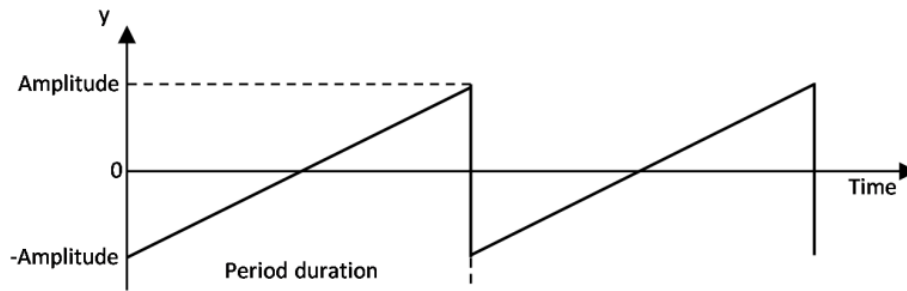


The number of calculated values of the signal profile per period duration is calculated as follows:

$$QuantityValues = \frac{Periodduration}{TimeintervakCyclicinterruptOB}$$

**Note** To obtain a continuous signal profile of the curve, the time interval of the cyclic interrupt OB should not be selected too large depending on the period duration.

The Figure below shows the signal profile of the calculated values.



**Change log**

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.06</b> 23.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added phase shift availability added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

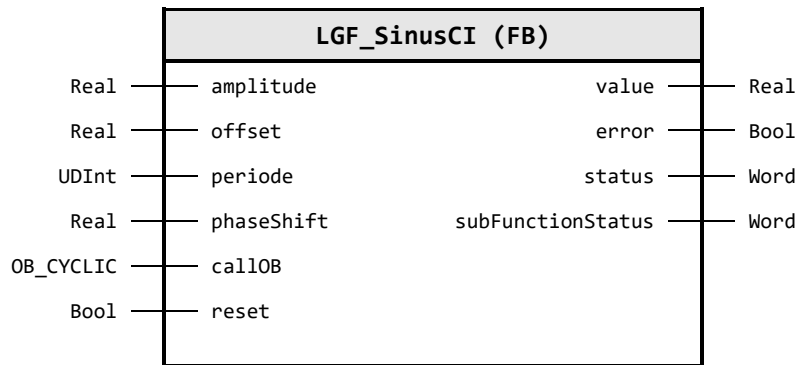
### 4.11.6 LGF\_SinusCI (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates a sinusoidal signal profile. For this it uses the time interval of the calling Cyclic Interrupt OB.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
amplitude	Real	1.0	Amplitude of the signal profile.
offset	Real	0.0	Offset of the signal profile in the Y-direction.
periode	UDInt	1000	Period duration of the signal profile in [ms]
phaseShift	Real	0.0	Phase offset in [ms]
callOB	OB_CYCLIC	---	Calling cyclic interrupt OB
reset	Bool	FALSE	Reset of the signal profile.

#### Output parameter

Identifier	Data type	Description
value	Real	Current value of the sinusoidal signal.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8600	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.
16#8601	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

## Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The block calculates the values for a sinusoidal signal profile, which is output to the output parameter `value`.

The `amplitude`, the `offset` in the Y-direction, the `period`, and the `phase shift` can be set at the input parameters.

The input parameter `reset` resets the signal profile. At the `value` output parameter, the value `0` is output as long as `reset` is set to `TRUE`.

The block must be called in a cyclic interrupt OB. The time interval of the calling cyclic interrupt OB is determined in the FB with the command `QRY_CINT`. For this, the constant name of the calling cyclic interrupt OB must be interconnected at the input parameter `callOB`.

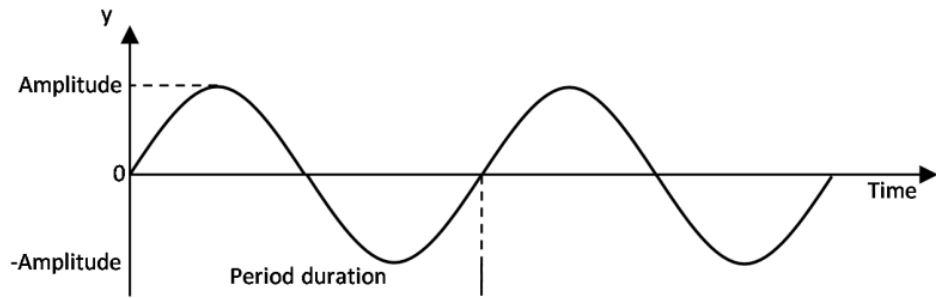


The number of calculated values of the signal profile per period duration is calculated as follows:

$$QuantityValues = \frac{Periodduration}{TimeintervakCyclicinterruptOB}$$

**Note** To obtain a continuous signal profile of the curve, the time interval of the cyclic interrupt OB should not be selected too large depending on the period duration.

The Figure below shows the signal profile of the calculated values.



### Change log

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 26.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added phase shift availability added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

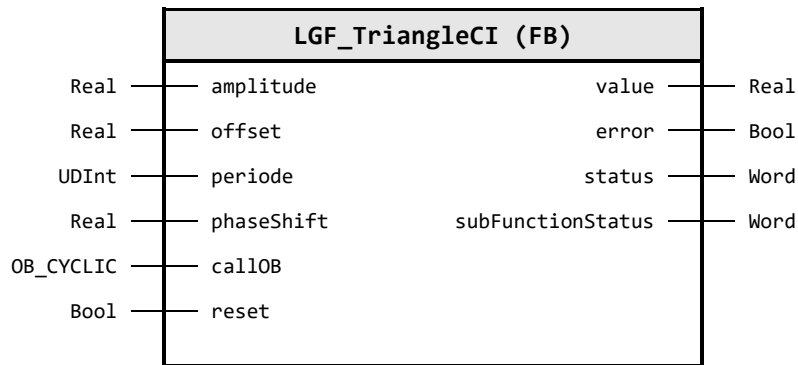
### 4.11.7 LGF\_TriangleCI (FB / V3.0.1)

Author: Siemens Industry Support

#### Short description

This function generates a triangular signal profile. For this it uses the time interval of the calling Cyclic Interrupt OB.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
amplitude	Real	1.0	Amplitude of the signal profile.
offset	Real	0.0	Offset of the signal profile in the Y-direction.
periode	UDInt	1000	Period duration of the signal profile in [ms]
phaseShift	Real	0.0	Phase offset in [ms]
callOB	OB_CYCLIC	---	Calling cyclic interrupt OB
reset	Bool	FALSE	Reset of the signal profile.

#### Output parameter

Identifier	Data type	Description
value	Real	Current value of the triangular signal.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8600	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.
16#8601	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

## Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The block calculates the values for a triangular signal profile, which is output to the output parameter `value`.

The `amplitude`, the `offset` in the Y-direction, the `period`, and the `phase shift` can be set at the input parameters.

The input parameter `reset` resets the signal profile. At the `value` output parameter, the value 0 is output as long as `reset` is set to `TRUE`.

The block must be called in a cyclic interrupt OB. The time interval of the calling cyclic interrupt OB is determined in the FB with the command `QRY_CINT`. For this, the constant name of the calling cyclic interrupt OB must be interconnected at the input parameter `callOB`.



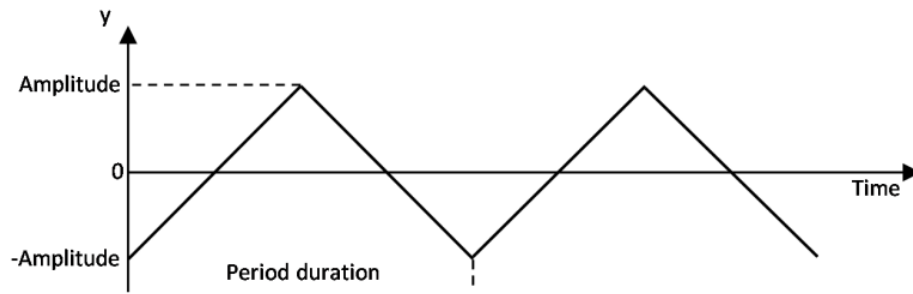
The number of calculated values of the signal profile per period duration is calculated as follows:

$$QuantityValues = \frac{Periodduration}{TimeintervakCyclicinterruptOB}$$

**Note** To obtain a continuous signal profile of the curve, the time interval of the cyclic interrupt OB should not be selected too large depending on the period duration.



The Figure below shows the signal profile of the calculated values.



**Change log**

Version & Date	Change description
<b>01.00.00</b> 03.07.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.03</b> 26.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added phase shift availability added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

## 4.12 Technology operations

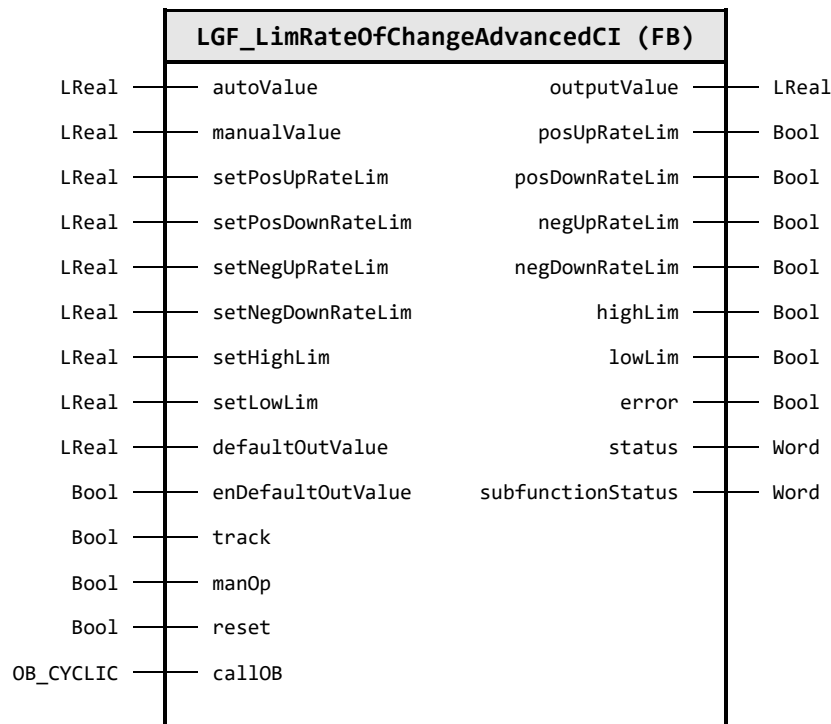
### 4.12.1 LGF\_LimRateOfChangeAdvancedCI (FB / V3.0.1)

Author: Siemens Digital Industries

#### Short description

The function `LGF_LimRateOfChangeAdvanced` limits the rate of change of an input variable. Jump functions become ramp functions. In addition, the block has various operating modes.

#### Block Interface



© Siemens AG 2021. All rights reserved.

#### Input parameter

Identifier	Data type	Default value	Description
autoValue	LReal	0.0	Signal to be processed and limited in its rate of change
manualValue	LReal	0.0	Manually controlled output value ('outputValue' = 'manualValue')
setPosUpRateLim	LReal	0.0	Rate of change per second for the rising ramp in the positive value range (1/second)
setPosDownRateLim	LReal	0.0	Rate of change per second for the falling ramp in the positive value range (1/second)
setNegUpRateLim	LReal	0.0	Rate of change per second for the rising ramp in the negative value range (1/second)
setNegDownRateLim	LReal	0.0	Rate of change per second for the falling ramp in the negative value range (1/second)
setHighLim	LReal	0.0	High limit value
setLowLim	LReal	0.0	Low limit value

#### 4 Program blocks

Identifier	Data type	Default value	Description
defaultOutValue	LReal	0.0	Value for pre-assignment of the output variable ('outputValue' = 'defaultOutValue')
enDefaultOutValue	Bool	FALSE	Assign default output value ('outputValue' = 'defaultOutValue')
track	Bool	FALSE	Follow / tracking of Input variable ('outputValue' = 'autoValue')
manOp	Bool	FALSE	Manual mode on ('outputValue' = 'manualValue')
reset	Bool	FALSE	Complete restart of function
callOB	OB_CYCLIC	---	Calling wake-alarm interrupt OB (cyclic interrupt OB)

#### Output parameter

Identifier	Data type	Description
outputValue	LReal	Output variable
posUpRateLim	Bool	Rise limitation in positive range tripped
posDownRateLim	Bool	Down rate limit in positive range reached
negUpRateLim	Bool	Up rate limit in negative range reached
negDownRateLim	Bool	Down rate limit in negative range reached
highLim	Bool	High limit reached
lowLim	Bool	Low limit reached
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8200	ERR_NEG_RATE_LIM Error: High limit lower than low limit. The high limit 'setHighLim' must be greater than the low limit 'setLowLim'.
16#8202	ERR_NEG_RATE_OF_CHANGE Error: Negative rate of change. The parameter for the change rate must not be negative.
16#8600	ERR_QRY_CINT Error in 'QRY_CINT' command - check 'subFunctionStatus' code
16#8601	ERR_OB_UNAVAILABLE Error: OB on input 'callOB' is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input 'callOB'.

## Functional description

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

For the positive/negative value range, two rates of change in each case for the ramp (rising and falling values) can be parameterized. The following operating modes can be selected via control inputs:

- Restart
- Pre-assigning an output
- Normal operation (automatic)
- Switch through controlled variable (manual)
- Tracking

The output variable can be limited through two parameterizable limits. An active limitation of the rate of change of a ramp, as well as an active limitation of the output variable are reported via outputs.

The time interval of the calling cyclic interrupt OB is determined by interconnecting the calling cyclic interrupt OB at the input parameter `callOB`.



### Restart

At restart `reset = TRUE`, the output `outputValue` is reset to `0.0`.

If `enDefaultOutValue = TRUE` is set, `defaultOutValue` is output. All signal outputs are set to `FALSE`.

### Pre-assigning an output

If `enDefaultOutValue = TRUE` is set, the value at `defaultOutValue` is output. When changing from `TRUE` to `FALSE`, `outputValue` is ramped from `defaultOutValue` to `autoValue`. When changing from `FALSE` to `TRUE`, the output `outputValue` immediately jumps to `defaultOutValue`.

### Normal operation

The ramps are straight lines of limitation and are based on a rate of change per second; if, for example, the parameter `setPosUpRateLim = 10.0` is assigned, then at a sampling time of `1s/100ms/10ms`, `10.0/1.0/0.1` will be added to `outputValue` at each block call, if `autoValue > outputValue`, until `autoValue` is reached.

The limitation of the rate of change can be parameterized in both positive and negative ranges for the increase and decrease.

Table: Marking of the ramps

Parameters	Ramp
setPosUpRateLim	outputValue > 0.0 and  outputValue  rising
setPosDownRateLim	outputValue > 0.0 and  outputValue  falling
setNegUpRateLim	outputValue < 0.0 and  outputValue  rising
setNegDownRateLim	outputValue < 0.0 and  outputValue  falling

If the ramps are not parameterized (setPosUpRateLim, setPosDownRateLim, setNegUpRateLim, and setNegDownRateLim equal 0.0), the output remains at 0.0 and normal operation is disabled.

### Tracking

If the input track = TRUE is set, the input variable autoValue is interconnected directly to the output variable outputValue. Thus, jumps of the input variable will also be output.

### Switch through controlled variable

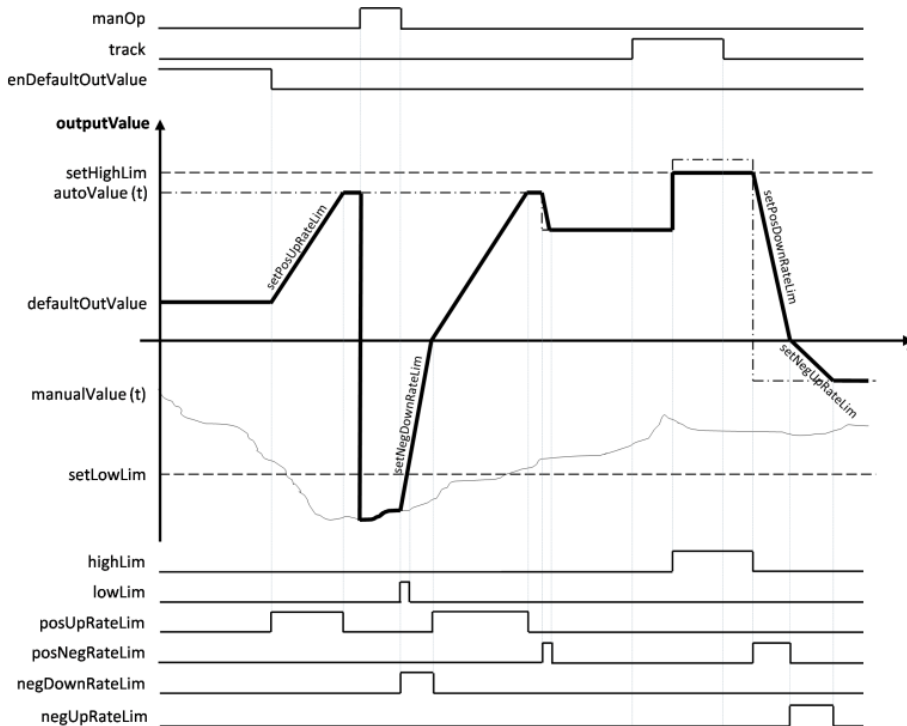
If manOp = TRUE is set, the controlled variable manualValue is interconnected directly to the output variable outputValue.

In this operating mode, the parameterization of the ramps or the high/low limitation of the output variable, and the pre-assignment of the output, are ineffective.

When changing from TRUE to FALSE, the output outputValue is ramped again after autoValue.

As soon as the value range between the low and high limits is reached, the high and low limits are reactivated.

Figure: Ramp function sequence, operating modes



**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 21.06.2016	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.07</b> 15.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 22.03.2021	<b>Simatic Systems Support</b> Insert documentation

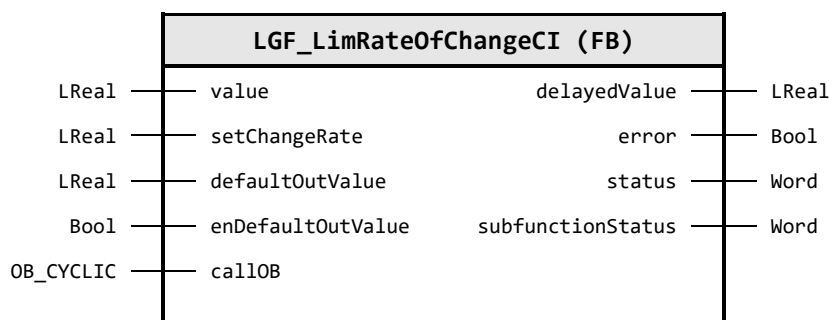
## 4.12.2 LGF\_LimRateOfChangeCI (FB / V3.0.1)

Author: Siemens Digital Industries

### Short description

This function limits the rate of change of an input variable. A jump function becomes a ramp function.

### Block Interface



### Input parameter

Identifier	Data type	Default value	Description
value	LReal	0.0	Signal to be processed and limited in its rate of change
setChangeRate	LReal	0.0	Rate of change of ramp function (1/second)
defaultOutValue	LReal	0.0	Value for pre-assignment of the output variable (`outputValue` = `defaultOutValue`)
enDefaultOutValue	Bool	FALSE	Assign default output value (`outputValue` = `defaultOutValue`)
callOB	OB_CYCLIC	---	Calling wake-alarm interrupt OB (cyclic interrupt OB)

### Output parameter

Identifier	Data type	Description
delayedValue	LReal	Output variable
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8200	ERR_NEG_RATE_LIM Error: Negative rate of change. The parameter for the change rate must not be negative.
16#8600	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

Code / Value	Identifier / Description
16#8601	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.

**Functional description**

**Note** The status of called commands is output in `subFunctionStatus`. In this case, the output value in `status` indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

The ramp is a limit line and refers to a rate of change per second; if, for example, `setChangeRate = 10.0` is parameterized at a sampling time of 1s/100ms/10ms for every block call, then if `value > delayedValue`, 10.0/1.0/0.1 is added to `delayedValue` until `value` is reached.

The limitation of the rate of change applies to both positive and negative values for the rise and fall.

The output `delayedValue` can be preset or initialized.

The time interval of the calling cyclic interrupt OB is determined by interconnecting the calling cyclic interrupt OB at the input parameter `callOB`.



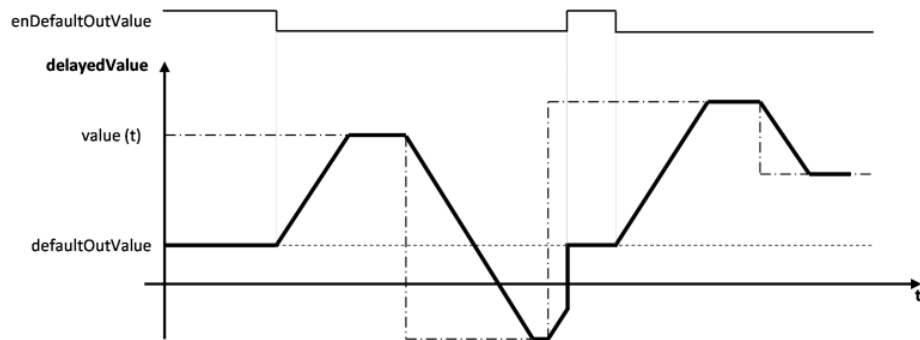
**Pre-assigning an output**

If `enDefaultOutValue = TRUE` is set, the value at `defaultOutValue` is output. When changing from `TRUE` to `FALSE`, the output `delayedValue` is ramped from `defaultOutValue` to `value`. When changing from `FALSE` to `TRUE`, the output `delayedValue` immediately jumps to `defaultOutValue`.



### Functional processes

The Figure below shows the ramp function sequence:



### Change log

Version & Date	Change description
<b>01.00.00</b> 21.06.2016	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.06</b> 15.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 22.03.2021	<b>Simatic Systems Support</b> Insert documentation

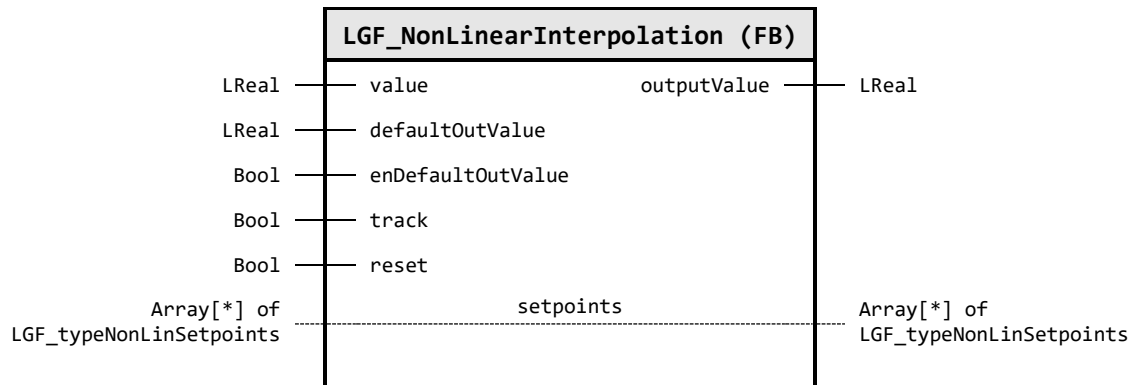
### 4.12.3 LGF\_NonLinearInterpolation (FB / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function implements a characteristic curve. The characteristic curve is defined via an interpolation point table with linear interpolation between the interpolation points. A prescribed input value generates an output value in each cycle based on the characteristic curve from the interpolation point table.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
value	LReal	0.0	Input value for calculating the output value over the defined characteristic curve.
defaultOutValue	LReal	0.0	Value for pre-assignment of the output variable ('outputValue' = 'defaultOutValue')
enDefaultOutValue	Bool	FALSE	Assign default output value ('outputValue' = 'defaultOutValue')
track	Bool	FALSE	The value of the output 'outputValue' follows the value of the input 'value' without using the characteristic curve as long as this input is set. ('outputValue' = 'value')
reset	Bool	FALSE	If the interpolation point table is changed in running operation, the input 'reset' must be activated afterwards. Otherwise, the block cannot guarantee correct execution. ('outputValue' = 0.0)

#### Output parameter

Identifier	Data type	Description
outputValue	LReal	The output value that has been calculated from the input value over the defined characteristic curve.

#### In/Out parameter

Identifier	Data type	Description
setpoints	Array[*] of LGF_typeNonLinSetpoints	Setpoint point table for defining the characteristic curve (polynomial)

**User defined datatype(s)****LGF\_typeNonLinSetpoints (UDT)**

Data type to setup a setpoint table for the function `LGF_NonLinearInterpolation`

Identifier	Data type	Default value	Description
inputValue	LReal	0.0	Input value to be interpolated
outputValue	LReal	0.0	Corresponding interpolated value

**Functional description**

The value of the output `outputValue` based on the following priority:

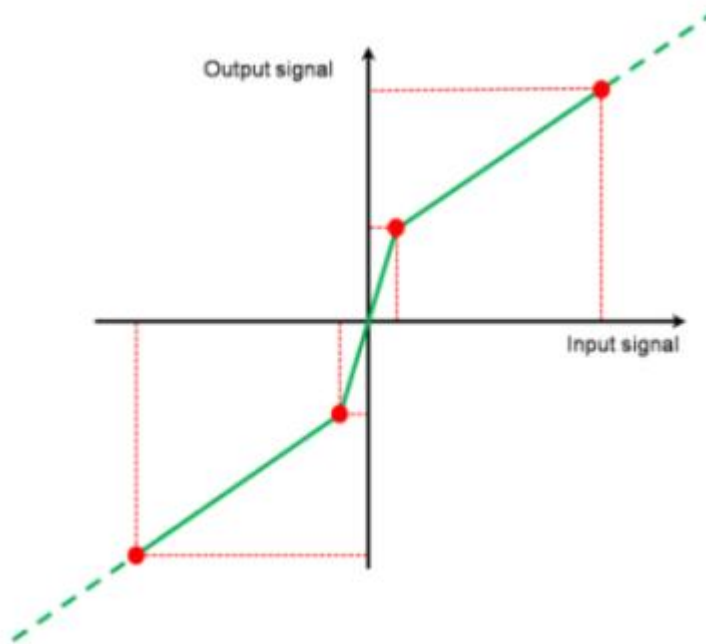
1. As long as the input `enDefaultOutValue` is set, the value defined via the parameter `defaultOutValue` will be output as output value.
2. As long as the input `reset` is set, the block is reset and the output value is `0.0`.
3. If the input `track` is set, the output value will be output directly as input value, without consideration of the characteristic curve.
4. Based on the input value, a characteristic curve value is calculated via the linearly interpolated, interpolation point table and output as an output value.
  - If the input value is between two interpolation points within the interpolation point table, the output value is calculated as the intersection with the connecting line between the preceding and following interpolation points (see Figure below).
  - If the input value is before the first interpolation point (lowest value defined in the interpolation point table), the output value will be calculated as the intersection of the line formed by the first two interpolation points of the interpolation point table.
  - If the input value is after the last interpolation point (highest value defined in the interpolation point table), the output value will be calculated as the intersection of the line formed by the last two interpolation points of the interpolation point table.

**Interpolation point table**

The interpolation point table is implemented through a variable of the data type Array. The type of the array corresponds to the PLC data type `LGF_typeNonLinSetpoints`.

You can create the interpolation point table in any global data block. The size of the array depends on the number of interpolation points.

Figure: Sample path of the output signal



**NOTICE**

To keep the computing time of the block as short as possible, there is no check of the parameterization or the data of the interpolation point table.

When entering the interpolation points in the interpolation point table, the following particularities must be considered. If these particularities are not taken into account, it can lead to a malfunction of the block.

- At least two interpolation points must be entered in the interpolation point table.
- The interpolation points in the interpolation point table must be entered in the Table in ascending order of the input values.

**Example**

Figure: Sample data block

Name	Data type	Default value
Static		
statTimeZone	TimeTransformationRule	
Bias	Int	0
DaylightBias	Int	60
DaylightStartMonth	USInt	3
DaylightStartWeek	USInt	5
DaylightStartWeekday	USInt	1
DaylightStartHour	USInt	2
DaylightStartMinute	USInt	0
StandardStartMonth	USInt	10
StandardStartWeek	USInt	5
StandardStartWeekday	USInt	1
StandardStartHour	USInt	3
StandardStartMinute	USInt	0
TimeZoneName	String[80]	'not even set ...

**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 04.01.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.06</b> 15.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 22.03.2021	<b>Simatic Systems Support</b> Insert documentation

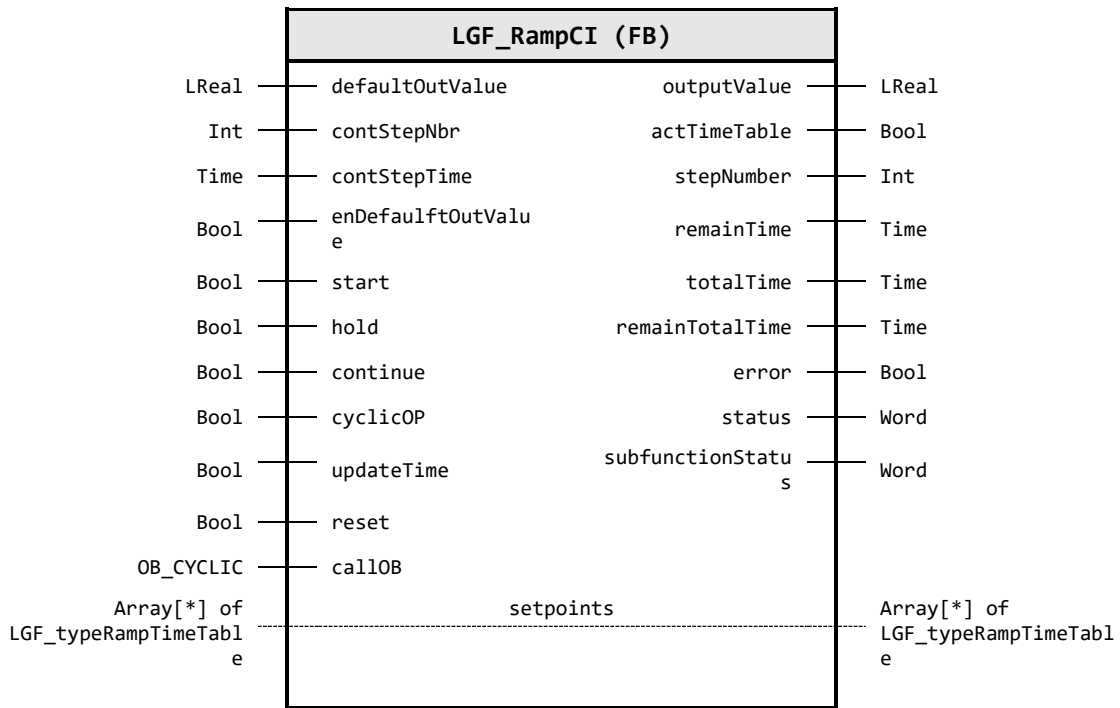
### 4.12.4 LGF\_RampCI (FB / V3.0.1)

Author: Siemens Digital Industries

#### Short description

The function generates a speed curve based on an interpolation point table. Linear interpolation occurs between the points within the prescribed time.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
defaultOutValue	LReal	0.0	Value for pre-assignment of the output variable (`outputValue` = `defaultOutValue`)
contStepNbr	Int	0	Number of the next interpolation point for continuing
contStepTime	Time	T#0MS	Remaining time to continue to the interpolation point `contStepNbr`
enDefaultOutValue	Bool	FALSE	Assign default output value (`outputValue` = `defaultOutValue`)
start	Bool	FALSE	Run down the interpolation point table
hold	Bool	FALSE	Freeze/ hold output at actual value
continue	Bool	FALSE	Continuing
cyclicOP	Bool	FALSE	Repeat interpolation point table cyclically
updateTime	Bool	FALSE	Update time values
reset	Bool	FALSE	Complete reset of function
callOB	OB_CYCLIC	---	Calling wake-alarm interrupt OB (cyclic interrupt OB)

**Output parameter**

Identifier	Data type	Description
outputValue	LReal	Output value
actTimeTable	Bool	The interpolation point table will be edited.
stepNumber	Int	current interpolation point number (interpolation point that is approached)
remainTime	Time	Remaining time until reaching the next interpolation point
totalTime	Time	Total time for setpoint table
remainTotalTime	Time	Total remaining time
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks

**In/Out parameter**

Identifier	Data type	Description
setpoints	Array[*] of LGF_typeRampTimeTable	Interpolation point table. You can find information on the data type `LGF_typeRampTimeTable` under the item "Global data".

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: Block is not being processed
16#7001	STATUS_FIRST_CALL Status: First call, Rising edge `start`.
16#7002	STATUS_FURTHER_CALLS Status: Further calls, cyclic operation, Input `cyclicOP` set.
16#8200	ERR_OB_UNAVAILABLE Error: OB on input `callOB` is not configured / present. Interconnect the constant name of a configured cyclic interrupt OB at the input `callOB`.
16#8201	ERR_ARRAY_LOWER_BOUND Error: Array does not start with 0 / Low array limit <= 0. The array with the interpolation points must start with the index 0.
16#8400	ERR_QRY_CINT Error in `QRY_CINT` command - check `subFunctionStatus` code

User defined datatype(s)

LGF\_typeRampTimeTable (UDT)

Data type to setup a speed curve based on a setpoint table for the function LGF\_RampCI

Identifier	Data type	Default value	Description
outputValue	LReal	0.0	Setpoint Value to reach by the interpolation curve
time	Time	T#0s	Time until the interpolation point is reached

Functional description

**Note** The status of called commands is output in subFunctionStatus. In this case, the output value in status indicates which command caused the error. In this case, refer to the TIA Portal Online Help section for information on the respective commands.

Global data

Together with the block, you automatically receive the PLC data type LGF\_typeRampTimeTable, which is composed of the parameters outVal for the value of a base point and time for the time, until the next base point is reached. The declaration takes place in a one-dimensional array of the data type LGF\_typeRampTimeTable beginning with the index 0. The array is created in a global data block and then passed to the module LGF\_RampCI.

Figure: Example of the declaration of the interpolation points

setpoints	Array[0..9] of *typeTimeTable*	
setpoints[0]	*typeTimeTable*	
outVal	Real	1.0
time	Time	t#5s
setpoints[1]	*typeTimeTable*	
outVal	Real	5.0
time	Time	t#3s
setpoints[2]	*typeTimeTable*	
setpoints[3]	*typeTimeTable*	

The parameter time of the last interpolation point must be parameterized with 0s, since there is no longer any successor interpolation point.

Principle of operation

With this block, speed curves can be executed based on parameterized interpolation points; in each call cycle values are output according to a schedule, and interpolation takes place between the interpolation points.

In each cycle the currently approached interpolation point number stepNumber, the actual remaining time remainTime until reaching the interpolation point, the total time totalTime, and the total remaining time until reaching the end of the speed curve remainTotalTime, are output. In addition, the output actTimeTable is set if the projected speed curve is currently being output.

The time interval of the calling cyclic interrupt OB is determined by interconnecting the calling cyclic interrupt OB at the input parameter callOB.

Figure: Interconnecting the cyclic interrupt OB





The following operating modes can be selected via control inputs:

- Restart
- Pre-assigning an output
- Output a speed curve
- Stop processing
- Specify processing step and processing time
- Switch-on cyclic operation+F1
- Update total time and remaining time

### Overview of the operating modes

Table: Overview of the operating modes

Operating mode	enDefaultOut Value	start	hold	continue	cyclic OP	updateTime	reset	Output/action
Restart							TRUE ↑	Block is initialized.
Pre-assigning an output	TRUE	TRUE					FALSE	defaultOutValue
Output a speed curve	FALSE	TRUE ↑	FALSE		FALSE		FALSE	outputValue(t); end value is held after processing
Stop speed curve	FALSE	TRUE	TRUE	FALSE			FALSE	current value of outputValue(t) is held
Specify processing step and processing time	FALSE	TRUE	TRUE	TRUE ↑			FALSE	outputValue(old)
			FALSE					Continue with parameterized interpolation point
Switch-on cyclic operation	FALSE	TRUE	FALSE		TRUE		FALSE	outputValue(t); after end of automatic restart
Update total time and remaining time						TRUE ↑	FALSE	Total time and remaining time are updated.

### Restart

The output `outValue` is reset to `0.0` with a rising edge at the input `reset`. With `enDefaultOutValue = TRUE`, `defaultOutValue` is output at `outputValue`. The total time and total remaining time are updated and output.

### Pre-assigning an output

If the speed curve should begin with a certain output value, then `enDefaultOutValue` must be `TRUE`. In this case the value `defaultOutValue` is present on the output of the timer. The internal processing of the speed curve continues during this time. If `enDefaultOutValue` changes to `FALSE` again, interpolation is performed to the currently active calibration point.

### Output a speed curve

With a rising edge at the input `start`, the speed curve is output - as long as `start` is `TRUE` or until the speed curve is terminated by reaching the last interpolation point. Through a subsequent rising edge, the speed curve is output again. In addition, the total time is updated at each switch-on.

### Switch-on cyclic operation

If, in addition to the input `start`, the input `cyclicOP` is also set to `TRUE`, the speed curve automatically returns to the start point after outputting the last interpolation point value and starts a new cycle.

There is no interpolation between the last interpolation point value and the starting point. The following must apply for a smooth transition: last interpolation point value = start point.

### Stop speed curve

With `hold = TRUE` the value of the output variable (including time processing) is frozen. When resetting `hold = FALSE`, the program continues at the point of interruption or at a parameterized point (see "Defining the processing step and processing time"). The processing time of the speed curve is extended by the holding time `T1*`. (see Figure below).

### Specify processing step and processing time

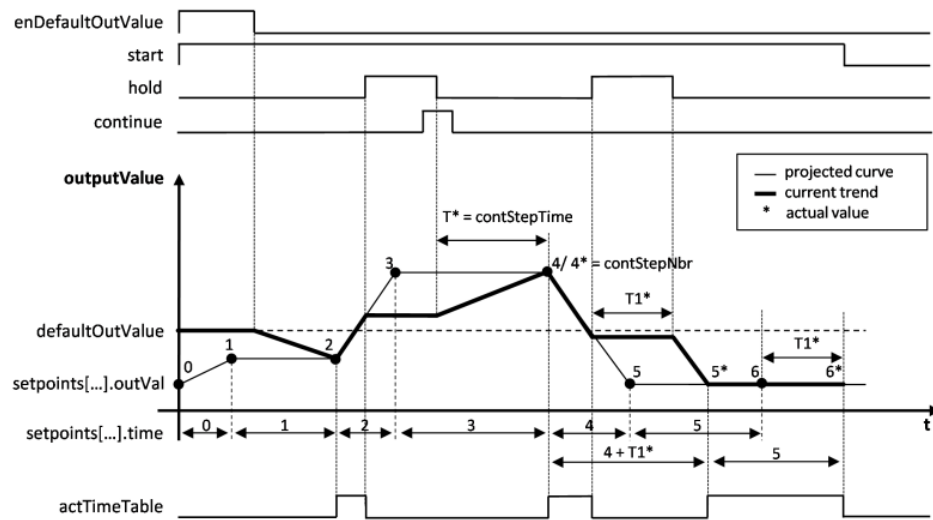
If the input parameter `continue` is set to `TRUE` for continuation while the speed curve is stopped (`hold = TRUE`), then after the input `hold` has been reset the interpolation point number `contStepNbr` (target interpolation point) will be approached within the time `contStepTime` (interpolation). The total remaining time will be recalculated.

### Updating total time and total remaining time

If values of the interpolation points are changed, the total time and the total remaining time of the speed curve can change. Since calculation of `totalTime` and `remainTotalTime` can significantly increase the processing time of the function block at many interpolation points, the calculation is only executed once with a rising edge on the `updateTime` input.

### Functional processes

Figure: Functional processes



### Change log

Version & Date	Change description
<b>01.00.00</b> 03.02.2017	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 16.05.2017	<b>Siemens Industry Online Support</b> Comment correction (REGION)
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.06</b> 15.11.2019	<b>Siemens Industry Presales Support</b> Code optimization, Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 22.03.2021	<b>Simatic Systems Support</b> Insert documentation Change UDT member name from `outValue` to `outputValue`

#### **4.12.5 Rules of simulated controlled systems**

In the entry “Closed-Loop Control of Simulated Controlled Systems” you will find the block library “LSim” for simulation of controlled systems for the controller families SIMATIC S7-1200 and SIMATIC S7-1500.

<https://support.industry.siemens.com/cs/ww/en/view/79047707>

## 4.13 Measurement operations

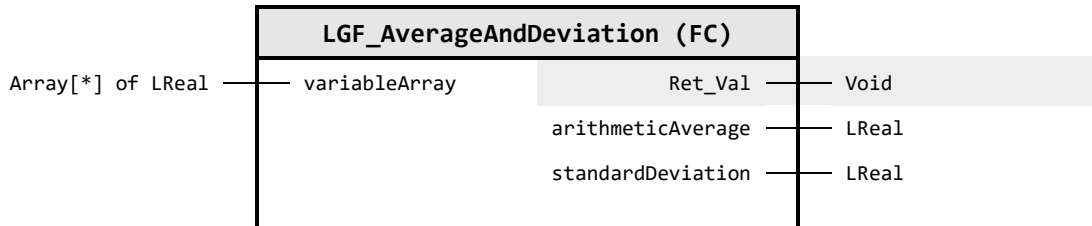
### 4.13.1 LGF\_AverageAndDeviation (FC / V3.0.1)

Author: Siemens Digital Industry

#### Short description

This function calculates the arithmetic mean and the standard deviation from a series of numbers.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
variableArray	Array[*] of LReal	Sequence of numbers to calculate with

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
arithmeticAverage	LReal	Calculated arithmetic average value
standardDeviation	LReal	Calculated standard deviation

#### Functional description

An array of any size is connected via the `variableArray` input. After reading-out the array boundaries, the arithmetic mean value and the standard deviation will be calculated from the values and both will be output.

#### Note

An array with too many elements can cause the cycle monitoring time to be exceeded.

## Change log

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 16.11.2015	<b>Siemens Industry Online Support</b> Bug fix at WRONG_TYPE: #error := true
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>02.00.00</b> 25.01.2019	<b>Simatic Systems Support</b> Data type changed from Variant to Array[*] of LReal
<b>02.00.01</b> 31.10.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.2 LGF\_DifferenceQuotientFC (FC / V3.0.1)

Author: Siemens Digital Industry

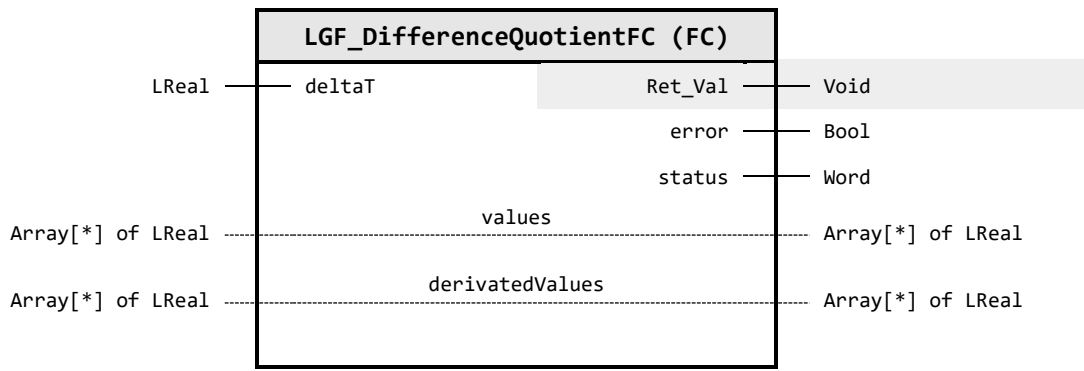
#### Short description

This function numerically differentiates a signal sampled equidistantly in time. For example, the velocity can be calculated from a measured locus curve, or the acceleration can be calculated from the measured velocity. In order to minimize the effects of a scattering measurement signal, this algorithm uses a compensating polynomial.

The function calculates the differentiated values acyclically.

The function reads an array that is differentiated.  $N - 4$  smoothed measured values can be calculated from  $N$  measured values. The output array contains the value 0 in the index (0,1,N-1,N). However, replacement values can be calculated.

#### Block Interface



#### Input parameter

Identifier	Data type	Description
deltaT	LReal	Equidistant distance between two measured values. (e.g. 1s)

#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
values	Array[*] of LReal	Values that will be included in the differentiation.
derivatedValues	Array[*] of LReal	The differentiated value range.

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8200	ERR_DELTA_T Error: Delta time `deltaT` must not be zero.
16#8400	ERR_ARRAYS_DIFFERENT Error: The Array sizes are not equal. The arrays `values` and `derivatedValues` must have the same size.
16#8401	ERR_NOT_ENOUGH_VALUES Error: Not enough values. The block requires five (5) values to calculate a differentiated value. Transfer additional values with a positive edge on the `insert` input.

**Functional description**

To calculate the difference quotient of a scattering signal, a third-degree compensation polynomial is first placed through the measured values. This polynomial is then differentiated. With this method, even a distorted input signal can be sensibly differentiated.

The difference quotient is calculated with the following formula:

$$y'(n) = \frac{y(n-2) - 8y(n-1) + 8y(n+1) - y(n+2)}{12 \cdot \text{deltaT}}$$

*deltaT*: equidistant distance between two measured values (e.g. 1s).

The function (FC) can calculate  $N - 4$  differentiated and smoothed measured values from  $N$  measured values. The output array would be assigned with 0 in the index (0,1,N-1,N). However, the following formalisms can be used to calculate substitute values:

$$y'(n-2) = \frac{-125(y(n-2) + 136y(n-1) + 48y(n) - 88y(n+1) + 29y(n+2))}{84 \cdot \text{deltaT}}$$

$$y'(n-1) = \frac{-38(y(n-2) - 2y(n-1) + 24y(n) + 26y(n+1) - 10y(n+2))}{84 \cdot \text{deltaT}}$$

$$y'(n+1) = \frac{10(y(n-2) - 26y(n-1) - 24y(n) + 2y(n+1) + 38y(n+2))}{84 \cdot \text{deltaT}}$$

$$y'(n+2) = \frac{-29(y(n-2) + 88y(n-1) - 48y(n) - 136y(n+1) + 125y(n+2))}{84 \cdot \text{deltaT}}$$

**Change log**

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 15.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation



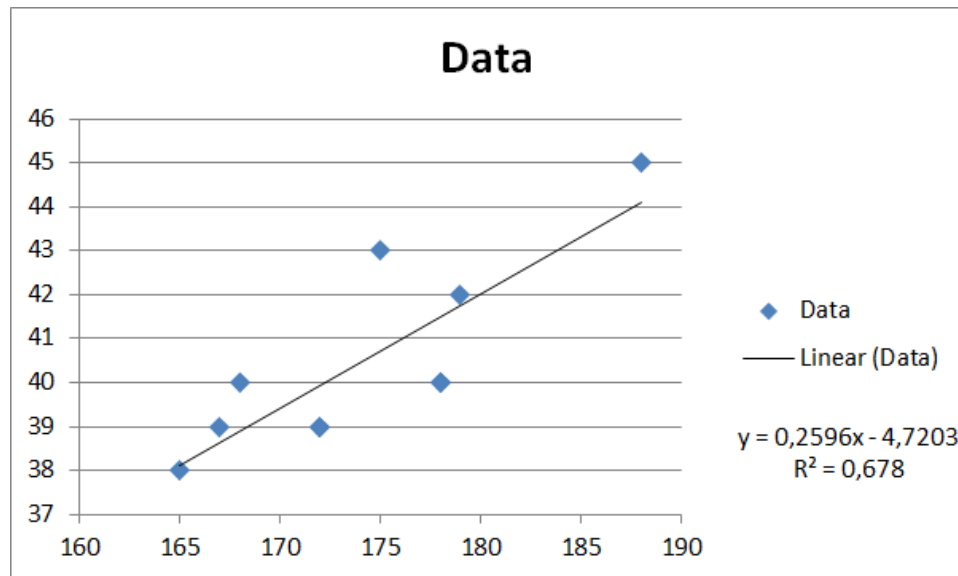
### 4.13.3 LGF\_RegressionLine (FC / V3.0.1)

Author: Siemens Digital Industry

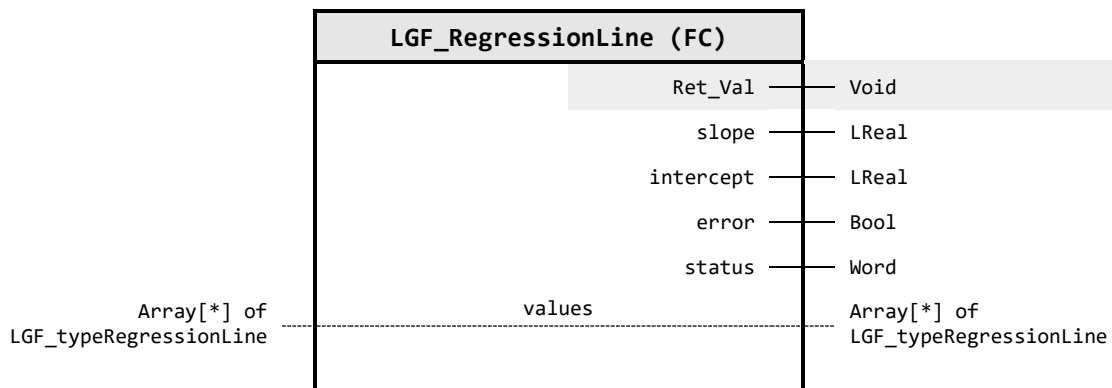
#### Short description

The simplest case of a regression is the regression line. This means that the assumed relationship between the input and output signal is a linear straight line.

Figure: Regression line



#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
slope	LReal	Gradient of straight line
intercept	LReal	The intersection with the Y axis
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of LGF_typeRegressionLine	The data points are transferred with their X- and Y-values. The data type `LGF_typeRegressionLine` has the following structure: <ul style="list-style-type: none"> <li>• x (Real)</li> <li>• y (Real)</li> </ul>

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#8200	ERR_NOT_ENOUGH_VALUES Error: Not enough Values. The block requires at least two pairs of values to calculate a regression line. Increase the size of the array at the input parameter `values` in the second dimension.

**User defined datatype(s)**

**LGF\_typeRegressionLine (UDT)**

The data type is for transferring datapoints (Key- Value pairs) to LGF\_RegressionLine and calculate the interpolated linear equation parameters slope and intercept.

Identifier	Data type	Default value	Description
x	Real	0.0	X-Axis value
y	Real	0.0	Y-Axis value

**Functional description**

The block calculates the regression line with the following line equation:

$$f(x) = m \cdot x + t$$

*m*: Gradient of straight line

*t*: Intersection with y-axis

*N*: number of array elements

The gradient *m* is calculated using the following equation:

$$m = \frac{n \cdot \sum_1^N (x(n) \cdot y(n)) - (\sum_1^N x(n)) \cdot \sum_1^N y(n)}{n \cdot \sum_1^N x^2(n) - (\sum_1^N x(n))^2}$$

The intersection *t* with the Y axis is calculated using the following equation:

$$t = \frac{\sum_1^N y(n)}{N} - b \cdot \frac{\sum_1^N x(n)}{N}$$

### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.04</b> 15.11.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.4 LGF\_SimpleSmoothingFC (FC / V3.0.1)

Author: Siemens Digital Industry

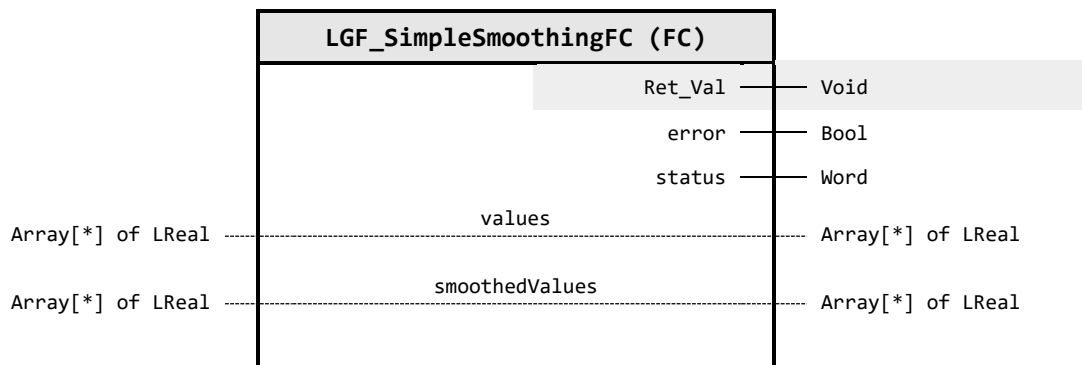
#### Short description

The function calculates the linear mean value acyclically.

The simplest form of smoothing a sequence of measured values is to calculate the linear mean value by three points.

The function reads an array that is smoothed.  $N - 2$  smoothed measured values can be calculated from  $N$  measured values. Therefore, the output array in the index (0) and index (N) contains the value 0.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
values	Array[*] of LReal	Values that are to be included in the smoothing.
smoothedValues	Array[*] of LReal	The smoothed values.

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8400	ERR_NOT_ENOUGH_VALUES Error: Not enough values. The block requires three (3) values to calculate a smoothed value. Increase the size of the array at the input parameter `values`. Adapt the array on the output parameter `smoothedValues` to the new size.

#### 4 Program blocks

Code / Value	Identifier / Description
16#8401	ERR_ARRAY_DIFFERENT Error: The Arraysizes are not equal. The arrays `values` and `smoothedValues` must have the same size.

#### Functional description

The function calculates the smoothed values using the following formula:

$$\overline{y(n)} = \frac{y(n-1) + y(n) + y(n+1)}{3}$$

The calculated value is output or the calculated values are output at output `smoothedValue`.

Based on this formula, the function cannot calculate values for the elements 0 and N.

#### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 15.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.5 LGF\_SmoothByPolynomFC (FC / V3.0.1)

Author: Siemens Digital Industry

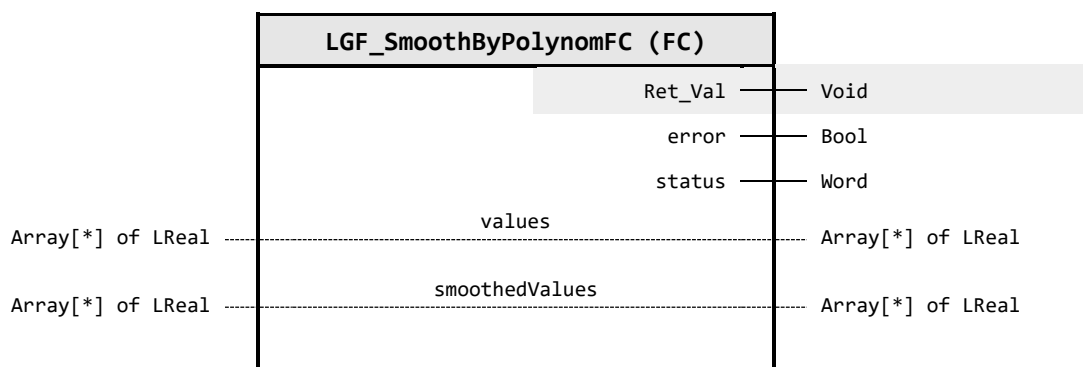
#### Short description

This function calculates the smoothed values by polynomial acyclically.

For smoothing, a 3rd degree polynomial is placed through five value points. The error squares of the distances between polynomial and real value are minimized. The smoothed values can be determined from the polynomial parameters obtained in this way.

The function reads an array that is smoothed.  $N - 4$  smoothed measured values can be calculated from  $N$  measured values. The output array contains the value 0 in the index (0,1,N-1,N). However, replacement values can be calculated.

#### Block Interface



#### Output parameter

Identifier	Data type	Description
Ret_Val	Void	Void - Function has no return value
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### In/Out parameter

Identifier	Data type	Description
values	Array[*] of LReal	Values that are to be included in the smoothing.
smoothedValues	Array[*] of LReal	The smoothed values.

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8400	ERR_ARRAYS_DIFFERENT Error: The Array sizes are not equal. The arrays `values` and `smoothedValues` must have the same size.

Code / Value	Identifier / Description
16#8401	ERR_NOT_ENOUGH_VALUES Error: Not enough values. The block requires five (5) values to calculate a smoothed value. Increase the size of the array at the input parameter `values`. Adapt the array on the output parameter `smoothedValues` to the new size.

### Functional description

The 3rd degree compensation polynomial is calculated as follows:

$$\overline{y(n)} = \frac{1}{35} \cdot (-3 \cdot y(n-2) + 12 \cdot y(n-1) + 17 \cdot y(n) + 12 \cdot y(n+1) - 3 \cdot y(n+2))$$

$N - 4$  smoothed measured values can thus be calculated from the  $N$  measured values. The output array contains the value 0 in the index (0.1,  $N-1$ ,  $N$ ).

These "missing" values are calculated with the following formalisms:

$$\overline{y(n-2)} = \frac{1}{70} \cdot (69 \cdot y(n-2) + 4 \cdot y(n-1) - 6 \cdot y(n) + 4 \cdot y(n+1) - y(n+2))$$

$$\overline{y(n-1)} = \frac{2}{70} \cdot (2 \cdot y(n-2) + 27 \cdot y(n-1) + 12 \cdot y(n) - 8 \cdot y(n+1) + 2 \cdot y(n+2))$$

$$\overline{y(n+1)} = \frac{2}{70} \cdot (2 \cdot y(n-2) - 8 \cdot y(n-1) + 12 \cdot y(n) + 27 \cdot y(n+1) + 2 \cdot y(n+2))$$

$$\overline{y(n+2)} = \frac{1}{70} \cdot (-y(n-2) + 4 \cdot y(n-1) - 6 \cdot y(n) + 4 \cdot y(n+1) + 69 \cdot y(n+2))$$

### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.05</b> 15.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.6 LGF\_Boxplot\_DInt (FB / V3.0.1)

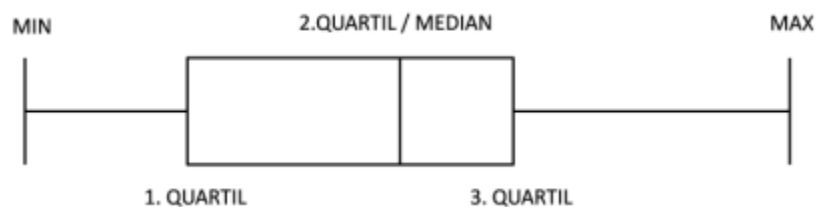
Author: Siemens Digital Industry

#### Short description

If you want to get an overview of existing data, you can use a Boxplot diagram. A Boxplot shows you in which area the data is located and how it is distributed over this area. A Boxplot consists of the following parameters:

- Minimum (smallest occurring value of the sample)
- Lower or first quartile (below this value are 25% of the sample values)
- Median or second quartile (below this value are 50% of the sample values)
- Upper or third quartile (below this value are 75% of the sample values)
- Maximum (largest occurring value of the sample)

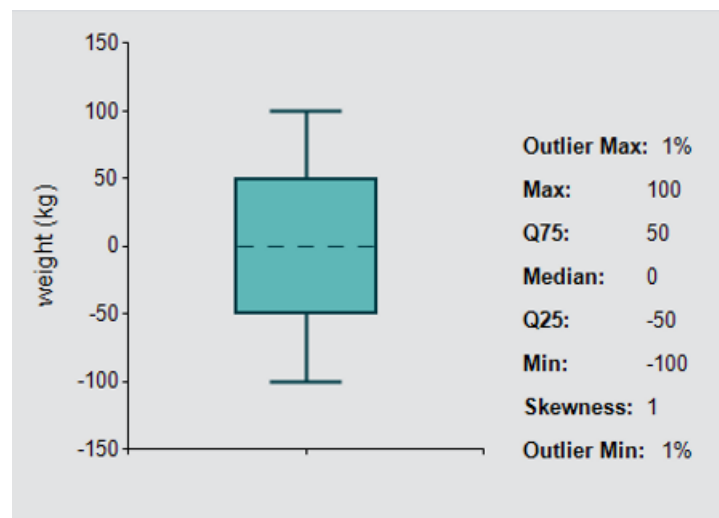
Figure: Boxplot



#### WinCC-Control

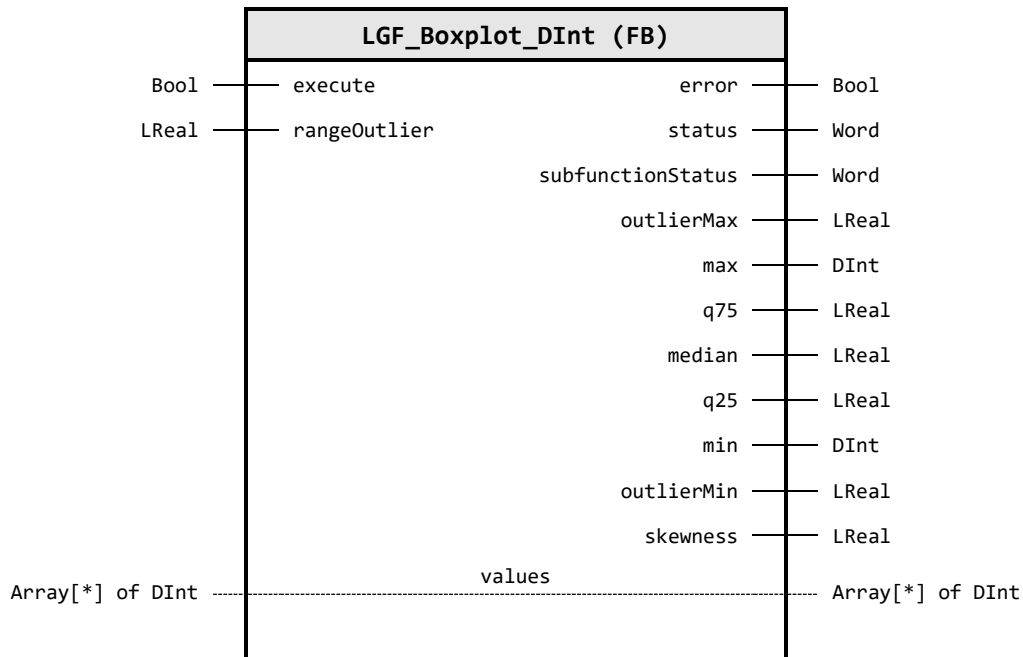
To visualize the Boxplot, the Siemens Industry Online Support offers you a Net-Control, which you can use in conjunction with WinCC Runtime Professional. You can find the controls in the `UserFiles` folder of this library.

Figure: .Net Control "Boxplot"





**Block Interface**



**Input parameter**

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Activation of the calculation with each positive edge.
rangeOutlier	LReal	1.5	Outlier detection: * 0: Outlier detection is deactivated * 0-1: Invalid value * >1: Outlier detection is activated.

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
outlierMax	LReal	Upper outliers in %.
max	DInt	Maximum Value, not an outlier.
q75	LReal	3rd quartile or Q75 of the data series.
median	LReal	2nd quartile or Median of the data series.
q25	LReal	1st quartile or Q25 of the data series.
min	DInt	Minimum Value, not an outlier.
outlierMin	LReal	Lower outliers in %.
skewness	LReal	Skewness of the data series.

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of DInt	The array containing the data series that is to be used for the calculation

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#8200	ERR_NEG_ARR_BOUND Error: Negative array boundary not allowed. Check the array at the input `values`.
16#8600	ERR_SHELL_SORT Error: Error in command `LGF_ShellSort_DInt`. Check `subFunctionStatus` code
16#9101	ERR_RANGE_NOT_OK Error: The parameter `rangeOutlier` type is invalid. Enter a valid `rangeOutlier` value for the parameter: * 0: Outlier detection is deactivated * >1 Valid value.

**Functional description**

The block sorts the data series and then calculates the so-called “five-point summary”:

Table: Five-point summary

Characteristic value of the five-point summary	Output parameter of the block
Minimum (smallest occurring value of the sample)	<code>min</code>
Lower or first quartile (below this value are 25% of the sample values)	<code>q25</code>
Median or second quartile (below this value are 50% of the sample values)	<code>median</code>
Upper or third quartile (below this value are 75% of the sample values)	<code>q75</code>
Maximum (largest occurring value of the sample)	<code>max</code>

If outlier detection is activated, the block first calculates the limits. From these limit values, the values are recognized as outliers:

$$Bound^{upper} = q_{75} + rangeOutlier \cdot (q_{75} - q_{25})$$

$$Bound^{lower} = q_{25} - rangeOutlier \cdot (q_{75} - q_{25})$$

The block then calculates new values for the parameters `max` and `min`, which lie within the outlier limits. The outliers are counted and output as a percentage.

To make it easier to judge how the data is distributed, the block also calculates the skew. The skewness lies between the values **-1** and **1** with the following meaning:

- **-1**: extremely left skewed distribution
- **0**: symmetrical distribution
- **1**: extreme right-skew distribution

The elements of the passed array are sorted in ascending order by the block. The `LGF_Shellsort_DInt` block is used for sorting.

The parameters are calculated as follows:

Table: Boxplot formulas Parameters Formula

Parameters	Formula
q25 (1st quartile)	$q_{25} = x_{(k)}$ with $k = \frac{\left\lceil \frac{1}{2}(n+1) \right\rceil + 1}{2} = \frac{n+3}{4}$
q50 (2nd quartile) median	$q_{50} = x_{\left(\frac{n+1}{2}\right)}$
q75 (3rd quartile)	$q_{75} = x_{(n+1-k)}$ with $(n+1-k) = \frac{3n+1}{4}$ <i>n</i> := number of samples (size of array) If the result of the element to be determined (from which the quartiles can be derived) is not an integer, the quartile is calculated from the linear fraction between the two adjacent samples.
skewness	$skewness = \frac{(q_{75} + q_{25}) - 2 \cdot q_{50}}{q_{75} - q_{25}}$ Note: This is just an approximation.

### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 05.11.2019	<b>Simatic Systems Support</b> Code reworked, regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.7 LGF\_Boxplot\_LReal (FB / V3.0.1)

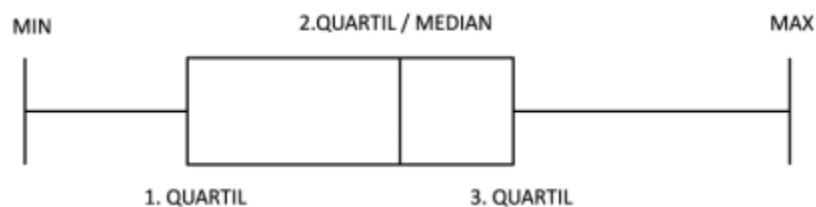
Author: Siemens Digital Industry

#### Short description

If you want to get an overview of existing data, you can use a Boxplot diagram. A Boxplot shows you in which area the data is located and how it is distributed over this area. A Boxplot consists of the following parameters:

- Minimum (smallest occurring value of the sample)
- Lower or first quartile (below this value are 25% of the sample values)
- Median or second quartile (below this value are 50% of the sample values)
- Upper or third quartile (below this value are 75% of the sample values)
- Maximum (largest occurring value of the sample)

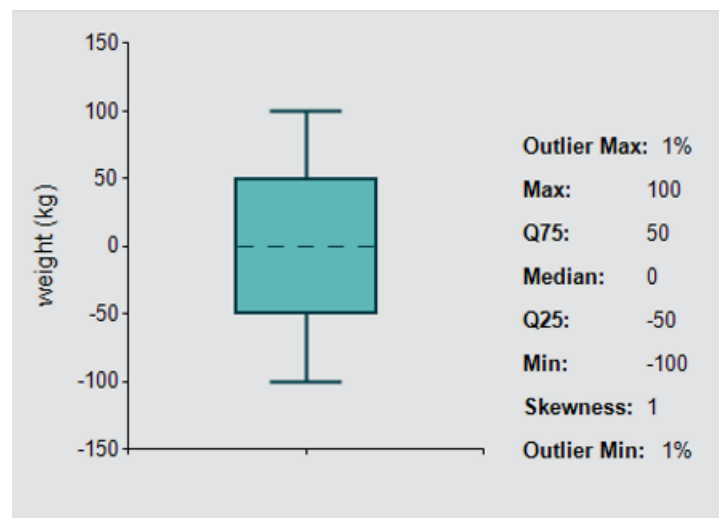
Figure: Boxplot



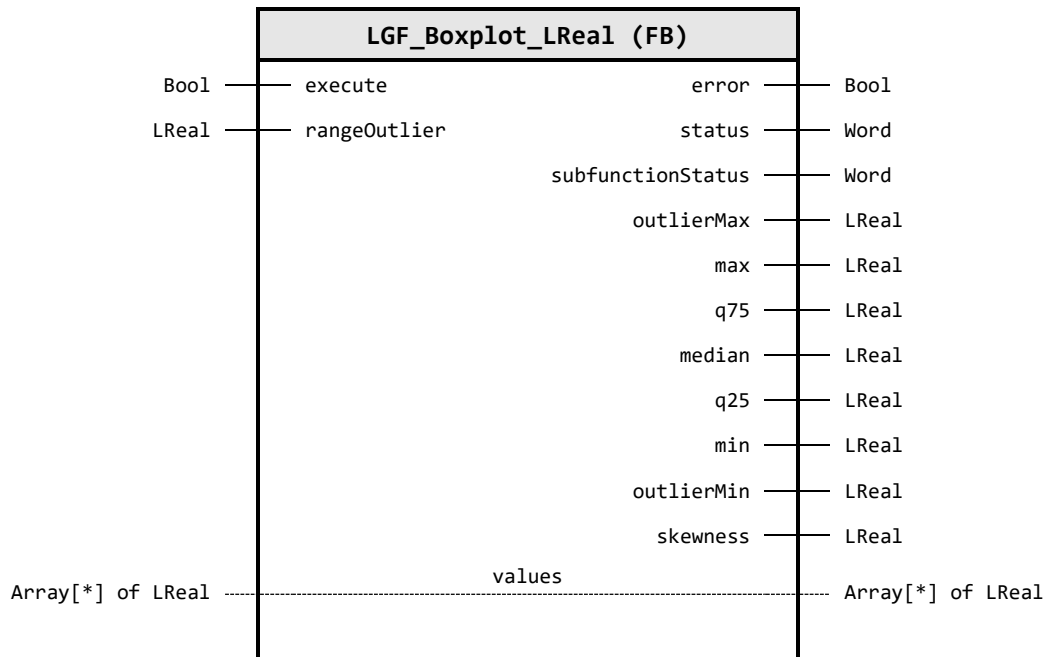
#### WinCC-Control

To visualize the Boxplot, the Siemens Industry Online Support offers you a Net-Control, which you can use in conjunction with WinCC Runtime Professional. You can find the controls in the `UserFiles` folder of this library.

Figure: .Net Control "Boxplot"



**Block Interface**



**Input parameter**

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Activation of the calculation with each positive edge.
rangeOutlier	LReal	1.5	Outlier detection: * 0: Outlier detection is deactivated * 0-1: Invalid value * >1: Outlier detection is activated.

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
outlierMax	LReal	Upper outliers in %.
max	LReal	Maximum Value, not an outlier.
q75	LReal	3rd quartile or Q75 of the data series.
median	LReal	2nd quartile or Median of the data series.
q25	LReal	1st quartile or Q25 of the data series.
min	LReal	Minimum Value, not an outlier.
outlierMin	LReal	Lower outliers in %.
skewness	LReal	Skewness of the data series.

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of LReal	The array containing the data series that is to be used for the calculation

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#8200	ERR_NEG_ARR_BOUND Error: Negative array boundary not allowed. Check the array at the input `values`.
16#8600	ERR_SHELL_SORT Error: Error in command `LGF_ShellSort_LReal`. Check `subFunctionStatus` code
16#9101	ERR_RANGE_NOT_OK Error: The parameter `rangeOutlier` type is invalid. Enter a valid `rangeOutlier` value for the parameter: * 0: Outlier detection is deactivated * >1 Valid value.

**Functional description**

The block sorts the data series and then calculates the so-called “five-point summary”:

Table: Five-point summary

Characteristic value of the five-point summary	Output parameter of the block
Minimum (smallest occurring value of the sample)	<code>min</code>
Lower or first quartile (below this value are 25% of the sample values)	<code>q25</code>
Median or second quartile (below this value are 50% of the sample values)	<code>median</code>
Upper or third quartile (below this value are 75% of the sample values)	<code>q75</code>
Maximum (largest occurring value of the sample)	<code>max</code>

If outlier detection is activated, the block first calculates the limits. From these limit values, the values are recognized as outliers:

$$Bound^{upper} = q_{75} + rangeOutlier \cdot (q_{75} - q_{25})$$

$$Bound^{lower} = q_{25} - rangeOutlier \cdot (q_{75} - q_{25})$$

The block then calculates new values for the parameters `max` and `min`, which lie within the outlier limits. The outliers are counted and output as a percentage.

## 4 Program blocks

To make it easier to judge how the data is distributed, the block also calculates the skew. The skewness lies between the values **-1** and **1** with the following meaning:

- **-1**: extremely left skewed distribution
- **0**: symmetrical distribution
- **1**: extreme right-skew distribution

The elements of the passed array are sorted in ascending order by the block. The `LGF_Shellsort_LReal` block is used for sorting.

The parameters are calculated as follows:

Table: Boxplot formulas Parameters Formula

Parameters	Formula
q25 (1st quartile)	$q_{25} = x_{(k)}$ with $k = \frac{\left\lfloor \frac{1}{2}(n+1) \right\rfloor + 1}{2} = \frac{n+3}{4}$
q50 (2nd quartile) median	$q_{50} = x_{\left(\frac{n+1}{2}\right)}$
q75 (3rd quartile)	$q_{75} = x_{(n+1-k)}$ with $(n+1-k) = \frac{3n+1}{4}$ <i>n</i> := number of samples (size of array) If the result of the element to be determined (from which the quartiles can be derived) is not an integer, the quartile is calculated from the linear fraction between the two adjacent samples.
skewness	$skewness = \frac{(q_{75} + q_{25}) - 2 \cdot q_{50}}{q_{75} - q_{25}}$ Note: This is just an approximation.

### Change log

Version & Date	Change description
<b>01.00.00</b> 05.11.2019	<b>Siemens Industry Online Support</b> First released version
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.8 LGF\_Boxplot\_UDInt (FB / V3.0.1)

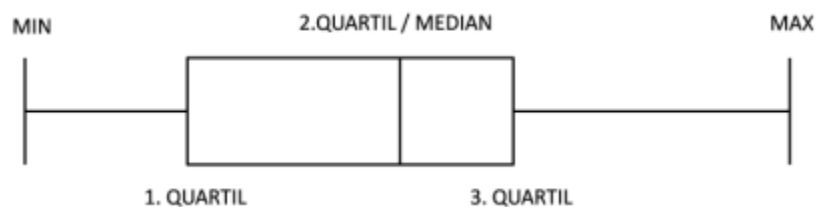
Author: Siemens Digital Industry

#### Short description

If you want to get an overview of existing data, you can use a Boxplot diagram. A Boxplot shows you in which area the data is located and how it is distributed over this area. A Boxplot consists of the following parameters:

- Minimum (smallest occurring value of the sample)
- Lower or first quartile (below this value are 25% of the sample values)
- Median or second quartile (below this value are 50% of the sample values)
- Upper or third quartile (below this value are 75% of the sample values)
- Maximum (largest occurring value of the sample)

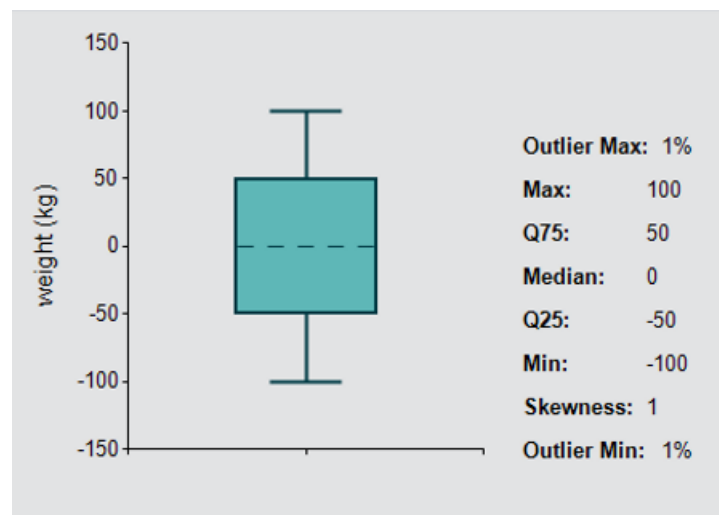
Figure: Boxplot



#### WinCC-Control

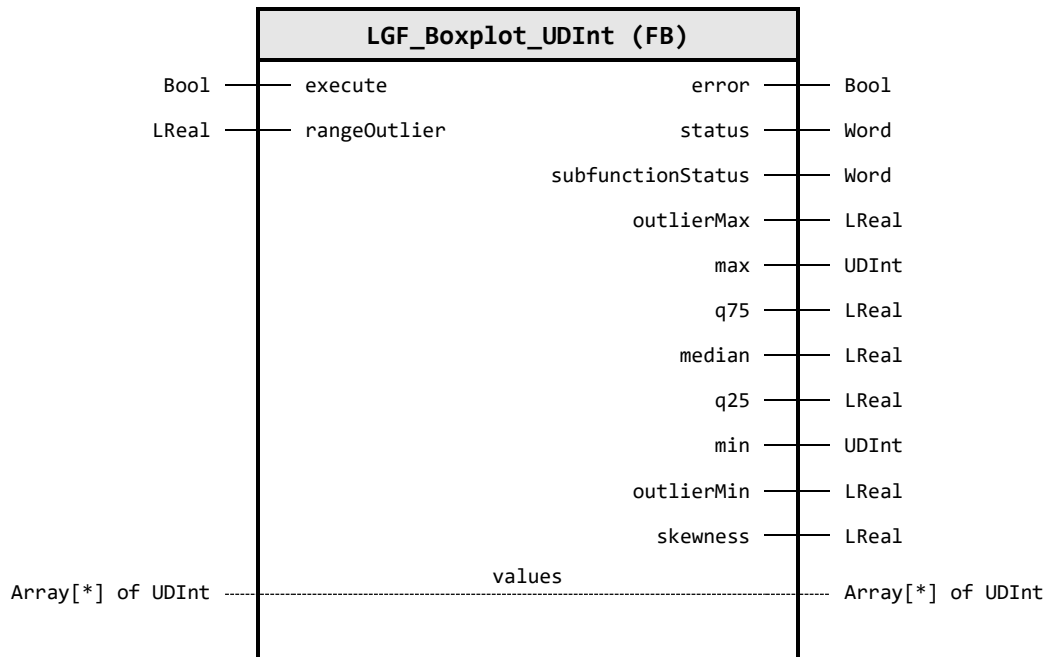
To visualize the Boxplot, the Siemens Industry Online Support offers you a Net-Control, which you can use in conjunction with WinCC Runtime Professional. You can find the controls in the `UserFiles` folder of this library.

Figure: .Net Control "Boxplot"





**Block Interface**



**Input parameter**

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Activation of the calculation with each positive edge.
rangeOutlier	LReal	1.5	Outlier detection: * 0: Outlier detection is deactivated * 0-1: Invalid value * >1: Outlier detection is activated.

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
outlierMax	LReal	Upper outliers in %.
max	UDInt	Maximum Value, not an outlier.
q75	LReal	3rd quartile or Q75 of the data series.
median	LReal	2nd quartile or Median of the data series.
q25	LReal	1st quartile or Q25 of the data series.
min	UDInt	Minimum Value, not an outlier.
outlierMin	LReal	Lower outliers in %.
skewness	LReal	Skewness of the data series.

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of UDInt	The array containing the data series that is to be used for the calculation

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#8200	ERR_NEG_ARR_BOUND Error: Negative array boundary not allowed. Check the array at the input `values`.
16#8600	ERR_SHELL_SORT Error: Error in command `LGF_ShellSort_UDInt`. Check `subFunctionStatus` code
16#9101	ERR_RANGE_NOT_OK Error: The parameter `rangeOutlier` type is invalid. Enter a valid `rangeOutlier` value for the parameter: * 0: Outlier detection is deactivated * >1 Valid value.

**Functional description**

The block sorts the data series and then calculates the so-called “five-point summary”:

Table: Five-point summary

Characteristic value of the five-point summary	Output parameter of the block
Minimum (smallest occurring value of the sample)	<code>min</code>
Lower or first quartile (below this value are 25% of the sample values)	<code>q25</code>
Median or second quartile (below this value are 50% of the sample values)	<code>median</code>
Upper or third quartile (below this value are 75% of the sample values)	<code>q75</code>
Maximum (largest occurring value of the sample)	<code>max</code>

If outlier detection is activated, the block first calculates the limits. From these limit values, the values are recognized as outliers:

$$Bound^{upper} = q_{75} + rangeOutlier \cdot (q_{75} - q_{25})$$

$$Bound^{lower} = q_{25} - rangeOutlier \cdot (q_{75} - q_{25})$$

The block then calculates new values for the parameters `max` and `min`, which lie within the outlier limits. The outliers are counted and output as a percentage.

## 4 Program blocks

To make it easier to judge how the data is distributed, the block also calculates the skew. The skewness lies between the values **-1** and **1** with the following meaning:

- **-1**: extremely left skewed distribution
- **0**: symmetrical distribution
- **1**: extreme right-skew distribution

The elements of the passed array are sorted in ascending order by the block. The `LGF_Shellsort_UDInt` block is used for sorting.

The parameters are calculated as follows:

Table: Boxplot formulas Parameters Formula

Parameters	Formula
q25 (1st quartile)	$q_{25} = x_{(k)} \quad \text{with} \quad k = \frac{\left\lceil \frac{1}{2}(n+1) \right\rceil + 1}{2} = \frac{n+3}{4}$
q50 (2nd quartile) median	$q_{50} = x_{\left(\frac{n+1}{2}\right)}$
q75 (3rd quartile)	$q_{75} = x_{(n+1-k)} \quad \text{with} \quad (n+1-k) = \frac{3n+1}{4}$ <i>n</i> := number of samples (size of array) If the result of the element to be determined (from which the quartiles can be derived) is not an integer, the quartile is calculated from the linear fraction between the two adjacent samples.
skewness	$skewness = \frac{(q_{75} + q_{25}) - 2 \cdot q_{50}}{q_{75} - q_{25}}$ Note: This is just an approximation.

### Change log

Version & Date	Change description
<b>01.00.00</b> 05.11.2019	<b>Siemens Industry Online Support</b> First released version
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.9 LGF\_DifferenceQuotientFB (FB / V3.0.1)

Author: Siemens Digital Industry

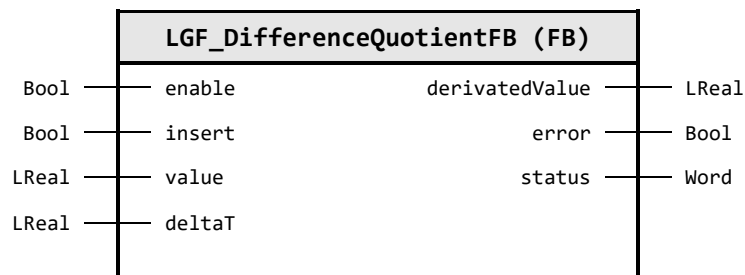
#### Short description

This function numerically differentiates a signal sampled equidistantly in time. For example, the velocity can be calculated from a measured locus curve, or the acceleration can be calculated from the measured velocity. In order to minimize the effects of a scattering measurement signal, this algorithm uses a compensating polynomial.

The function block calculates the differentiated values cyclically.

The function block reads-in a value with each positive edge on the `insert` been read in, the block calculates a differentiated value and outputs it.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
enable	Bool	FALSE	Activates the block. As long as enable is `TRUE`, the block can accept values on the parameter `value`.
insert	Bool	FALSE	Accepts the value at the input `value` at positive edge and outputs a `derivatedValue` if five values have been read in.
value	LReal	0.0	Value that must be included in the differentiation.
deltaT	LReal	0.0	Equidistant distance between two measured values. (e.g. 1s)

#### Output parameter

Identifier	Data type	Description
derivatedValue	LReal	The differentiated value.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling

Code / Value	Identifier / Description
16#7002	STATUS_SUBSEQUENT_CALL Status: Processing is active. Subsequent call of FB.
16#7010	STATUS_NOT_ENOUGH_VALUES Status: Not enough values. The block requires five (5) values to calculate a differentiated value. Transfer additional values with a positive edge on the `insert` input.
16#8200	ERR_DELTA_T Error: Delta time `deltaT` must not be zero.

### Functional description

To calculate the difference quotient of a scattering signal, a third-degree compensation polynomial is first placed through the measured values. This polynomial is then differentiated. With this method, even a distorted input signal can be sensibly differentiated.

The difference quotient is calculated with the following formula:

$$y'(n) = \frac{y(n-2) - 8y(n-1) + 8y(n+1) - y(n+2)}{12 \cdot \text{deltaT}}$$

*deltaT*: equidistant distance between two measured values (e.g. 1s).

The function (FC) can calculate  $N - 4$  differentiated and smoothed measured values from  $N$  measured values. The output array would be assigned with 0 in the index (0,1,N-1,N). However, the following formalisms can be used to calculate substitute values:

$$y'(n-2) = \frac{-125(y(n-2) + 136y(n-1) + 48y(n) - 88y(n+1) + 29y(n+2))}{84 \cdot \text{deltaT}}$$

$$y'(n-1) = \frac{-38(y(n-2) - 2y(n-1) + 24y(n) + 26y(n+1) - 10y(n+2))}{84 \cdot \text{deltaT}}$$

$$y'(n+1) = \frac{10(y(n-2) - 26y(n-1) - 24y(n) + 2y(n+1) + 38y(n+2))}{84 \cdot \text{deltaT}}$$

$$y'(n+2) = \frac{-29(y(n-2) + 88y(n-1) - 48y(n) - 136y(n+1) + 125y(n+2))}{84 \cdot \text{deltaT}}$$

### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 04.11.2019	<b>Simatic Systems Support</b> Code reworked. Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

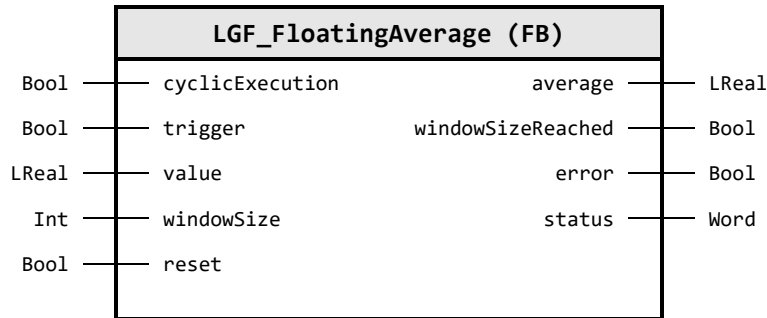
### 4.13.10 LGF\_FloatingAverage (FB / V3.0.2)

Author: Siemens Digital Industry

#### Short description

This function calculates a moving arithmetic mean value from REAL values. This method can be used to smooth data series. The values can be read in cyclically or triggered.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
cyclicExecution	Bool	false	TRUE: cyclic operation, trigger not in use
trigger	Bool	FALSE	Read in `value` with every pulse at input `trigger`
value	LReal	0.0	Value/s from which the moving average is to be determined.
windowSize	Int	100	Window length for sliding averaging in the range from 1..100. The standard value is 100.
reset	Bool	FALSE	TRUE: The block is reset and the calculation starts again.

#### Output parameter

Identifier	Data type	Description
average	LReal	Moving / Floating average
windowSizeReached	Bool	FALSE: Maximum window width not yet reached, TRUE: Maximum window width reached
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#0000	STATUS_FINISHED_NO_ERROR Status: Execution finished without errors
16#8200	ERR_WRONG_WINDOW_SIZE Error: Incorrect window size/width set. Set a value between 1 and 100.

## Functional description

**Note** The block LGF\_FloatingAverage does not query the data type for the input parameter `value`. For data types other than REAL, either an implicit conversion is performed automatically or an error is generated during compilation.

You can find further information in the Chapter “Overview of Data Type Conversion” in the Online Help section of the TIA Portal or under:

<https://support.industry.siemens.com/cs/ww/en/view/109773506/100611494667>

The block calculates the (moving) mean value based on the set window width. The window width indicates the maximum number of values read in last. After the maximum number of values has been read, the output `windowSizeReached` is set and each newly read value replaces the oldest value (FIFO principle).

Two options are available for reading the values. With the input `cyclicExecution`, the values are read and calculated cyclically. With the `trigger` input, the values are read in and calculated with each pulse.

## Change log

Version & Date	Change description
<b>01.00.00</b> 16.06.2016	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.01.00</b> 21.02.2017	<b>Siemens Industry Online Support</b> Adding variable window size for calculation Optimizing calculation algorithm
<b>01.01.01</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.01.02</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.01.03</b> 07.11.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 16.06.2020	<b>Simatic Systems Support</b> refactor and simplify code
<b>03.00.02</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.11 LGF\_Histogram\_DInt (FB / V3.0.1)

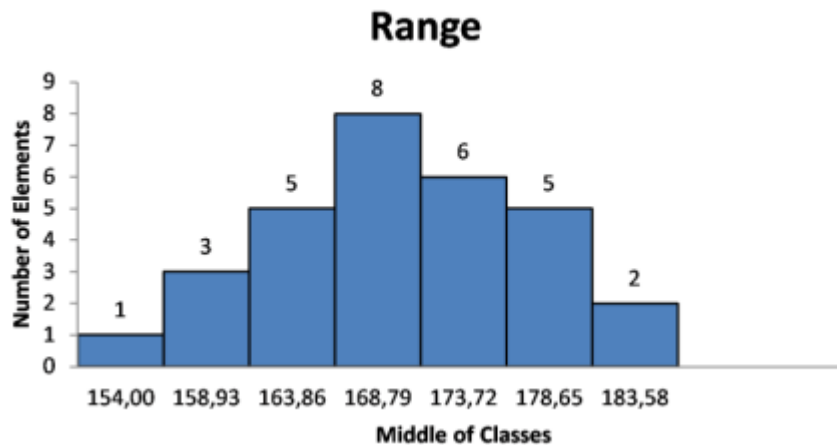
Author: Siemens Digital Industry

#### Short description

The histogram shows the frequency distribution of a sample by class. A class describes a value interval in which the individual frequencies are added together. After specifying the number of classes, the class width and the respective class center are calculated. The number of classes is limited to 15.

The distribution is represented as a rectangle around the class mean with the class width and the cumulated frequency as height.

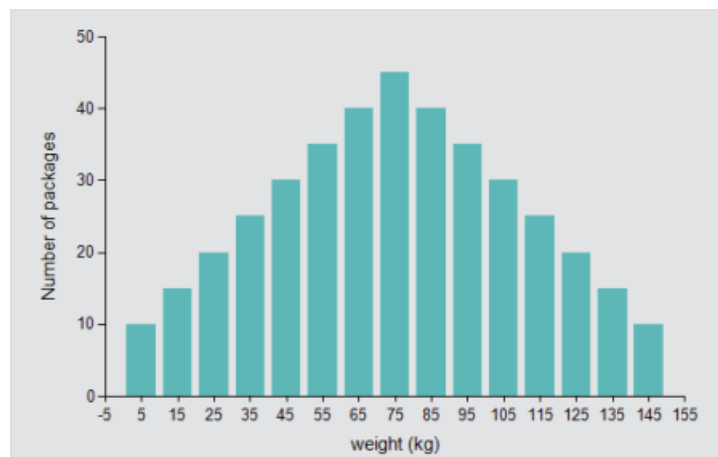
Figure: Distribution



#### WinCC-Control

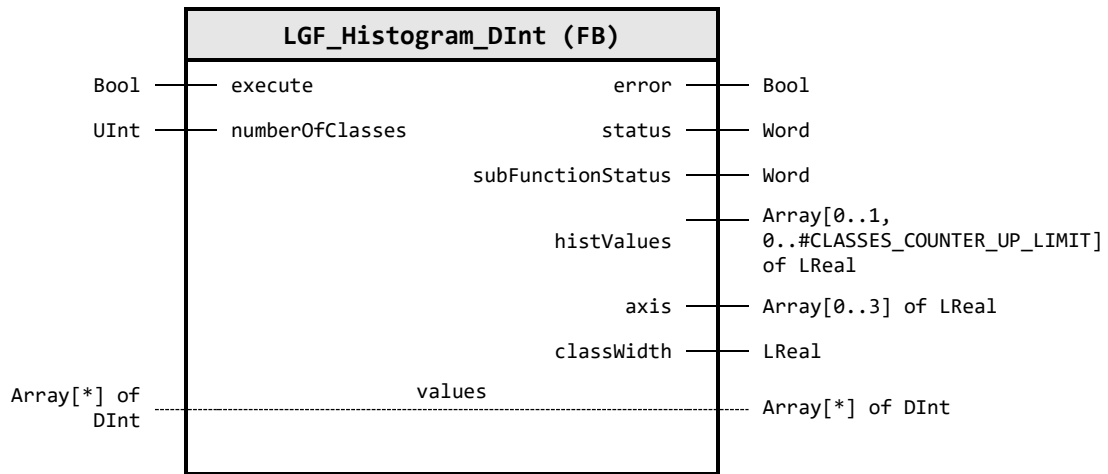
To visualize the Boxplot, the Siemens Industry Online Support offers you a Net-Control, which you can use in conjunction with WinCC Runtime Professional. You can find the controls in the `UserFiles` folder of this library.

Figure: .Net Control "Histogram"





**Block Interface**



**Input parameter**

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Activation of the calculation with each positive edge.
numberOfClasses	UInt	0	Number of desired classes.

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subFunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
histValues	Array[0..1, 0..#CLASSES_COUNTER_UP_LIMIT] of LReal	Outputs the calculated values in a two-dimensional array. • `histValues[0,0..14]` displays the relative frequency of the individual classes. • `histValues[1,0..14]` displays the class centers. • If fewer than 15 classes are desired, the array elements that are not required are output with 0.
axis	Array[0..3] of LReal	Specifies the axis values: • Lower X axis value • Upper X axis value • Lower Y axis value • Upper Y axis value
classWidth	LReal	Returns the calculated class width.

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of DInt	The array containing the data series that is to be used for the calculation

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#8600	ERR_SHELL_SORT Error: Error in command `LGF_ShellSort_DInt`. Check `subFunctionStatus` code
16#9101	ERR_WRONG_NO_CLASSES Error: Incorrect number of classes. Give the parameter `numberOfClasses` a valid value (1 to 15).

**Functional description**

The block sorts the transferred data and calculates the general class width using the transferred class count and data range. The block then counts the values that lie within a class. In order to draw a histogram, the block also calculates the necessary X and Y coordinates.

The elements of the passed array `values` are sorted in ascending order by the block. The `LGF_ShellSort_UDInt` block is used for sorting.

The number of classes can be specified using the following rule of thumb:

$$\text{Number of classes} = \sqrt{\text{number of elements}}$$

e.g. 100 values →  $\text{Number of classes} = \sqrt{100} = 10$

**Formulas**

The block uses the following formula to calculate the class width:

$$\text{classWidth} = \frac{\text{max} - \text{min}}{\text{Number of classes}}$$

**Change log**

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>02.00.00</b> 06.11.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.12 LGF\_Histogram\_LReal (FB / V3.0.1)

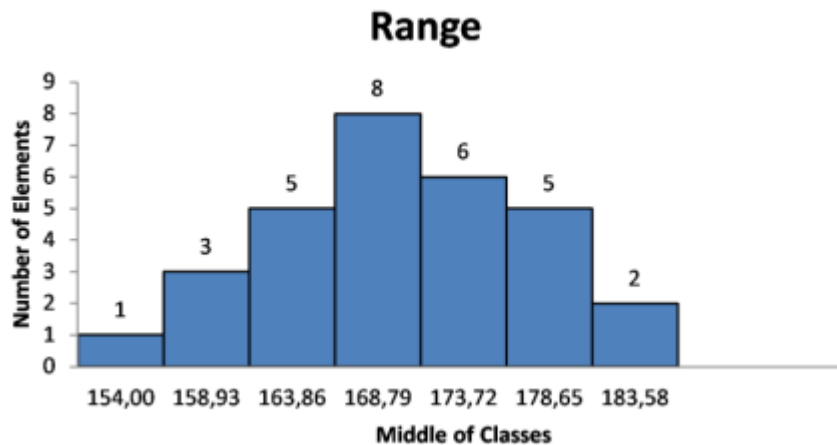
Author: Siemens Digital Industry

#### Short description

The histogram shows the frequency distribution of a sample by class. A class describes a value interval in which the individual frequencies are added together. After specifying the number of classes, the class width and the respective class center are calculated. The number of classes is limited to 15.

The distribution is represented as a rectangle around the class mean with the class width and the cumulated frequency as height.

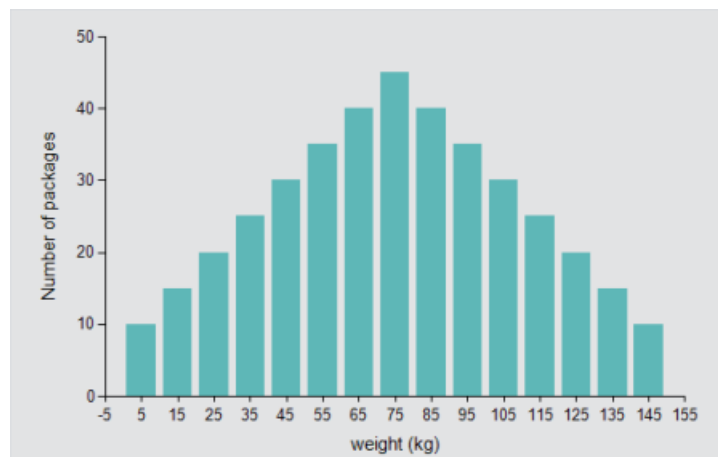
Figure: Distribution



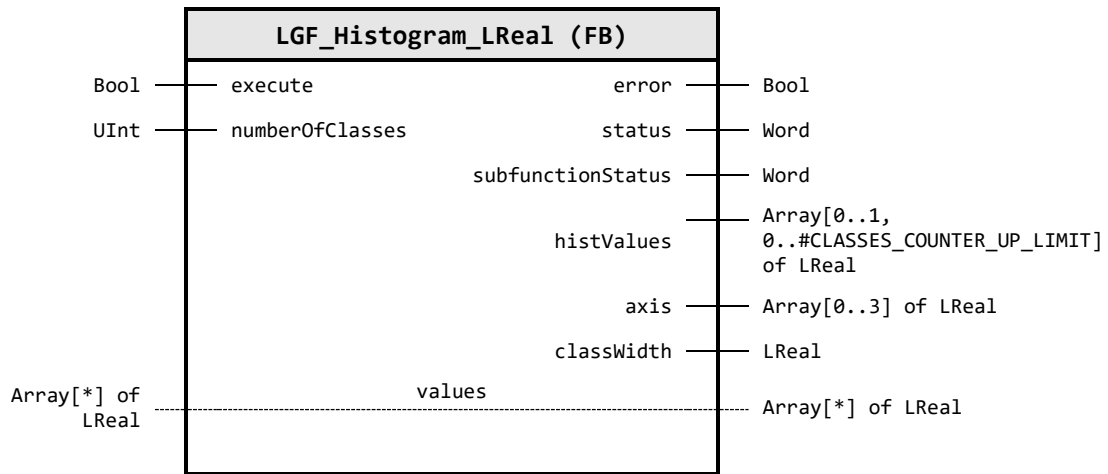
#### WinCC-Control

To visualize the Boxplot, the Siemens Industry Online Support offers you a Net-Control, which you can use in conjunction with WinCC Runtime Professional. You can find the controls in the `UserFiles` folder of this library.

Figure: .Net Control "Histogram"



**Block Interface**



**Input parameter**

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Activation of the calculation with each positive edge.
numberOfClasses	UInt	0	Number of desired classes.

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
histValues	Array[0..1, 0..#CLASSES_COUNTER_UP_LIMIT] of LReal	Outputs the calculated values in a two-dimensional array. • `histValues[0,0..14]` displays the relative frequency of the individual classes. • `histValues[1,0..14]` displays the class centers. • If fewer than 15 classes are desired, the array elements that are not required are output with 0.
axis	Array[0..3] of LReal	Specifies the axis values: • Lower X axis value • Upper X axis value • Lower Y axis value • Upper Y axis value
classWidth	LReal	Returns the calculated class width.

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of LReal	The array containing the data series that is to be used for the calculation

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#8600	ERR_SHELL_SORT Error: Error in command `LGF_ShellSort_LReal`. Check `subFunctionStatus` code
16#9101	ERR_WRONG_NO_CLASSES Error: Incorrect number of classes. Give the parameter `numberOfClasses` a valid value (1 to 15).

**Functional description**

The block sorts the transferred data and calculates the general class width using the transferred class count and data range. The block then counts the values that lie within a class. In order to draw a histogram, the block also calculates the necessary X and Y coordinates.

The elements of the passed array `values` are sorted in ascending order by the block. The `LGF_ShellSort_UDInt` block is used for sorting.

The number of classes can be specified using the following rule of thumb:

$$\text{Number of classes} = \sqrt{\text{number of elements}}$$

e.g. 100 values →  $\text{Number of classes} = \sqrt{100} = 10$

**Formulas**

The block uses the following formula to calculate the class width:

$$\text{classWidth} = \frac{\text{max} - \text{min}}{\text{Number of classes}}$$

**Change log**

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>02.00.00</b> 06.11.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.13 LGF\_Histogram\_UDInt (FB / V3.0.1)

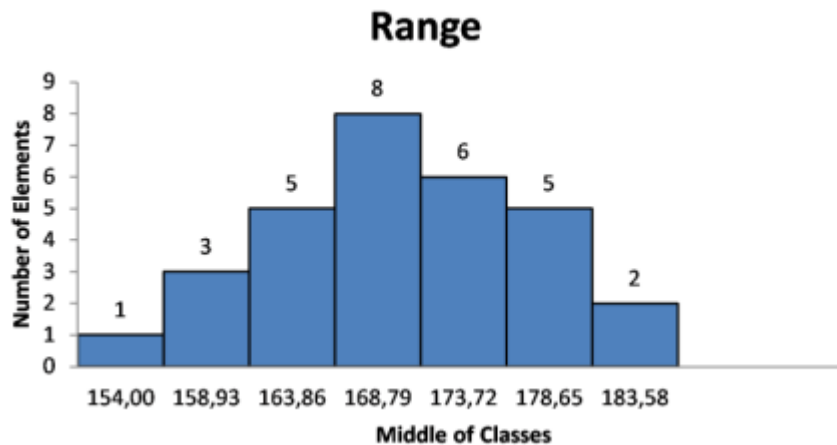
Author: Siemens Digital Industry

#### Short description

The histogram shows the frequency distribution of a sample by class. A class describes a value interval in which the individual frequencies are added together. After specifying the number of classes, the class width and the respective class center are calculated. The number of classes is limited to 15.

The distribution is represented as a rectangle around the class mean with the class width and the cumulated frequency as height.

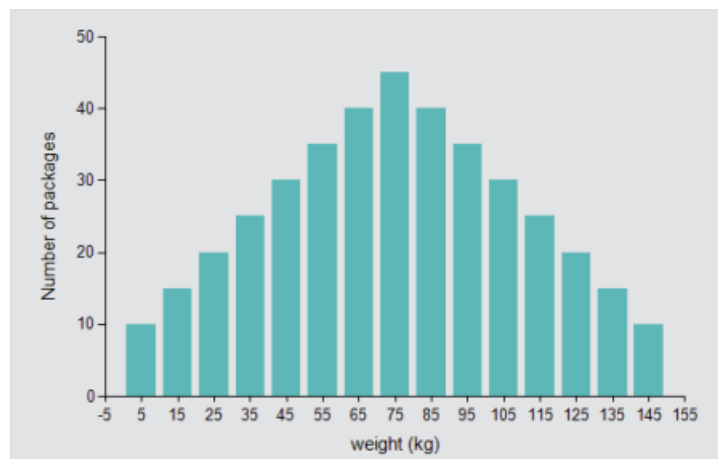
Figure: Distribution



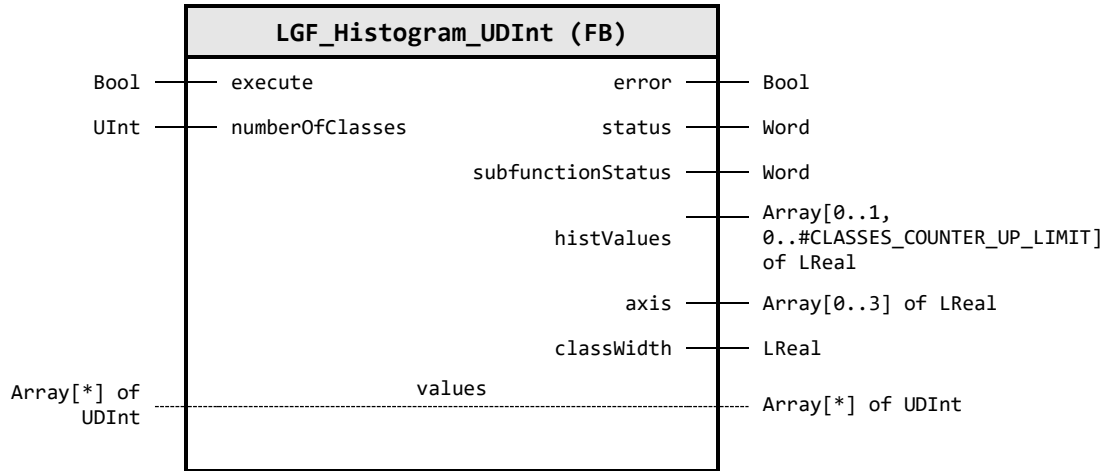
#### WinCC-Control

To visualize the Boxplot, the Siemens Industry Online Support offers you a Net-Control, which you can use in conjunction with WinCC Runtime Professional. You can find the controls in the `UserFiles` folder of this library.

Figure: .Net Control "Histogram"



**Block Interface**



**Input parameter**

Identifier	Data type	Default value	Description
execute	Bool	FALSE	Activation of the calculation with each positive edge.
numberOfClasses	UInt	0	Number of desired classes.

**Output parameter**

Identifier	Data type	Description
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)
subfunctionStatus	Word	Status or return value of called FB's, FC's and system blocks
histValues	Array[0..1, 0..#CLASSES_COUNTER_UP_LIMIT] of LReal	Outputs the calculated values in a two-dimensional array. • `histValues[0,0..14]` displays the relative frequency of the individual classes. • `histValues[1,0..14]` displays the class centers. • If fewer than 15 classes are desired, the array elements that are not required are output with 0.
axis	Array[0..3] of LReal	Specifies the axis values: • Lower X axis value • Upper X axis value • Lower Y axis value • Upper Y axis value
classWidth	LReal	Returns the calculated class width.

**In/Out parameter**

Identifier	Data type	Description
values	Array[*] of UDInt	The array containing the data series that is to be used for the calculation

**Status & Error codes**

Code / Value	Identifier / Description
16#0000	STATUS_EXECUTION_FINISHED Status: Execution finished without errors
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#8600	ERR_SHELL_SORT Error: Error in command `LGF_ShellSort_UDInt`. Check `subFunctionStatus` code
16#9101	ERR_WRONG_NO_CLASSES Error: Incorrect number of classes. Give the parameter `numberOfClasses` a valid value (1 to 15).

**Functional description**

The block sorts the transferred data and calculates the general class width using the transferred class count and data range. The block then counts the values that lie within a class. In order to draw a histogram, the block also calculates the necessary X and Y coordinates.

The elements of the passed array `values` are sorted in ascending order by the block. The `LGF_ShellSort_UDInt` block is used for sorting.

The number of classes can be specified using the following rule of thumb:

$$\text{Number of classes} = \sqrt{\text{number of elements}}$$

e.g. 100 values →  $\text{Number of classes} = \sqrt{100} = 10$

**Formulas**

The block uses the following formula to calculate the class width:

$$\text{classWidth} = \frac{\text{max} - \text{min}}{\text{Number of classes}}$$

**Change log**

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>02.00.00</b> 06.11.2019	<b>Simatic Systems Support</b> Code refactoring, comments added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation



### 4.13.14 LGF\_SimpleSmoothingFB (FB / V3.0.1)

Author: Siemens Digital Industry

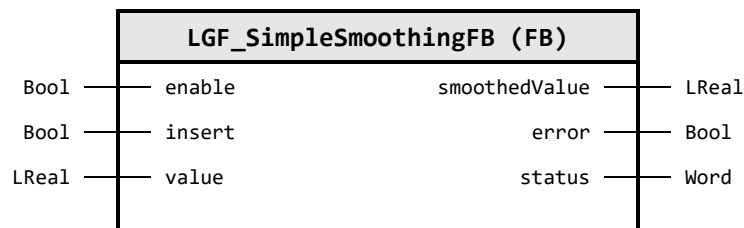
#### Short description

The function calculates the linear mean value cyclically.

The simplest form of smoothing a sequence of measured values is to calculate the linear mean value by three points.

The function reads-in a value with each positive edge on the `insert` input. As soon as three values have been read in, the block calculates a smoothed value and outputs it.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
enable	Bool	FALSE	Activates the block. As long as enable is `TRUE`, the block can accept values on the parameter `value`.
insert	Bool	FALSE	Accepts the value at the input `value` at positive edge and outputs a `smoothedValue` if three values have been read in.
value	LReal	0.0	Value that is to be included in the smoothing.

#### Output parameter

Identifier	Data type	Description
smoothedValue	LReal	The smoothed value.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling
16#7002	STATUS_SUBSEQUENT_CALL Status: Processing is active. Subsequent call of FB.

## 4 Program blocks

Code / Value	Identifier / Description
16#7010	<b>STATUS_NOT_ENOUGH_VALUES</b> Status: Not enough values. The block requires three (3) values to calculate a smoothed value. Transfer additional values with a positive edge on the `insert` input.

### Functional description

The function calculates the smoothed values using the following formula:

$$\overline{y(n)} = \frac{y(n-1) + y(n) + y(n+1)}{3}$$

The calculated value is output or the calculated values are output at output `smoothedValue`.

Based on this formula, the function cannot calculate values for the elements 0 and N.

### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.03</b> 05.11.2019	<b>Simatic Systems Support</b> Regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

### 4.13.15 LGF\_SmoothByPolynomFB (FB / V3.0.1)

Author: Siemens Digital Industry

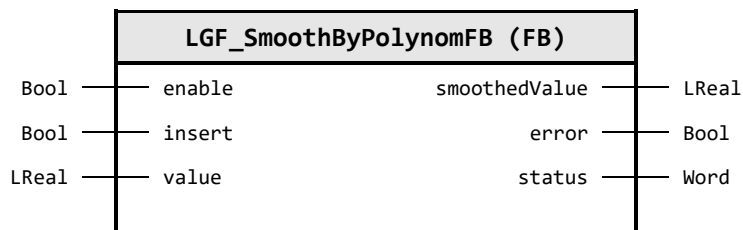
#### Short description

This function calculates the smoothed values by polynomial cyclically.

For smoothing, a 3rd degree polynomial is placed through five value points. The error squares of the distances between polynomial and real value are minimized. The smoothed values can be determined from the polynomial parameters obtained in this way.

The function reads-in a value with each positive edge on the `insert` input. As soon as five values have been read in, the block calculates a smoothed value and outputs it.

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
enable	Bool	FALSE	Activates the block. As long as enable is `TRUE`, the block can accept values on the parameter `value`.
insert	Bool	FALSE	Accepts the value at the input `value` at positive edge and outputs a `smoothedValue` if five values have been read in.
value	LReal	0.0	Value that is to be included in the smoothing.

#### Output parameter

Identifier	Data type	Description
smoothedValue	LReal	The smoothed value.
error	Bool	FALSE: No error TRUE: An error occurred during the execution of the FB
status	Word	16#0000-16#7FFF: Status of the FB 16#8000-16#FFFF: Error identification (see following Table)

#### Status & Error codes

Code / Value	Identifier / Description
16#7000	STATUS_NO_CALL Status: No call of FB. The block waits for activation through the parameter `enable`.
16#7001	STATUS_FIRST_CALL Status: First call of FB after enabling

Code / Value	Identifier / Description
16#7002	STATUS_SUBSEQUENT_CALL Status: Processing is active. Subsequent call of FB.
16#7010	STATUS_NOT_ENOUGH_VALUES Status: Not enough values. The block requires five (5) values to calculate a smoothed value. Transfer additional values with a positive edge on the `insert` input.

### Functional description

The 3rd degree compensation polynomial is calculated as follows:

$$\overline{y(n)} = \frac{1}{35} \cdot (-3 \cdot y(n-2) + 12 \cdot y(n-1) + 17 \cdot y(n) + 12 \cdot y(n+1) - 3 \cdot y(n+2))$$

$N - 4$  smoothed measured values can thus be calculated from the  $N$  measured values. The output array contains the value 0 in the index (0.1,  $N-1$ ,  $N$ ).

These "missing" values are calculated with the following formalisms:

$$\overline{y(n-2)} = \frac{1}{70} \cdot (69 \cdot y(n-2) + 4 \cdot y(n-1) - 6 \cdot y(n) + 4 \cdot y(n+1) - y(n+2))$$

$$\overline{y(n-1)} = \frac{2}{70} \cdot (2 \cdot y(n-2) + 27 \cdot y(n-1) + 12 \cdot y(n) - 8 \cdot y(n+1) + 2 \cdot y(n+2))$$

$$\overline{y(n+1)} = \frac{2}{70} \cdot (2 \cdot y(n-2) - 8 \cdot y(n-1) + 12 \cdot y(n) + 27 \cdot y(n+1) + 2 \cdot y(n+2))$$

$$\overline{y(n+2)} = \frac{1}{70} \cdot (-y(n-2) + 4 \cdot y(n-1) - 6 \cdot y(n) + 4 \cdot y(n+1) + 69 \cdot y(n+2))$$

### Change log

Version & Date	Change description
<b>01.00.00</b> 23.11.2018	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 05.11.2019	<b>Simatic Systems Support</b> Bugfixes, regions, comments and constants are added
<b>03.00.00</b> 23.04.2020	<b>Simatic Systems Support</b> Set version to V3.0.0, harmonize the version of the whole library
<b>03.00.01</b> 06.04.2021	<b>Simatic Systems Support</b> Insert documentation

## 4.14 Legacy / Counter operations

### 4.14.1 LGF\_CountFallInDWord (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

The function analyzes a variable of the type DWord and outputs how often a 1-0 sequence (falling edge) occurs in the variable.

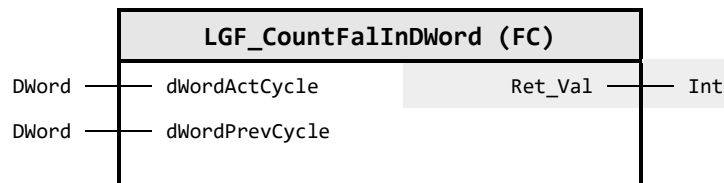
#### Note

#### LEGACY FUNCTION

Please update and use the FB with the same name LGF\_CountFalInDWord in the future!

This function is no longer maintained!

#### Block Interface



#### Input parameter

Identifier	Data type	Description
dWordActCycle	DWord	Input Double word in which the falling edges are counted
dWordPrevCycle	DWord	Double word from the previous cycle

#### Output parameter

Identifier	Data type	Description
Ret_Val	Int	Number of falling edges in the DWord

#### Functional description

In a variable of the data type DWORD, the block counts the falling edges (1-0 transitions) from left to right. The output `countFalInDWord` outputs the number of falling edges.

So that falling edges at the variable limit are also detected, the input `value` is copied to the static variable `statDWordPrevCycle` at the end of the evaluation and evaluated in the next cycle.

#### Example

The following example illustrates the block's functionality. In this case, it is assumed that a signal of unknown length is continuously sampled in the form of double words (DWORD) per cycle.

Within this signal, the 1-0 sequences (falling edges) must be counted and output continuously. To detect the falling edge on variable limits, as in this example, the input "statDWordPrevCycle" must be interconnected with the double word of the previous sampling.

Table: Example

#### 4 Program blocks

---

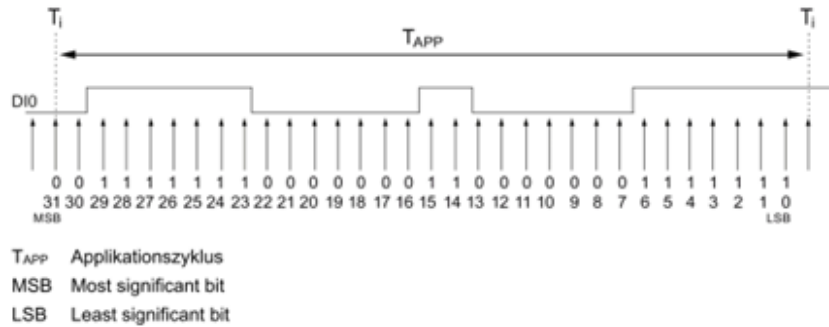
DWord previous cycle statDWordPrevCycle	DWord actual cycle value
1001_0000_0001_1010_1001_0000_0001_1011	0010_1010_0001_1111_0100_0011_1000_0101

Number of 1-0 sequences (falling edges): Ret\_Val= 8

**Application example**

Excerpt from the manual of the technology module TM Timer DIDQ 16x24V:  
 With the oversampling function, the technology module records the state of the respective digital input per application cycle (e.g. OB61) at 32 points in time with a uniform time interval. The 32 states are jointly returned as 32-bit values in the checkback interface.

Figure: Example of an oversampling of DI0 on TM Timer DIDQ 16x24V



The LGF\_CountFallnDWord block is used, in this case, to count how often a falling edge occurs.

SIMATIC ET 200MP/S7-1500 Technology Module TM Timer DIDQ 16x24V (6ES7552-1AA00-0AB0)

<https://support.industry.siemens.com/cs/ww/en/view/95153313>

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation and LEGACY Hint Please use the FB with the same name `LGF_CountFallnDWord` in the future

### 4.14.2 LGF\_CountRisInDWord (FC / V3.0.1)

Author: Siemens Digital Industry Support

#### Short description

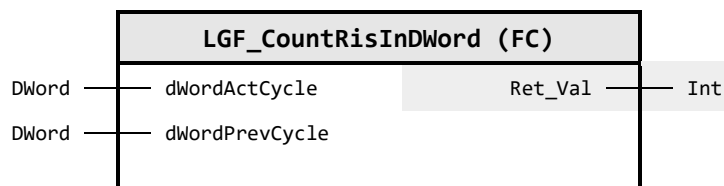
The function analyzes a variable of the type DWORD and outputs how often a 0-1 sequence (rising edge) occurs in the variable.

#### Note

#### LEGACY FUNCTION

Please update and use the FB with the same name LGF\_CountRisInDWord in the future!  
This function is no longer maintained!

#### Block Interface



#### Input parameter

Identifier	Data type	Description
dWordActCycle	DWord	Input Double word in which the rising edges are counted
dWordPrevCycle	DWord	Double word from the previous cycle

#### Output parameter

Identifier	Data type	Description
Ret_Val	Int	Number of rising edges in the DWord

#### Functional description

In a variable of the data type DWORD, the block counts the rising edges (0-1 transitions) from left to right. The output `countRisInDWord` outputs the number of rising edges.

So that rising edges at the variable limit are also detected, the input `value` is copied to the static variable `statDWordPrevCycle` at the end of the evaluation and evaluated in the next cycle.

#### Example

The following example illustrates the block's functionality. In this case, it is assumed that a signal of unknown length is continuously sampled in the form of double words (DWORD) per cycle.

Within this signal, the 0-1 sequences (rising edges) must be counted and output continuously. To detect the rising edge on variable limits, as in this example, the input "statDWordPrevCycle" must be interconnected with the double word of the previous sampling.

Table: Example



#### 4 Program blocks

---

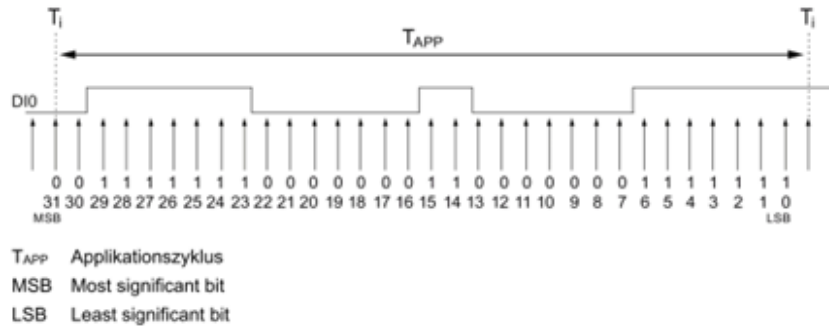
DWord previous cycle statDWordPrevCycle	DWord actual cycle value
1001_0000_0001_1010_1001_0000_0001_1010	1010_1010_0001_1111_0100_0011_1000_0101

Number of 0-1 sequences (rising edges): Ret\_Val = 9

**Application example:**

Excerpt from the manual of the technology module TM Timer DIDQ 16x24V:  
 With the oversampling function, the technology module records the state of the respective digital input per application cycle (e.g. OB61) at 32 points in time with a uniform time interval. The 32 states are jointly returned as 32-bit values in the checkback interface.

Figure: Example of an oversampling of DI0 on TM Timer DIDQ 16x24V



The block LGF\_CountRisInDWordFB is used in this case to count how often a rising edge occurs.

SIMATIC ET 200MP/S7-1500 Technology Module TM Timer DIDQ 16x24V (6ES7552-1AA00-0AB0)

<https://support.industry.siemens.com/cs/ww/en/view/95153313>

© Siemens AG 2021. All rights reserved

**Change log**

Version & Date	Change description
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA V14 Update 1
<b>01.00.02</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.03</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>03.00.01</b> 12.11.2020	<b>Simatic Systems Support</b> Insert documentation and LEGACY Hint Please use the FB with the same name `LGF_CountRisInDWord` in the future

## 4.15 Legacy / Signal generators

### 4.15.1 LGF\_SawTooth (FB / V3.0.1)

Author: Siemens Digital Industries

#### Short description

This function generates a sawtooth-shaped signal profile. Each sawtooth consists of a defined number of steps (increments).

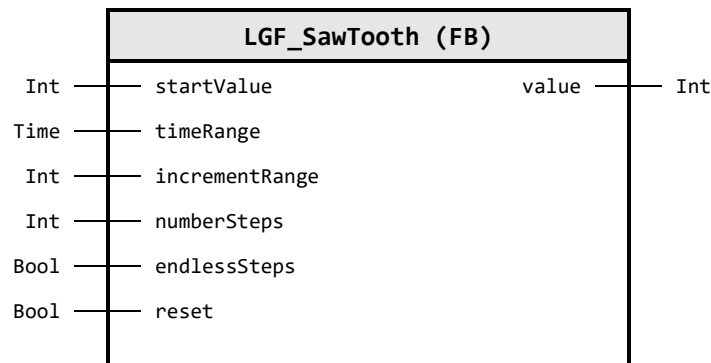
#### Note

#### LEGACY FUNCTION

Please update and use the FB with the same name LGF\_CountRisInDWord in the future!

This function is no longer maintained!

#### Block Interface



#### Input parameter

Identifier	Data type	Default value	Description
startValue	Int	0	Start value at which the signal begins.
timeRange	Time	T#0s	Time after which the output parameter `value` is incremented
incrementRange	Int	0	Size of the jump from one increment to the next.
numberSteps	Int	0	Number of increments per sawtooth. (In the case of an endless sawtooth signal, this information is not necessary).
endlessSteps	Bool	FALSE	Specifies whether an endless sawtooth signal will be generated. `TRUE` - Activated, `FALSE` - Disabled
reset	Bool	FALSE	Sawtooth starts again at the start value, `startValue`.

#### Output parameter

Identifier	Data type	Description
value	Int	Current value of the sawtooth signal.

## Functional description

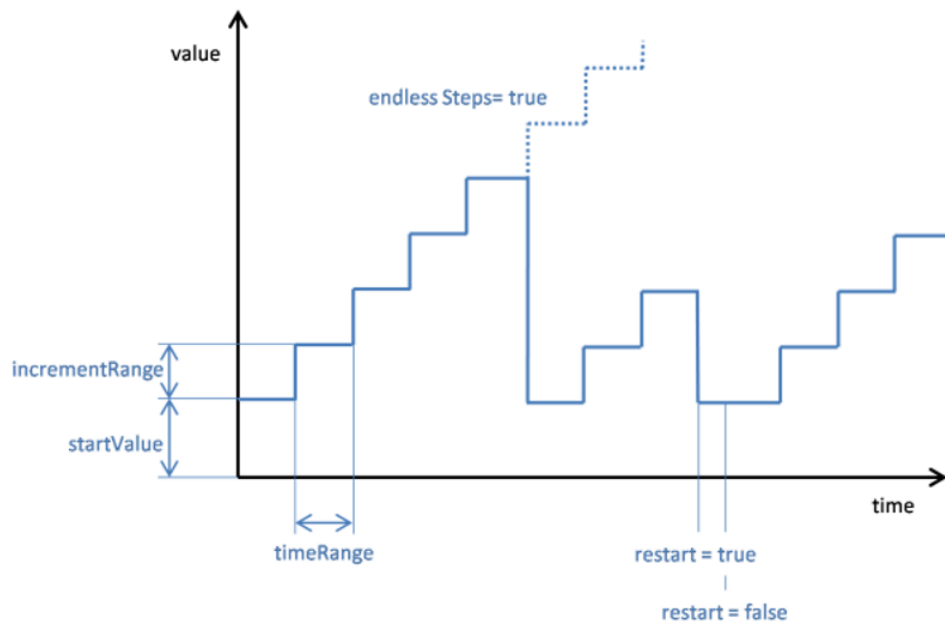
**Note** Please note that changes at the input parameters only become effective with reset.

The block calculates the values for a sawtooth-shaped signal profile, which is output to the output parameter `value`. The signal begins with the start value `startValue` and is added with the value `increment` after each elapse of the time interval `timeRange`. The value can also be negative.

If the variable `endlessSteps` is set to `FALSE`, the number of add operations is counted. If this exceeds the value `numberSteps`, the output parameter `value` is set back to the start value. A new sawtooth begins.

If the variable `endlessSteps` is set to `TRUE`, the value `increment` is added without interruption, starting once at `startValue`. If the maximum positive INT value range (32767) of the output parameter `value` is exceeded, `value` changes to the maximum negative INT value range (-32768) and will continue to be added up.

**Note** The duration of a sawtooth at `endlessSteps` on `FALSE` is calculated as follows:  
 $\text{Duration} = \#timeRange * (\#numberSteps + 1)$



**Change log**

<b>Version &amp; Date</b>	<b>Change description</b>
<b>01.00.00</b> 19.08.2015	<b>Siemens Industry Online Support</b> First released version
<b>01.00.01</b> 02.11.2015	<b>Siemens Industry Online Support</b> Bug fix
<b>01.00.02</b> 02.01.2017	<b>Siemens Industry Online Support</b> Upgrade: TIA Portal V14 Update 1
<b>01.00.03</b> 17.08.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15 Update 2
<b>01.00.04</b> 23.11.2018	<b>Siemens Industry Online Support</b> Upgrade: TIA V15.1
<b>01.00.10</b> 23.09.2019	<b>Simatic Systems Support</b> Code refactoring, regions and more comments added
<b>03.00.01</b> 15.02.2021	<b>Simatic Systems Support</b> Insert documentation

## 5 PLC data types

### 5.1 Date and timer operations

#### 5.1.1 LGF\_typeGPS (UDT)

##### Description

Datatype for GPS Coordinates Latitude and Longitude.  
Child Datatypes in Degree, Minutes, Seconds and the Direction.  
Datatype for a whole GPS Data set.

##### Parameter description

Identifier	Data type	Default value	Description
latitude	LGF_typeGPS_DM S	---	Latitude child element
longitude	LGF_typeGPS_DM S	---	Longitude child element

#### 5.1.2 LGF\_typeGPS\_DD (UDT)

##### Description

Datatype for GPS Coordinates in decimal degrees.  
For latitude and longitude.  
Datatype for a whole GPS Data set.

##### Parameter description

Identifier	Data type	Default value	Description
latitude	Real	0.0	Degrees latitude with decimal places (Unit: degree decimal), North = positive; South = negative) valid value range [-90.00000..90.00000]
longitude	Real	0.0	Degrees longitude in degrees with decimal places (Unit: degree decimal), East = positive; West = negative) valid range [-180.00000..180.00000]

#### 5.1.3 LGF\_typeGPS\_DMS (UDT)

##### Description

Datatype for GPS Coordinates in Degree, Minutes, Seconds and the Direction.  
Can be used for latitude and as well for longitude.  
The Datatype is already used e.g. in LGF\_typeGPS.

##### Parameter description

Identifier	Data type	Default value	Description
dir	Char	"	Direction [N, S, E, W, n, s, e, w]
deg	UInt	0	Degrees, Latitude [-89..+ 89]; Longitude [-179..+179]
min	UInt	0	Minutes [0..+59]
sec	UInt	0	Seconds [0..+59]

## 5.2 Technology operations

### 5.2.1 LGF\_typeNonLinSetpoints (UDT)

#### Description

Data type to setup a setpoint table for the function `LGF_NonLinearInterpolation`

#### Parameter description

Identifier	Data type	Default value	Description
inputValue	LReal	0.0	Input value to be interpolated
outputValue	LReal	0.0	Corresponding interpolated value

### 5.2.2 LGF\_typeRampTimeTable (UDT)

#### Description

Data type to setup a speed curve based on a setpoint table for the function `LGF_RampCI`

#### Parameter description

Identifier	Data type	Default value	Description
outputValue	LReal	0.0	Setpoint Value to reach by the interpolation curve
time	Time	T#0s	Time until the interpolation point is reached

## 5.3 Measurement operations

### 5.3.1 LGF\_typeRegressionLine (UDT)

#### Description

The data type is for transferring datapoints (Key- Value pairs) to `LGF_RegressionLine` and calculate the interpolated linear equation parameters slope and intercept.

#### Parameter description

Identifier	Data type	Default value	Description
x	Real	0.0	X-Axis value
y	Real	0.0	Y-Axis value



## 6 PLC tags & constants

### 6.1.1 LGF\_CONSTANTS

Constant identifier, values and description

Identifier & Value	Description
<b>LGF_BYTE_MAX</b> 16#FF	Byte - maximum value of byte - unsigned 8 bit
<b>LGF_BYTE_MIN</b> 16#00	Byte - minimum value of byte - unsigned 8 bit
<b>LGF_CHAR_BLANK</b> ' '	Char ' ' - used as empty fill character
<b>LGF_CHAR_FILL_ZERO</b> '0'	Char '0' - used as default fill character
<b>LGF_DEBUG</b> FALSE	Debug const - used for DEBUG Purpose to activate DEBUG code, normally debug Code is NOT included in productive code
<b>LGF_DINT_MAX</b> DINT#+2147483647	DInt - maximum value of double integer - signed 32 bit
<b>LGF_DINT_MIN</b> DINT#-2147483648	DInt - minimum value of double integer - signed 32 bit
<b>LGF_DWORD_MAX</b> 16#FFFFFFFF	DWord - maximum value of double word - unsigned 32 bit
<b>LGF_DWORD_MIN</b> 16#00000000	DWord - minimum value of double word - unsigned 32 bit
<b>LGF_INT_MAX</b> INT#+32767	Int - maximum value of integer - signed 16 bit
<b>LGF_INT_MIN</b> INT#-32768	Int - minimum value of integer - signed 16 bit
<b>LGF_LREAL_INFINITY_NEG</b> 16#FFF0000000000000	LReal / Double - negative Infinity
<b>LGF_LREAL_INFINITY_POS</b> 16#7FF0000000000000	LReal / Double - positive Infinity
<b>LGF_LREAL_MAX_NORM</b> 16#7FEFFFFFFFFFFFFFFF	LReal / Double - maximum value in normalized format
<b>LGF_LREAL_MAX_SUBNORM</b> 16#000FFFFFFFFFFFFFFF	LReal / Double - maximum value in denormalized format
<b>LGF_LREAL_MIN_NORM</b> 16#0010000000000000	LReal / Double - minimum value in normalized format
<b>LGF_LREAL_MIN_SUBNORM</b> 16#0000000000000001	LReal / Double - minimum value in denormalized format
<b>LGF_LREAL_NAN</b> 16#7FF8000000000000	LReal / Double - NAN - Not A Number
<b>LGF_REAL_INFINITY_NEG</b> 16#FF800000	Real / Float - negative Infinity
<b>LGF_REAL_INFINITY_POS</b> 16#7F800000	Real / Float - positive Infinity
<b>LGF_REAL_MAX_NORM</b> 16#7F7FFFFF	Real / Float - maximum value in normalized format
<b>LGF_REAL_MAX_SUBNORM</b> 16#007FFFFF	Real / Float - minimum value in denormalized format

## 6 PLC tags & constants

Identifier & Value	Description
<b>LGF_REAL_MIN_NORM</b> 16#00800000	Real / Float - minimum value in normalized format
<b>LGF_REAL_MIN_SUBNORM</b> 16#00000001	Real / Float - minimum value in denormalized format
<b>LGF_REAL_NAN</b> 16#7FC00000	Real / Float - NAN - Not A Number
<b>LGF_SINT_MAX</b> SINT#+127	SInt - maximum value of short integer - signed 8 bit
<b>LGF_SINT_MIN</b> SINT#-128	SInt - minimum value of short integer - signed 8 bit
<b>LGF_STRING_LENGTH_MAX</b> 254	String - maximum length of String
<b>LGF_UDINT_MAX</b> UDINT#+4294967295	UDInt - maximum value of double integer - unsigned 32 bit
<b>LGF_UDINT_MIN</b> UDINT#0	UDInt - minimum value of double integer - unsigned 32 bit
<b>LGF_UINT_MAX</b> UINT#+65535	UInt - maximum value of integer - unsigned 16 bit
<b>LGF_UINT_MIN</b> UINT#0	UInt - minimum value of integer - unsigned 16 bit
<b>LGF_USINT_MAX</b> USINT#+255	USInt - maximum value of short integer - unsigned 8 bit
<b>LGF_USINT_MIN</b> USINT#0	USInt - minimum value of short integer - unsigned 8 bit
<b>LGF_WORD_MAX</b> 16#FFFF	Word - maximum value of word - unsigned 16 bit
<b>LGF_WORD_MIN</b> 16#0000	Word - minimum value of word - unsigned 16 bit
<b>LGF_WSTRING_LENGTH_MAX</b> 16382	WString - maximum length of WString

**6.1.2 LGF\_CONSTANTS\_UNITS**

Constant identifier, values and description

Identifier & Value	Description
<b>LGF_UNIT_ATTO</b> 0.00000000000000000000000001	Atto / a / 10 <sup>-18</sup>
<b>LGF_UNIT_CENTI</b> 0.01	Centi / c / 10 <sup>-2</sup>
<b>LGF_UNIT_DECA</b> 10.0	Deca / da / 10 <sup>1</sup>
<b>LGF_UNIT_DECI</b> 0.1	Deci / d / 10 <sup>-1</sup>
<b>LGF_UNIT_EXA</b> 10000000000000000000.0	Exa / E / 10 <sup>18</sup>
<b>LGF_UNIT_FEMTO</b> 0.000000000000000000000001	Femto / f / 10 <sup>-15</sup>
<b>LGF_UNIT_GIGA</b> 1000000000.0	Giga / G / 10 <sup>9</sup>
<b>LGF_UNIT_HECTO</b> 100.0	Hecto / h / 10 <sup>2</sup>
<b>LGF_UNIT_KILO</b> 1000.0	Kilo / k / 10 <sup>3</sup>
<b>LGF_UNIT_MEGA</b> 1000000.0	Mega / M / 10 <sup>6</sup>
<b>LGF_UNIT_MICRO</b> 0.000001	Micro / $\mu$ / 10 <sup>-6</sup>
<b>LGF_UNIT_MILLI</b> 0.001	Milli / m / 10 <sup>-3</sup>
<b>LGF_UNIT_NANO</b> 0.000000001	Nano / n / 10 <sup>-9</sup>
<b>LGF_UNIT_PERCENT</b> 0.01	Percent / % / present a fraction from hundred
<b>LGF_UNIT_PERMILL</b> 0.001	Per mill / ‰ / present a fraction from thousand
<b>LGF_UNIT_PETA</b> 10000000000000000.0	Peta / P / 10 <sup>15</sup>
<b>LGF_UNIT_PICO</b> 0.0000000000000001	Pico / p / 10 <sup>-12</sup>
<b>LGF_UNIT_TERA</b> 1000000000000.0	Tera / T / 10 <sup>12</sup>
<b>LGF_UNIT_YOCTO</b> 0.000000000000000000000000000001	Yocto / y / 10 <sup>-24</sup>
<b>LGF_UNIT_YOTTA</b> 10000000000000000000000000.0	Yotta / Y / 10 <sup>24</sup>
<b>LGF_UNIT_ZEPTO</b> 0.000000000000000000000000000001	Zepto / z / 10 <sup>-21</sup>
<b>LGF_UNIT_ZETTA</b> 1000000000000000000000000.0	Zetta / Z / 10 <sup>21</sup>

**6.1.3 LMATH\_CONSTANTS**

Constant identifier, values and description

Identifier & Value	Description
<b>LMATH_2_PI</b> 6.28318530717958647692528676655 900	Two Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_3_PI</b> 9.4247779607693797	Three Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_4_PI</b> 12.5663706143591729538505735331 180	Four Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_6_PI</b> 18.8495559215387594	Six Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_8_PI</b> 25.1327412287183459	Eighth Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_DAYS_PER_HOUR</b> 0.04166666666666666666666666666666667	Number of days per hour (1/24); used for time related calculations
<b>LMATH_DEGREE_TO_RAD</b> 0.0174532925199432958	Conversion for degree to radians = $\pi / 180$
<b>LMATH_E</b> 2.71828182845904523536028747135 266	Euler's number, the base of the natural logarithm (Napier's constant)
<b>LMATH_E_INV</b> 0.36787944117144232159552377016 147	Inverse Euler's number, the inverse base of the natural logarithm (Napier's constant)
<b>LMATH_EXP_INV_E</b> 1.444667861009766	Tetration right convergence limit $e^{(1/e)}$
<b>LMATH_EXP_MINUS_E</b> 0.06598803584531256	Tetration left convergence limit $e^{(-e)}$
<b>LMATH_FACULTY_00</b> UDINT#1	Faculty calculated number of 0
<b>LMATH_FACULTY_01</b> UDINT#1	Faculty calculated number of 1
<b>LMATH_FACULTY_02</b> UDINT#2	Faculty calculated number of 2
<b>LMATH_FACULTY_03</b> UDINT#6	Faculty calculated number of 3
<b>LMATH_FACULTY_04</b> UDINT#24	Faculty calculated number of 4
<b>LMATH_FACULTY_05</b> UDINT#120	Faculty calculated number of 5
<b>LMATH_FACULTY_06</b> UDINT#720	Faculty calculated number of 6
<b>LMATH_FACULTY_07</b> UDINT#5040	Faculty calculated number of 7
<b>LMATH_FACULTY_08</b> UDINT#40320	Faculty calculated number of 8
<b>LMATH_FACULTY_09</b> UDINT#362880	Faculty calculated number of 9
<b>LMATH_FACULTY_10</b> UDINT#3628800	Faculty calculated number of 10

6 PLC tags & constants

Identifier & Value	Description
<b>LMATH_FACULTY_11</b> UDINT#39916800	Faculty calculated number of 11
<b>LMATH_FACULTY_12</b> UDINT#479001600	Faculty calculated number of 12
<b>LMATH_HOURS_PER_DAY</b> 24	Number of hours per day; used for time related calculations
<b>LMATH_HOURS_PER_MINUTE</b> 0.016666666666666666666666666666666667	Number of hours per minute (1/60); used for time related calculations
<b>LMATH_HOURS_PER_YEAR</b> 8760	Number of hours per year; used for time related calculations
<b>LMATH_LI2</b> 1.04516378011749278484458888919461	Natural logarithm integral function li(2)
<b>LMATH_LN_PI</b> 1.14472988585	Natural logarithm of pi
<b>LMATH_LN_SQRT2</b> 0.34657359028	Natural logarithm of sqrt(2)
<b>LMATH_LN10</b> 2.30258509299404568401799145468440	Natural logarithm of ten - needed for example for the ten's logarithm conversion / calculation
<b>LMATH_LN2</b> 0.69314718055994530941723212145818	Natural logarithm of two - needed for example for the binary logarithm conversion / calculation
<b>LMATH_LREAL_INFINITY_NEG</b> 16#FFF0000000000000	LReal / Double - negative Infinity
<b>LMATH_LREAL_INFINITY_POS</b> 16#7FF0000000000000	LReal / Double - positive Infinity
<b>LMATH_LREAL_MAX_NORM</b> 16#7FEFFFFFFFFFFFFFFF	LReal / Double - maximum value in normalized format
<b>LMATH_LREAL_MAX_SUBNORM</b> 16#000FFFFFFFFFFFFFFF	LReal / Double - maximum value in denormalized format
<b>LMATH_LREAL_MIN_NORM</b> 16#0010000000000000	LReal / Double - minimum value in normalized format
<b>LMATH_LREAL_MIN_SUBNORM</b> 16#0000000000000001	LReal / Double - minimum value in denormalized format
<b>LMATH_LREAL_NAN</b> 16#7FF8000000000000	LReal / Double - NAN - Not A Number
<b>LMATH_MINUTES_PER_HOUR</b> 60	Number of minutes per hour; used for time related calculations
<b>LMATH_MINUTES_PER_SECOND</b> 0.016666666666666666666666666666666667	Number of minutes per second (1/60); used for time related calculations
<b>LMATH_ONE_EIGHTH_PI</b> 0.039788735772973834	Eighth part of Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_ONE_HALF_PI</b> 1.57079632679489661923132169163980	Half Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_ONE_QUARTER_PI</b> 0.78539816339744830961566084581988	Quarter Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_ONE_SIXTH_PI</b> 0.0530516476972984453	Sixth Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)

## 6 PLC tags & constants

Identifier & Value	Description
<b>LMATH_ONE_THIRD_PI</b> 0.1061032953945968907	One-third Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_PHI_GOLDEN_RATIO</b> 1.61803398874989484820458683436563	Golden ratio is also called the golden mean or golden section (Latin: sectio aurea)
<b>LMATH_PHI_LN_GOLDEN_RATIO</b> 0.48121182505960344749775891342436341	Natural logarithm of golden ratio is also called the golden mean or golden section (Latin: sectio aurea)
<b>LMATH_PI</b> 3.14159265358979323846264338327950	Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_PI_INV</b> 0.31830988618379067153776752674503	Inverse Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_RAD_TO_DEGREE</b> 57.2957795130823209	Conversion for radians to degrees = $180 / \pi$
<b>LMATH_REAL_INFINITY_NEG</b> 16#FF800000	Real / Float - negative Infinity
<b>LMATH_REAL_INFINITY_POS</b> 16#7F800000	Real / Float - positive Infinity
<b>LMATH_REAL_MAX_NORM</b> 16#7FFFFFFF	Real / Float - maximum value in normalized format
<b>LMATH_REAL_MAX_SUBNORM</b> 16#007FFFFFFF	Real / Float - maximum value in denormalized format
<b>LMATH_REAL_MIN_NORM</b> 16#00800000	Real / Float - minimum value in normalized format
<b>LMATH_REAL_MIN_SUBNORM</b> 16#00000001	Real / Float - minimum value in denormalized format
<b>LMATH_REAL_NAN</b> 16#7FC00000	Real / Float - NAN - Not A Number
<b>LMATH_SECONDS_PER_MINUTE</b> 60	Number of seconds in a minute; used for time related calculations
<b>LMATH_SQRT_2_PI</b> 2.50662827463100050241576528481104	Square root of two Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_SQRT_PI</b> 1.77245385090551602729816748334114	Square root of Pi, the ratio of a circle's circumference to its diameter (Archimedes' constant or Ludolph's number)
<b>LMATH_SQRT2</b> 1.41421356237309504880168872420970	Square root of 2 (Pythagoras' constant)
<b>LMATH_SQRT2_INV</b> 0.70710678118654752440084436210485	Inverse square root of 2 (Pythagoras' constant)
<b>LMATH_SQRT3</b> 1.73205080756887729352744634150590	Square root of 3 (Theodorus' constant)
<b>LMATH_SQRT3_INV</b> 0.57735026918962576450914878050195	Inverse square root of 3 (Theodorus' constant)
<b>LMATH_SQRT5</b> 2.23606797749978969	Square root of 5 (This number appears in the fractional expression for the golden ratio)
<b>LMATH_SQRT5_INV</b> 0.4472135954999579386	Inverse square root of 5 (This number appears in the fractional expression for the golden ratio)

## 6 PLC tags & constants

---

Identifier & Value	Description
<b>LMATH_YEARS_PER_HOUR</b> 1.14155251141552511415525114155 25e-4	Number of years per hour (1/8760); used for time related calculations

### 6.1.4 LMATH\_CONSTANTS\_EXTENDED

Constant identifier, values and description

Identifier & Value	Description
<b>LMATH_ALLADI_GRINSTEAD</b> 0.8093940205	Alladi-Grinstead constant is the Infinite product constant
<b>LMATH_APERY</b> 1.2020569031595942853997381615 1145	Apery's constant
<b>LMATH_BACKHOUSE</b> 1.4560749485826896713995953511 1654	Backhouse's constant
<b>LMATH_BERNSTEIN</b> 0.28016949902386913303	Bernstein's constant
<b>LMATH_BRAUN_PRIME_QUADR</b> 0.8705883800	Braun's constant for prime quadruplets
<b>LMATH_BRAUN_TWIN_PRIME</b> 1.9021605823	Braun's constant for twin primes
<b>LMATH_BRUIJN_NEWMAN</b> -2.7E-9	de Bruijn-Newman constant
<b>LMATH_CAHEN</b> 0.6434105463	Cahen's constant is defined as an infinite series of unit fractions, with alternating signs, derived from Sylvester's sequence
<b>LMATH_CATALAN</b> 0.9159655941772190150546035149 3238	Catalan's constant G, which appears in combinatorics
<b>LMATH_EMBREE_TREFETHEN</b> 0.70258	Embree-Trefethen constant is a threshold value labelled $\beta^*$
<b>LMATH_ERDOS_BORWEIN</b> 1.6066951524152917637833015231 9092	Erdos-Borwein constant is the sum of the reciprocals of the Mersenne numbers
<b>LMATH_EULER_MASCHERONI</b> 0.5772156649015328606065120900 8240	Euler-Mascheroni constant recurring in analysis and number theory, usually denoted by the lowercase Greek letter gamma ( $\gamma$ ).
<b>LMATH_EULER_MASCHERONI_IN V</b> 1.7324547146006334735830253158 6084	Euler-Mascheroni inverse constant recurring in analysis and number theory, usually denoted by the lowercase Greek letter gamma ( $\gamma$ ).
<b>LMATH_EULER_MASCHERONI_SQ R</b> 0.3331779238077186743183761363 5524	Euler-Mascheroni square constant recurring in analysis and number theory, usually denoted by the lowercase Greek letter gamma ( $\gamma$ ).
<b>LMATH_FEIGENBAUM_ALFA</b> 2.5029078750958928222839028732 1822	Feigenbaum constants are two mathematical constants which both express ratios in a bifurcation diagram for a non-linear map
<b>LMATH_FEIGENBAUM_DELTA</b> 4.6692016091029906718532038204 6620	Feigenbaum constants are two mathematical constants which both express ratios in a bifurcation diagram for a non-linear map
<b>LMATH_FRANSEN_ROBINSON</b> 2.8077702420285193652215011865 5777	Fransén-Robinson constant, sometimes denoted F, is the mathematical constant that represents the area between the graph of the reciprocal Gamma function, $1 / \Gamma(x)$ , and the positive x axis
<b>LMATH_GAUSS_KUZMIN_WIRSIN G</b> 0.3036630028987326585974481219 0156	Gauss-Kuzmin-Wirsing constant is the transfer operator of the Gauss map



## 6 PLC tags &amp; constants

Identifier & Value	Description
<b>LMATH_GOLOMB_DICKMAN</b> 0.6243299885435508709929363831 0083	Golomb-Dickman constant arises in the theory of random permutations and in number theory
<b>LMATH_GOMPERTZ</b> 0.5963473623231940743410784993 6928	Gompertz Constant OEIS A073003
<b>LMATH_HAFNER_SARNAK_MCCURLEY</b> 0.35323637185499598454	Hafner-Sarnak-McCurley constant representing the probability that the determinants of two randomly chosen square integer matrices will be relatively prime
<b>LMATH_KHINCHIN</b> 2.6854520010653064453097148354 8180	Khinchin's constant for almost all real numbers $x$ , coefficients $a_i$ of the continued fraction expansion of $x$ have a finite geometric mean that is independent of the value of $x$ and is known as Khinchin's constant
<b>LMATH_LANDAU</b> 0.5	Landau's constant
<b>LMATH_LANDAU_RAMANUJAN</b> 0.7642236535892206629906987312 5009	Landau-Ramanujan constant
<b>LMATH_LAPLACE_LIMIT</b> 0.6627434193491815809747420971 0925	Laplace limit is the maximum value of the eccentricity for which a solution to Kepler's equation, in terms of a power series in the eccentricity, converges
<b>LMATH_LEGENDRE</b> 1.0	Legendre's constant, capture the asymptotic behavior of the prime-counting function $\pi(x)$ . Its value is now known to be exactly 1.
<b>LMATH LENGYEL</b> 1.0986858055	Lengyel's constant
<b>LMATH_LEVY</b> 3.2758229187218111597876818824 5384	Levy's constant occurs in an expression for the asymptotic behavior of the denominators of the convergent of continued fractions
<b>LMATH_LIEB_QUARE_ICE</b> 1.5396007178	Lieb's square ice constant is a mathematical constant used in the field of combinatorics to quantify the number of Eulerian orientations of grid graphs.
<b>LMATH_MEISSEL_MERTEENS</b> 0.2614972128476427837554268386 0870	Meissel–Mertens constant limiting difference between the harmonic series summed only over the primes and the natural logarithm of the natural logarithm
<b>LMATH_MILLS</b> 1.3063778838630806904686144926 0261	Mills' constant is defined as the smallest positive real number $A$ such that the floor function of the double exponential function $[A^{3^n}]$ is a prime number, for all natural numbers $n$ .
<b>LMATH_MRB</b> 0.187859	MRB constant is defined as the upper limit of the partial sums
<b>LMATH_NIVEN</b> 1.7052111401053677642885514534 3451	Niven's constant is the largest exponent appearing in the prime factorization of any natural number $n$ "on average"
<b>LMATH_OMEGA</b> 0.5671432904097838729999686622 1036	Omega constant defined as the unique real number that satisfies the equation - Omega $e^{\Omega} = 1$
<b>LMATH_PARABOLIC</b> 2.2955871493926380740342980491 8949	Universal parabolic constant is defined as the ratio, for any parabola, of the arc length of the parabolic segment formed by the latus rectum to the focal parameter.
<b>LMATH_PLASTIC_RATIO</b> 1.3247179572447460259609088544 7810	Plastic number the unique real solution of the cubic equation $x^3 = x + 1$
<b>LMATH_PORTER</b> 1.4670780794	Porter's constant $C$ arises in the study of the efficiency of the Euclidean algorithm

## 6 PLC tags & constants

Identifier & Value	Description
<b>LMATH_RAMANUJAN_SOLDNER</b> 1.4513692348833810502839684858 9203	Ramanujan-Soldner constant defined as the unique positive zero of the logarithmic integral function
<b>LMATH_SIERPINSKI</b> 2.5849817595792532170658935873 8317	Sierpiński's constant usually denoted as K.
<b>LMATH_TWIN_PRIME</b> 0.6601618158468695739278121100 1456	twin prime constant C2
<b>LMATH_VISWANATH</b> 1.1319882487943	Viswanath's constant, the growth rate of the random Fibonacci sequence is equal to

### 6.1.5 LPHYSICS\_CONSTANTS

Constant identifier, values and description

Identifier & Value	Description
<b>LPHYSIC_B_FREQUENCY</b> 5.8789238E+10	Wien frequency displacement law constant - Hz / K
<b>LPHYSIC_B_LAMBDA</b> 2.8977685E-3	Wien wavelength displacement law constant - m*K
<b>LPHYSIC_C_0</b> 2.99792458E+08	Speed of light in vacuum - m/s
<b>LPHYSIC_COEF_NI_A</b> 0.5485	Coefficients for temperature calculation - Nickel
<b>LPHYSIC_COEF_NI_B</b> 0.665E-3	Coefficients for temperature calculation - Nickel
<b>LPHYSIC_COEF_NI_C</b> 2.805E-9	Coefficients for temperature calculation - Nickel
<b>LPHYSIC_COEF_PT_A</b> 3.90802E-3	Coefficients for temperature calculation - Platin
<b>LPHYSIC_COEF_PT_A_TEMP</b> 3.9083E-3	Coefficients for temperature calculation - Platin
<b>LPHYSIC_COEF_PT_B</b> -5.802000E-007	Coefficients for temperature calculation - Platin
<b>LPHYSIC_COEF_PT_B_TEMP</b> -5.775000E-007	Coefficients for temperature calculation - Platin
<b>LPHYSIC_COEF_PT_C</b> -4.273500E-012	Coefficients for temperature calculation - Platin
<b>LPHYSIC_COEF_PT_TEMP_ACCURACY</b> 0.01	Coefficients for temperature calculation - Platin
<b>LPHYSIC_COEF_SI_A</b> 7.64E-3	Coefficients for temperature calculation - Silicium
<b>LPHYSIC_COEF_SI_B</b> 1.66E-5	Coefficients for temperature calculation - Silicium
<b>LPHYSIC_E</b> 1.602176487E-19	Elementary charge - C [A*s]
<b>LPHYSIC_EPSILON_ZERO</b> 8.854187817E-12	Vacuum permittivity, permittivity of free space or electric constant or the distributed capacitance of the vacuum - F/m
<b>LPHYSIC_F</b> 9.64853399E+4	Faraday constant - C/mol
<b>LPHYSIC_G</b> 6.67428E-11	Gravidity constant - N*m <sup>2</sup> /kg <sup>2</sup>
<b>LPHYSIC_G_N</b> 9.80665	Regular acceleration of gravidity at earth - m/s <sup>2</sup>
<b>LPHYSIC_H</b> 6.62606896E-34	Plank constant - J*s
<b>LPHYSIC_K</b> 1.3806504E-19	Boltzmann constant - J/K
<b>LPHYSIC_MU_ZERO</b> 1.2566370614E-6	Vacuum permeability, permeability of free space, permeability of vacuum, or magnetic constant, is the magnetic permeability in a classical vacuum - V*s/A*m
<b>LPHYSIC_N_A</b> 6.022140857E23	Avogadro constant - mol

## 6 PLC tags & constants

Identifier & Value	Description
<b>LPHYSIC_P_N</b> 101325.0	Normal standard pressure / standard atmosphere ATM - Pa
<b>LPHYSIC_PLANCK_CHARGE</b> 1.875545956E-18	Plank charge constant - C [A*s]
<b>LPHYSIC_PLANCK_H_REDUCED</b> 1.054571628E-34	Plank constant divided by 2 PI, is the quantization of angular momentum - J*s
<b>LPHYSIC_PLANCK_LENGTH</b> 1.616229E-35	Plank length constant - m
<b>LPHYSIC_PLANCK_MASS</b> 2.176470E-8	Plank mass constant - kg
<b>LPHYSIC_PLANCK_TEMPERATUR E</b> 1.416808E+32	Plank temperature constant - K
<b>LPHYSIC_PLANCK_TIME</b> 5.39116E-44	Plank time constant - s
<b>LPHYSIC_R</b> 8.31447215	Gas constant - universal - J/(mol*K)
<b>LPHYSIC_R_AIR</b> 287.058	Gas constant - regular dry air - J/(mol*K)
<b>LPHYSIC_R_INF</b> 1.0973731568527E+7	Rydberg constant relating to the electromagnetic spectra of an atom - 1/m
<b>LPHYSIC_SIGMA</b> 5.670367E-8	Stefan's constant, "the total intensity radiated over all wavelengths increases as the temperature increases" - W/(m <sup>2</sup> *K <sup>4</sup> )
<b>LPHYSIC_T_0</b> -273.15	Zero degree temperature offset from Celsius to Kelvin
<b>LPHYSIC_U</b> 1.660538782E-27	Atomic mass unit - kg
<b>LPHYSIC_Z0</b> 376.73031346177	Impedance of free space, square root of $\mu_0 / \epsilon_0$ - Ohm

# 7 Appendix

## 7.1 Service and support

### Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos, all information is accessible with just a few mouse clicks:

<https://support.industry.siemens.com>

### Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers - ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

<https://www.siemens.com/supportrequest>

### SITRAIN - Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

<https://www.siemens.com/sitrain>

### Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contract's

You can find detailed information on our range of services in the service catalog web page:

<https://support.industry.siemens.com/cs/sc>

### Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

<https://support.industry.siemens.com/cs/ww/en/sc/2067>

## 7.2 Industry Mall



The Siemens Industry Mall is the platform on which the entire Siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:

<https://mall.industry.siemens.com>

## 7.3 Links and Literature

No.	Topic
\1\	Siemens Industry Online Support <a href="https://support.industry.siemens.com">https://support.industry.siemens.com</a>
\2\	Link to the entry page of the application exampl <a href="https://support.industry.siemens.com/cs/ww/en/view/109479728">https://support.industry.siemens.com/cs/ww/en/view/109479728</a>
\3\	Programming Guidelines and Programming Style guide for SIMATIC S7-1200 and S7-1500 <a href="https://support.industry.siemens.com/cs/ww/en/view/81318674">https://support.industry.siemens.com/cs/ww/en/view/81318674</a>
\4\	Library with PLC data types (LPD) for STEP 7 (TIA Portal) and SIMATIC S7-1200 / S7-1500 <a href="https://support.industry.siemens.com/cs/ww/en/view/109482396">https://support.industry.siemens.com/cs/ww/en/view/109482396</a>
\5\	Guideline on Library Handling in Tia Portal <a href="https://support.industry.siemens.com/cs/ww/en/view/109747503">https://support.industry.siemens.com/cs/ww/en/view/109747503</a>
\6\	Libraries in the TIA Portal <a href="https://support.industry.siemens.com/cs/ww/en/view/109738702">https://support.industry.siemens.com/cs/ww/en/view/109738702</a>

## 7.4 Change documentation

### Versioning of the library

The library and library elements are maintained in accordance with the table below:

P	a.	b.	c.
	Non-compatible change	Compatible change	Error correction
	<ul style="list-style-type: none"> <li>- Reduction of interfaces</li> <li>- Changing the interfaces</li> <li>- Incompatible extension of functionality</li> </ul>	<ul style="list-style-type: none"> <li>- Extension of the interfaces</li> <li>- Compatible extension of functionality</li> </ul>	<ul style="list-style-type: none"> <li>- Bug fix</li> </ul>

### Versioning example

Example for changing the version:

Library	FB1	FB2	FC1	FC2	Comment
1.0.0	1.0.0	1.0.0	1.0.0	-	Released
1.0.1	1.0.1	1.0.0	1.0.0	-	Troubleshooting of FB1
1.0.2	1.0.1	1.0.1	1.0.0	-	Optimization of FB2
1.1.0	1.1.0	1.0.1	1.0.0	-	Extension to FB1
1.2.0	1.2.0	1.0.1	1.0.0	-	Extension to FB1
2.0.0	2.0.0	1.0.1	2.0.0	-	New functionality on FB1 and FC1
2.0.1	2.0.0	1.0.2	2.0.0	-	Troubleshooting FB2
3.0.0	2.0.0	1.0.2	2.0.0	1.0.0	New function FC2
3.0.1	2.0.1	1.0.3	2.0.1	1.0.1	Upgrade to new TIA Portal version
3.0.2	2.0.2	1.0.4	2.0.2	1.0.1	New functions, bug fixes

## 8 Change log

Version & Date	Change description
<b>V5.1.0</b> 11/2021	<p><b><u>NEW:</u></b>            LGF_BinaryMaskCompare / V01.00.00            LGF_CountBooleanEdges / V01.00.00            LGF_GetBitStates / V01.00.00            LGF_ShiftRegister / V03.00.00            ++++++</p> <p><b><u>UPDATED:</u></b>            LGF_BitReset / V03.00.01            • Insert documentation            LGF_BitSet / V03.00.01            • Insert documentation            LGF_BitSetTo / V03.00.01            • Insert documentation            LGF_BitTest / V03.00.01            • Insert documentation            LGF_BitToggle / V03.00.01            • Insert documentation            LGF_PulseRelay / V03.00.01            • Insert documentation            LGF_AstroClock / V03.00.01            • Bug fix - not enabled - block still running            • Insert documentation            LGF_GetCalendarDay / V03.00.01            • Insert documentation            LGF_GetCalendarWeek_ISO / V03.00.01            • Insert documentation            LGF_GetCalendarWeek_US / V03.00.01            • Insert documentation            LGF_IsGermanHoliday / V03.00.01            • fix bug in Constant "DAYS_AFTER_EASTER_60" from 6 to 60            • Insert documentation            LGF_SetTime / V03.00.02            • Bug fix - bias correction for time offsets (200 / 330)            • Insert documentation            LGF_TimerSwitch / V03.00.01            • Insert documentation            LGF_BitCount / V03.00.02            • Insert documentation            LGF_CountFallInDWord / V03.00.01            • Insert documentation            LGF_CountRisInDWord / V03.00.01            • Insert documentation            LGF_CompareLReal / V03.00.01            • Insert documentation            LGF_CompareLRealByPrecision / V03.00.01            • Insert documentation            LGF_CompareVariant / V03.00.01            • Insert documentation</p>



## 8 Change log

Version & Date	Change description
	<p>LGF_CalcDistance_2D / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_CalcDistance_3D / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_GetFactorial / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_Integration / V03.00.02</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> <li>• Fix bug - incompatibility with S7-1200 and LTIME</li> </ul> <p>LGF_IsValueInLimits / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_IsValueInRange / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_IsValueInTolerance / V03.00.02</p> <ul style="list-style-type: none"> <li>• Bug fix - negative setpoint verification</li> <li>• Insert documentation</li> </ul> <p>LGF_NthRoot / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_Random_DInt / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_Random_Real / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_Random_UDInt / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_RandomRange_Dint / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_RandomRange_Real / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_RandomRange_UDInt / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_ScaleLinear / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> <li>• Move to folder "Math operations"</li> </ul> <p>LGF_SearchMinMax / V03.00.01</p> <ul style="list-style-type: none"> <li>• Rework constants and comments</li> <li>• Insert documentation</li> </ul> <p>LGF_SearchMinMax_DInt / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_SearchMinMax_LReal / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_SearchMinMax_UDInt / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_StoreMinMax / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_MatrixAddition / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_MatrixCompare / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_MatrixInverse / V03.00.01</p> <ul style="list-style-type: none"> <li>• Insert documentation</li> </ul> <p>LGF_MatrixMultiplication / V03.00.01</p>

Version & Date	Change description
	<ul style="list-style-type: none"> <li>• Insert documentation LGF_MatrixScalarMultiplication / V03.00.01</li> <li>• Insert documentation LGF_MatrixSubtraction / V03.00.01</li> <li>• Insert documentation LGF_MatrixTranspose / V03.00.01</li> <li>• Insert documentation LGF_CalcCRC16 / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code> LGF_CalcCRC16Advanced / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code>, <code>finalXorValue</code>, <code>reflectInput</code>, <code>reflectResult</code> LGF_CalcCRC32 / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code> LGF_CalcCRC32Advanced / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code>, <code>finalXorValue</code>, <code>reflectInput</code>, <code>reflectResult</code> LGF_CalcCRC8 / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code> LGF_CalcCRC8Advanced / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code>, <code>finalXorValue</code>, <code>reflectInput</code>, <code>reflectResult</code> LGF_CalcCRC8For1Byte / V03.00.01</li> <li>• Insert documentation • Assign default start values to optional inputs - <code>initValue</code>, <code>mask</code> LGF_FIFO / V03.00.01</li> <li>• Insert documentation LGF_IsParityEven / V03.00.01</li> <li>• Insert documentation LGF_IsParityOdd / V03.00.01</li> <li>• Insert documentation LGF_LIFO / V03.00.01</li> <li>• Insert documentation LGF_ShellSort_DInt / V03.00.01</li> <li>• Insert documentation LGF_ShellSort_LReal / V03.00.01</li> <li>• Insert documentation LGF_ShellSort_UDInt / V03.00.01</li> <li>• Insert documentation LGF_BinaryToGray / V03.00.01</li> <li>• Insert documentation LGF_DTLtoString_DE / V03.00.01</li> <li>• Insert documentation LGF_DTLtoString_ISO / V03.00.01</li> <li>• Insert documentation LGF_DTLToUnixTime / V03.00.01</li> </ul>

## 8 Change log

Version & Date	Change description
	<ul style="list-style-type: none"> <li>• Insert documentation LGF_GpsDDToGps / V03.00.02</li> <li>• Fix tempStatus initialization</li> <li>• Insert documentation LGF_GpsToGpsDD / V03.00.02</li> <li>• Fix tempStatus initialization</li> <li>• Insert documentation LGF_GrayToBinary / V03.00.01</li> <li>• Insert documentation LGF_IntToString / V03.00.01</li> <li>• Insert documentation LGF_StringToDTL_DE / V03.00.01</li> <li>• Insert documentation LGF_StringToDTL_ISO / V03.00.01</li> <li>• Insert documentation LGF_StringToInt / V03.00.01</li> <li>• Insert documentation</li> <li>• ENO handling done by STRG_VAL system function LGF_StringToTaddr / V03.00.01</li> <li>• Insert documentation LGF_StringToTime / V03.00.01</li> <li>• Insert documentation LGF_TaddrToString / V03.00.01</li> <li>• Insert documentation LGF_TimeToString / V03.00.01</li> <li>• Insert documentation LGF_UnixTimeToDTL / V03.00.01</li> <li>• Insert documentation LGF_MergeBitsToDWord / V03.00.01</li> <li>• Insert documentation LGF_MergeBitsToWord / V03.00.01</li> <li>• Insert documentation LGF_MergeBytesToDWord / V03.00.01</li> <li>• Insert documentation LGF_MergeBytesToWord / V03.00.01</li> <li>• Insert documentation LGF_MergeWordsToDWord / V03.00.01</li> <li>• Insert documentation LGF_SplitByteToBits / V03.00.01</li> <li>• Insert documentation LGF_SplitDWordToBits / V03.00.01</li> <li>• Insert documentation LGF_SplitDWordToBytes / V03.00.01</li> <li>• Insert documentation LGF_SplitDWordToWords / V03.00.01</li> <li>• Insert documentation LGF_SplitWordToBits / V03.00.01</li> <li>• Insert documentation LGF_SplitWordToBytes / V03.00.01</li> <li>• Insert documentation LGF_CelsiusToFahrenheit / V03.00.01</li> </ul>

Version & Date	Change description
	<ul style="list-style-type: none"> <li>• Insert documentation LGF_CelsiusToKelvin / V03.00.01</li> <li>• Insert documentation LGF_ConvertTemperature / V03.00.01</li> <li>• Rename from “LGF_TemperatureConvert” to “LGF_ConvertTemperature”</li> <li>• to start with the verb</li> <li>• include the Rankine conversion</li> <li>• Code refactoring, regions, comments and constants</li> <li>• Set version to V3.0.0</li> <li>• harmonize the version of the whole library</li> <li>• Insert documentation LGF_FahrenheitToCelsius / V03.00.01</li> <li>• Insert documentation LGF_FahrenheitToKelvin / V03.00.01</li> <li>• Insert documentation LGF_KelvinToCelsius / V03.00.01</li> <li>• Insert documentation LGF_KelvinToFahrenheit / V03.00.01</li> <li>• Insert documentation LGF_KelvinToRankine / V03.00.01</li> <li>• Insert documentation LGF_RankineToKelvin / V03.00.01</li> <li>• Insert documentation LGF_CosinusCI / V03.00.01</li> <li>• Insert documentation LGF_Frequency / V03.00.01</li> <li>• Insert documentation LGF_Impulse / V03.00.01</li> <li>• Insert documentation LGF_RectangleCI / V03.00.01</li> <li>• Insert documentation LGF_SawToothCI / V03.00.01</li> <li>• Insert documentation LGF_SinusCI / V03.00.01</li> <li>• Insert documentation LGF_TriangleCI / V03.00.01</li> <li>• Insert documentation LGF_LimRateOfChangeAdvancedCI / V03.00.01</li> <li>• Insert documentation LGF_LimRateOfChangeCI / V03.00.01</li> <li>• Insert documentation LGF_NonLinearInterpolation / V03.00.01</li> <li>• Insert documentation LGF_RampCI / V03.00.01</li> <li>• Insert documentation</li> <li>• Change UDT member name from <code>outValue</code> to <code>outputValue</code></li> <li>• Insert documentation LGF_AverageAndDeviation / V03.00.01</li> <li>• Insert documentation LGF_Boxplot_DInt / V03.00.01</li> <li>• Insert documentation LGF_Boxplot_LReal / V03.00.01</li> </ul>

## 8 Change log

Version & Date	Change description
	<ul style="list-style-type: none"> <li>• Insert documentation LGF_Boxplot_UDInt / V03.00.01</li> <li>• Insert documentation LGF_DifferenceQuotientFB / V03.00.01</li> <li>• Insert documentation LGF_DifferenceQuotientFC / V03.00.01</li> <li>• Insert documentation LGF_FloatingAverage / V03.00.02</li> <li>• Insert documentation LGF_Histogram_DInt / V03.00.01</li> <li>• Insert documentation LGF_Histogram_LReal / V03.00.01</li> <li>• Insert documentation LGF_Histogram_UDInt / V03.00.01</li> <li>• Insert documentation LGF_RegressionLine / V03.00.01</li> <li>• Insert documentation LGF_SimpleSmoothingFB / V03.00.01</li> <li>• Insert documentation LGF_SimpleSmoothingFC / V03.00.01</li> <li>• Insert documentation LGF_SmoothByPolynomFB / V03.00.01</li> <li>• Insert documentation LGF_SmoothByPolynomFC / V03.00.01</li> <li>• Insert documentation LGF_SawTooth / V03.00.01</li> <li>• Insert documentation</li> </ul>
<b>V5.0.0</b> 04/2020	<b>NEW:</b> LGF_BitReset / V03.00.00 LGF_BitSet / V03.00.00 LGF_BitSetTo / V03.00.00 LGF_BitTest / V03.00.00 LGF_BitToggle / V03.00.00 LGF_PulseRelay / V03.00.00 LGF_AstroClock / V03.00.00 LGF_GetCalendarDay / V03.00.00 LGF_GetCalendarWeek_ISO / V03.00.00 LGF_GetCalendarWeek_US / V03.00.00 LGF_IsGermanHoliday / V03.00.00 LGF_SetTime / V03.00.00 LGF_TimerSwitch / V03.00.00 LGF_CountFallInDWord / V03.00.00 LGF_CountRisInDWord / V03.00.00 LGF_CompareLReal / V03.00.00 LGF_CompareLRealByPrecision / V03.00.00 LGF_CompareVariant / V03.00.00 LGF_CalcDistance_2D / V03.00.00 LGF_CalcDistance_3D / V03.00.00 LGF_GetFactorial / V03.00.00 LGF_Integration / V03.00.00 LGF_IsValueInLimits / V03.00.00 LGF_IsValueInRange / V03.00.00 LGF_IsValueInTolerance / V03.00.00 LGF_NthRoot / V03.00.00 LGF_Random_DInt / V03.00.00 LGF_Random_Real / V03.00.00 LGF_Random_UDInt / V03.00.00

## 8 Change log

Version & Date	Change description
	LGF_RandomRange_DInt / V03.00.00
	LGF_RandomRange_Real / V03.00.00
	LGF_RandomRange_UDInt / V03.00.00
	LGF_SearchMinMax / V03.00.00
	LGF_SearchMinMax_DInt / V03.00.00
	LGF_SearchMinMax_LReal / V03.00.00
	LGF_SearchMinMax_UDInt / V03.00.00
	LGF_StoreMinMax / V03.00.00
	LGF_MatrixInverse / V03.00.00
	LGF_MatrixTranspose / V03.00.00
	LGF_MatrixAddition / V03.00.00
	LGF_MatrixCompare / V03.00.00
	LGF_MatrixMultiplication / V03.00.00
	LGF_MatrixScalarMultiplication / V03.00.00
	LGF_MatrixSubtraction / V03.00.00
	LGF_CalcCRC16 / V03.00.00
	LGF_CalcCRC16Advanced / V03.00.00
	LGF_CalcCRC32 / V03.00.00
	LGF_CalcCRC32Advanced / V03.00.00
	LGF_CalcCRC8 / V03.00.00
	LGF_CalcCRC8Advanced / V03.00.00
	LGF_CalcCRC8For1Byte / V03.00.00
	LGF_FIFO / V03.00.00
	LGF_IsParityEven / V03.00.00
	LGF_IsParityOdd / V03.00.00
	LGF_LIFO / V03.00.00
	LGF_ShellSort_DInt / V03.00.00
	LGF_ShellSort_LReal / V03.00.00
	LGF_ShellSort_UDInt / V03.00.00
	LGF_BinaryToGray / V03.00.00
	LGF_DTLtoString_DE / V03.00.00
	LGF_DTLtoString_ISO / V03.00.00
	LGF_DTLtoUnixTime / V03.00.00
	LGF_GpsDDToGps / V03.00.00
	LGF_GpsToGpsDD / V03.00.00
	LGF_GrayToBinary / V03.00.00
	LGF_IntToString / V03.00.00
	LGF_ScaleLinear / V03.00.00
	LGF_StringToDTL_DE / V03.00.00
	LGF_StringToDTL_ISO / V03.00.00
	LGF_StringToInt / V03.00.00
	LGF_StringToTaddr / V03.00.00
	LGF_StringToTime / V03.00.00
	LGF_TaddrToString / V03.00.00
	LGF_TimeToString / V03.00.00
	LGF_UnixTimeToDTL / V03.00.00
	LGF_CelsiusToFahrenheit / V03.00.00
	LGF_CelsiusToKelvin / V03.00.00
	LGF_ConvertTemperature / V16.12.2018
	LGF_FahrenheitToCelsius / V03.00.00
	LGF_FahrenheitToKelvin / V03.00.00
	LGF_KelvinToCelsius / V03.00.00
	LGF_KelvinToFahrenheit / V03.00.00
	LGF_KelvinToRankine / V03.00.00
	LGF_RankineToKelvin / V03.00.00
	LGF_CosinusCI / V03.00.00
	LGF_Frequency / V03.00.00
	LGF_Impulse / V03.00.00
	LGF_RectangleCI / V03.00.00
	LGF_SawToothCI / V03.00.00
	LGF_SinusCI / V03.00.00
	LGF_TriangleCI / V03.00.00
	LGF_LimRateOfChangeAdvancedCI / V03.00.00

## 8 Change log

Version & Date	Change description
	LGF_LimRateOfChangeCI / V03.00.00
	LGF_NonLinearInterpolation / V03.00.00
	LGF_RampCI / V03.00.00
	LGF_AverageAndDeviation / V03.00.00
	LGF_Boxplot_DInt / V03.00.00
	LGF_Boxplot_LReal / V03.00.00
	LGF_Boxplot_UDInt / V03.00.00
	LGF_DifferenceQuotientFB / V03.00.00
	LGF_DifferenceQuotientFC / V03.00.00
	LGF_FloatingAverage / V03.00.01
	LGF_Histogram_DInt / V03.00.00
	LGF_Histogram_LReal / V03.00.00
	LGF_Histogram_UDInt / V03.00.00
	LGF_RegressionLine / V03.00.00
	LGF_SimpleSmoothingFB / V03.00.00
	LGF_SimpleSmoothingFC / V03.00.00
	LGF_SmoothByPolynomFB / V03.00.00
	LGF_SmoothByPolynomFC / V03.00.00
	LGF_SawTooth / V01.00.10
	LGF_BitCount / V03.00.00
	LGF_MergeBitsToByte / V03.00.01
	LGF_MergeBitsToDWord / V03.00.00
	LGF_MergeBitsToWord / V03.00.00
	LGF_MergeBytesToDWord / V03.00.00
	LGF_MergeBytesToWord / V03.00.00
	LGF_MergeWordsToDWord / V03.00.00
	LGF_SplitByteToBits / V03.00.00
	LGF_SplitDWordToBits / V03.00.00
	LGF_SplitDWordToBytes / V03.00.00
	LGF_SplitDWordToWords / V03.00.00
	LGF_SplitWordToBits / V03.00.00
	LGF_SplitWordToBytes / V03.00.00