SIEMENS

Library Description • 10/2015

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

STEP 7 (TIA Portal)

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

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these application examples and other Siemens publications − e.g. Catalogs − the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Table of Contents

1 Library Overview

1.1 General

TIA Portal hat an extensive number of “ready-to-use” instructions (mathematical functions, times, counters, etc.). In addition, there are the following useful basic functions:

These functions are supplied in form of a library and can be used unrestricted. The completed functions can be adjusted as desired and universally used.

The library described here is versioned and continuously expanded. Information on versioning is available in chapter 5.1 Explanation on versioning.

1.2 Hardware and software requirements

Requirements for this library

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 universally used with the following controller:

- S7-1200 and S7-1200 F product family
- S7-1500 and S7-1500 F product family
- Simulation with S7-PLCSIM (as of V13 SP1)

Software

- STEP 7 (TIA Portal) Basic or Professional as of V13 SP1 Update 4

Note

It is generally possible to open a library with STEP 7 basic, even though STEP 7 Professional elements (e.g. S7-1500 controller) are contained. In this case the user is informed with a message when opening the library.

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

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

2 How to work with the Library

All blocks in the “LGF” library are unrestricted in connection with S7-1200 and S7-1500 controllers.

The majority of the blocks are stored in the library as types. The blocks are hence versioned and can use all of the advantages.

- Central update function of library elements
- Versioning library elements

Note Information on the general handling of libraries is available in the programming guideline for S7-1200 and S7-1500 in chapter “Libraries”.

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

Further information on libraries is available in the TIA Portal:

- Automation Tasks in 10 minutes or less TIA-Portal: Time Savers - Global Libraries
  https://support.industry.siemens.com/cs/ww/en/view/78529894
- Which elements of STEP 7 (TIA Portal) and WinCC (TIA Portal) can you store in a library as Type or as Master Copy?
  https://support.industry.siemens.com/cs/ww/en/view/109476862
- When starting TIA Portal V13 onwards, how do you get a global library to open automatically and use it as corporate library, for example?
  https://support.industry.siemens.com/cs/ww/en/view/100451450
- How can you open a global library with write access rights in STEP 7 (TIA Portal)?
  https://support.industry.siemens.com/cs/ww/en/view/37364723

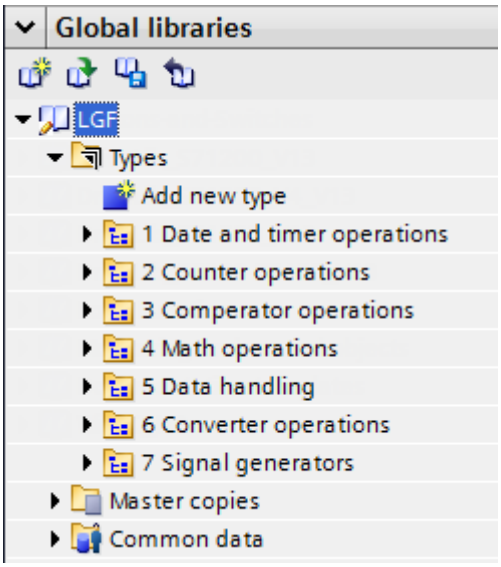
3 Explanation of the Blocks

The following chapters describe all blocks of the “Library General Functions” library. The chapters are structured according to the same structure as the library itself.

All blocks are divided into application areas or categories:

1. Date and timer operations
2. Counter operations
3. Comperator operations
4. Math operations
5. Data handling
6. Converter operations
7. Signal generators

Figure 3-1: Global library (LGF)



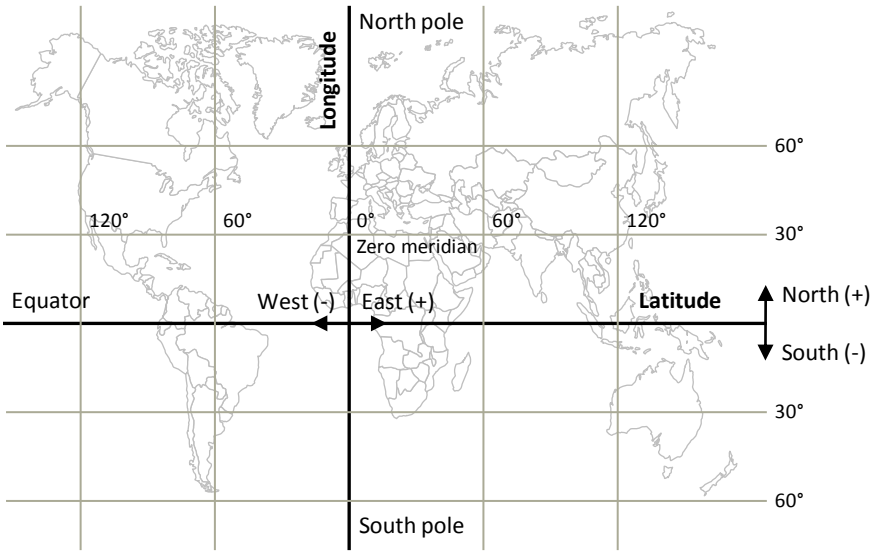
3.1 Date and timer operations

3.1.1 FB LGF_Astro

Short description

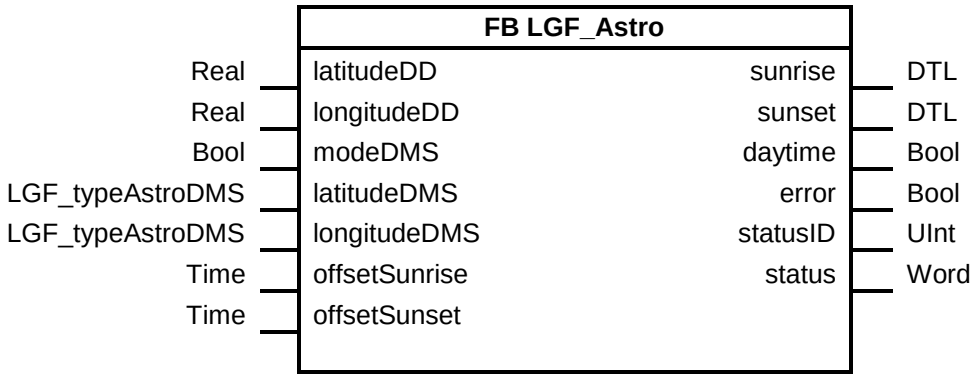
This block calculates the times of sunrise and sunset for a certain location on earth. The exact position is transferred by the block in form of geographical coordinates (longitude and latitude).

Figure 3-2: Earth with longitude and latitude



Block

Figure 3-3: BC LGF_Astro



Input parameters

Table 3-1: Input parameters

Parameter	Data type	Description
latitudeDD	Real	Latitude in degree with decimal points (unit: degrees decimal), valid value range [-90.00000..+90.00000]° This is a common coordinate format in GPX files (GPS).
longitudeDD	Real	Longitude in degree with decimal points (unit: degrees decimal)[1] valid value range [-180.0000..+180.0000]°
modeDMS	Bool	0 : transfer format of the position in “degrees decimal” via the formal parameters “latitudeDD” and “longitudeDD” 1 : transfer format of the position in direction, degrees, minutes and seconds via the formal parameters “latitudeDMS” and “longitudeDMS”
latitudeDMS	LGF_typeAstroDMS	Latitude in cardinal direction; degrees; minutes and seconds in PLC data type “LGF_typeAstroDMS”. Valid parameter values [N,S]; [0..90]; [0..59]; [0..59] Valid value range (total parameter values) [N, S, n, s]; [0..90]° This is a common coordinate format in the navigation with maps.
longitudeDMS	LGF_typeAstroDMS	Longitude in direction; degrees; minutes and seconds in PLC data type “LGF_typeAstroDMS”. Valid parameter values [E, W]; [0..180]; [0..59]; [0..59] Valid value range (total parameter values) [E, W, e, w]; [0..180]° The letter for east is internationally given as “E” (East).
offsetSunrise	Time	Offset of the switch-on time for “daytime”
offsetSunset	Time	Offset of the switch-off time for “daytime”

Output parameters

Table 3-2: Output parameters

Parameter	Data type	Description
sunrise	DTL	Sunrise at the given location taking into consideration the “offsetSunrise”
sunset	DTL	Sunset at the given location taking into consideration the “offsetSunset”
daytime	Bool	If the local time of the controller lies between “sunrise” and “sunset”, “daytime” outputs the value “TRUE”.
error	Bool	0: no errors 1: error in block, “statusID” outputs error source, “status” outputs error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-3: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	Wrong direction specified at the “latitudeDMS.dir” input	Only the following letters are permitted: N, n, S, s, W, w, E, e
1	16#8201	Wrong values at “latitiudeDMS”	Check the values at • “latitudeDMS.deg” • “latitudeDMS.min” • “latitudeDMS.sec”
1	16#8202	Wrong direction specified at “longitudeDMS.dir”	Only the following letters are permitted: N, n, S, s, W, w, E, e
1	16#8203	Wrong values at “longitudeDMS”	Check the values at • “longitudeDMS.deg” • “longitudeDMS.min” • “longitudeDMS.sec”
1	16#8204	Wrong value at input “latitudeDD”	Check the current value at the input.
1	16#8205	Wrong value at input “longitudeDD”	Check the current value at the input.
2		Error/status of subordinate block “RD_SYS_T”.	
3	-	Error/status of subordinate block “RD_LOC_T”.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

If the processes shall run automatically depending on the change between day and night, the function of an astronomical clock is necessary. Examples would be switching street lights on and off or opening and closing blinds.

If these processes should be executed with a time delay, i.e. before or after sunrise or sunset, an offset is required.

Note An exact execution of the function requires guaranteeing, that the system time and the local time of the SIMATIC controller have been 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 sunrise and sunset and are output at the outputs “sunrise” and “sunset”. If the system time of the SIMATIC controller is between these values, the “daytime” output is set to the value “TRUE”.

Note

Since the times for sunrise and sunset change on a daily basis, it might be possible that output “daytime” “stays” permanently on “TRUE” or “FALSE” over a longer period of time:

- at respectively large offset values
- at a location beyond the polar circle

The input of the coordinates can be given in the format “DMS” (with PLC data type “LGF_typeAstroDMS”) or “degree.decimal”.

The FORMAT which is ACTIVE, is defined with the formal parameter “modeDMS” (see Table 3-1).

The input of the coordinate values is checked for valid values. For invalid values, a respective error code is output at “status” (see Table 3-1).

If an invalid coordinate value is pending at a formal parameter, and this formal parameter was enabled via “modeDMS”, the outputs “sunrise” and “sunset” are set to the value DTL#1970-01-01-00:00:00.

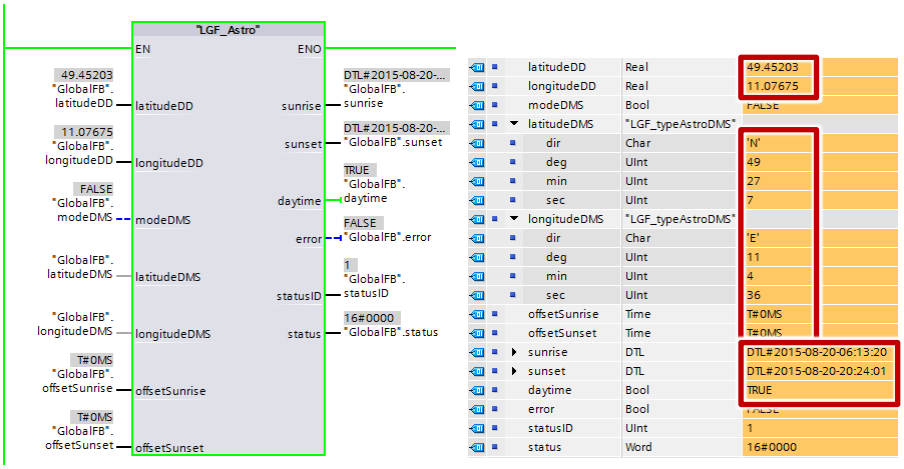
Example

The following example illustrates the functioning of the block.

Table 3-4: Geographical coordinates for Nürnberg-Moorenbrunn, date and system time

Longitude:	+ 11.07675°	or	E 11° 4' 36"
Latitude:	+ 49.45203°	or	N 49° 27' 7"
Date:	20.08.2015	Local time:	15:15:44

Figure 3-4: FB LGF_Astro, monitoring the block online with the parameters as well as the actual parameters via the watch table



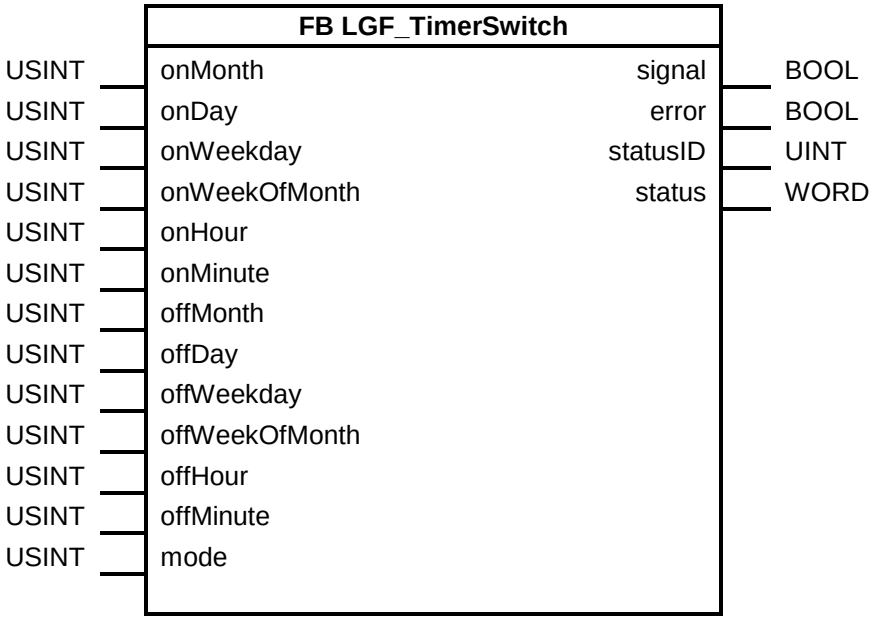
3.1.2 FB LGF_TimerSwitch

Short description

This block is a time switch. It is possible to specify switching times as daily, weekly, monthly, annual, and week of the month.

Block

Figure 3-5: FB LGF_TimerSwitch



Input parameters

Table 3-5: Input parameters

Parameter	Data type	Description
onMonth	USINT	Month, in which the signal shall be set.
onDay	USINT	Day, at which the signal shall be set.
onWeekday	USINT	Day of the week, at which the signal shall be set.
onHour	USINT	Hour, at which the signal shall be set.
onMinute	USINT	Minute, at which the signal shall be set.
offMonth	USINT	Month, in which the signal shall be reset.
offDay	USINT	Day, at which the signal shall be reset.
offWeekday	USINT	Day of the week, at which the signal shall be reset.
offHour	USINT	Hour, at which the signal shall be reset.
offMinute	USINT	Minute, at which the signal shall be reset.
mode	USINT	Specifying the mode (see function principle).

Output parameters

Table 3-6: Output parameters

Parameter	Data type	Description
signal	BOOL	Output signal
error	Bool	0: no errors 1: error in block, “statusID” outputs error source, “status” outputs error code
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-7: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	No valid current value was transferred to the “mode” input.	Only the values “1”, “2”, “3”, “4” and “52” are permitted.
2	-	Error/status of subordinate block “RD_LOC_T”.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block offers various types of time switches, which are defined in the “mode” parameter:

- Daily time switch (mode = 1)
- Weekly time switch (mode = 2)
- Monthly time switch (mode = 3)
- Annual time switch (mode = 4)

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

Table 3-8

Mode	Required formal parameters
Daily time switch (mode = 1)	• onHour / offHour • onMinute / offMinute
Weekly time switch (mode = 2)	• onWeekday / offWeekday • onHour / offHour • onMinute / offMinute

Mode	Required formal parameters
Monthly time switch (mode = 3)	• onDay / offDay • onHour / offHour • onMinute / offMinute
Annual time switch (mode = 4)	• onMonth / offMonth • onDay / offDay • onHour / offHour • onMinute / offMinute

If the set start time of the current local time corresponds to the controller, output “signal” is set to “TRUE”. If the set switch-off time of the current time corresponds to the, output “signal” is reset.

3.2 Counter operations

3.2.1 FC LGF_CountFallnDWord

Short description

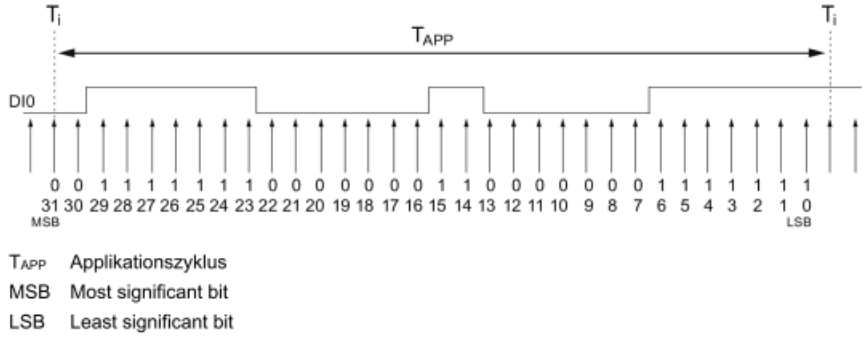
This block analyzes a DWORD type variable and detects falling edges. The block outputs the number of falling edges in the DWORD.

Application example

Abstract from the manual of technology module TM Timer DIDQ 16x24V

The Oversampling function is used by the technology module to detect the status of the respective digital input for each application cycle (for example, OB61) at 32 points in time at regular intervals. The 32 states are returned together in the feedback interface as a 32-bit value.

Figure 3-6: Example for oversampling of DO0 at Timer DIDQ 16x24V



T_{APP}	Applikationszyklus
MSB	Most significant bit
LSB	Least significant bit

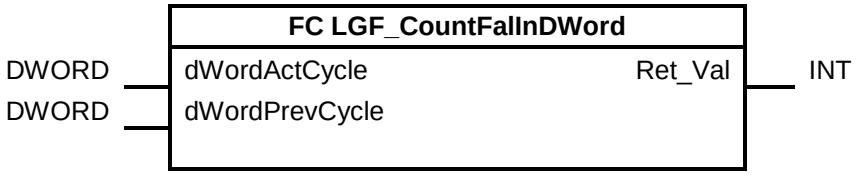
The block LGF_CountFallnDWord counts in that case all falling edges.

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

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

Block

Figure 3-7: FC LGF_CountFallnDWord



Input parameters

Table 3-9: Input parameters

Parameter	Data type	Description
dWordActCycle	DWORD	Double word in which the falling edges are counted
dWordPrevCycle	DWORD	Double word from the previous cycle

Output parameters

Table 3-10: Output parameters

Parameter	Data type	Description
Ret_Val	INT	Number of falling edges in the double word

Mode of Operation

In a DWord data type variable, the block counts the falling edges (1-0 transitions) from left to right. Output “Ret_Val” outputs the number of falling edges here.

In order for the falling edges at variable limits to be detected, the “dWordPrevCycle” input must be connected with the input of the variable of the previous cycle.

Example

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

Within this signal, the falling edges shall be continuously counted and output. In order for the falling edges at the variable limits to be detected - as in this example - the “dWordPrevCycle” input must be interconnected with the previous scanning.

Table 3-11: Example

DWORD previous cycle (dWordPrevCycle)	DWORD (dWord)
1001_0000_0001_1010_1001_0000_0001_1011	**0**01**0**_1**0**1**0**_0001_1111_**0**1**0**0_0011_1**0**00_01**0**1

Number of falling edges (“Ret_Val”): 8

3.2.2 FC LGF_CountRisInDWord

Short description

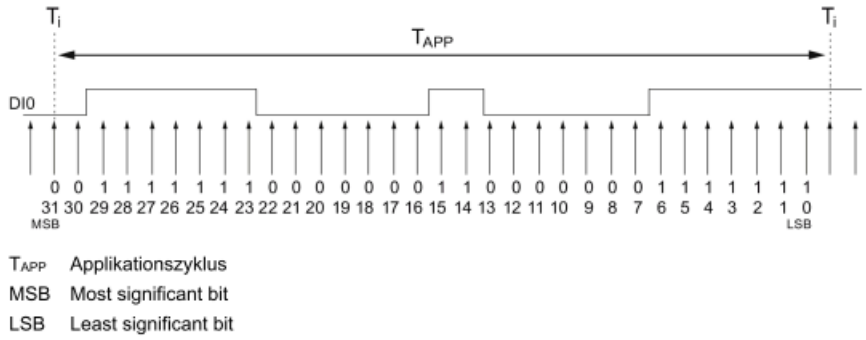
This block analyzes a DWORD type variable and detects rising edges. The block outputs the number of rising edges in the DWORD.

Application example

Abstract from the manual of technology module TM Timer DIDQ 16x24V

The Oversampling function is used by the technology module to detect the status of the respective digital input for each application cycle (for example, OB61) at 32 points in time at regular intervals. The 32 states are returned together in the feedback interface as a 32-bit value.

Figure 3-8: Example for oversampling of DO0 at Timer DIDQ 16x24V



T_{APP} Applikationszyklus
MSB Most significant bit
LSB Least significant bit

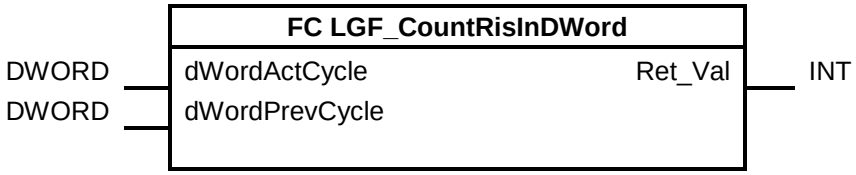
The block LGF_CountRisInDWord counts in that case all rising edges.

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

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

Block

Figure 3-9: FC LGF_CountRisInDWord



Input parameters

Table 3-12: Input parameters

Parameter	Data type	Description
dWordActCycle	DWORD	Double word in which the rising edge is counted
dWordPrevCycle	DWORD	Double word from the previous cycle

Output parameters

Table 3-13: Output parameters

Parameter	Data type	Description
Ret_Val	INT	Number of rising edges in the double word

Mode of Operation

In a DWord data type variable, the block counts the rising edges (0-1 transitions) from left to right. Output “Ret_Val” outputs the number of rising edges here.

In order for the rising edges at variable limits to be detected, the “dWordPrevCycle” input must be switched with the variable of the previous cycle.

Example

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

Within this signal, the rising edges shall be continuously counted and output. In order for the rising edges at the variable limits to be detected - as in this example - the “dWordPrevCycle” input must be interconnected with the previous scanning.

Table 3-14: Example

DWORD previous cycle (dWordPrevCycle)	DWORD (dWord)
1001_0000_0001_1010_1001_0000_0001_1010	**1**0**1**0_**1**0**1**0_000**1**_1111_0**1**00_00**1**1_1000_0**1**0**1**

Number of rising edges (“Ret_Val”): 9

3.3 Comperator operations

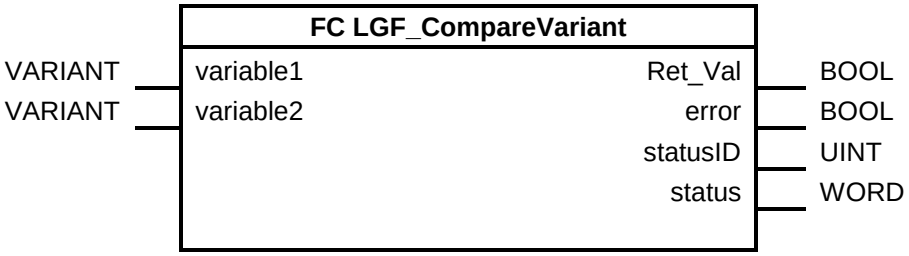
3.3.1 FC LGF_CompareVariant

Short description

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

Block

Figure 3-10: FC LGF_CompareVariant



Input parameters

Table 3-15: Input parameters

Parameter	Data type	Description
variable1	VARIANT	Comparator variable with any data type
variable2	VARIANT	Comparator variable with any data type

Output parameters

Table 3-16: Output parameters

Parameter	Data type	Description
Ret_Val	BOOL	0: data types of the comparator variables differ. 1: data types of the comparator variables are equal.
error	Bool	0: no errors 1: error in block, “statusID” outputs error source, “status” outputs error code.
statusID	UINT	“statusID” outputs the ID of the block that reports the status. See following table.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-17: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-

Mode of Operation

This block compares two (structured) actual parameters and outputs whether they correspond to the value.

Note "ARRAY of BOOL" type variables cannot be checked for equality by means of the function, since the used instruction "CountOfElements" also counts the filling elements.
(e.g. for an ARRAY[0..1] of BOOL, 8 is returned)

3.4 Math operations

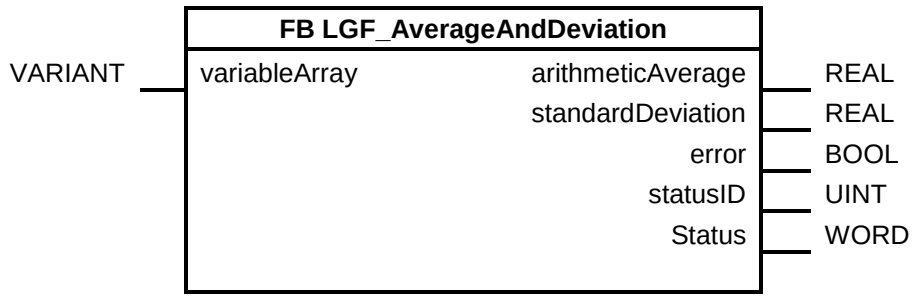
3.4.1 FB LGF_AverageAndDeviation

Short description

From a series of numbers, this block determines the arithmetic mean and the standard deviation.

Block

Figure 3-11: FB LGF_AverageAndDeviation



Input parameters

Table 3-18: Input parameters

Parameter	Data type	Description
variableArray	VARIANT	Number series to be calculated with.

Output parameters

Table 3-19: Output parameters

Parameter	Data type	Description
arithmeticAverage	REAL	Arithmetic mean
standardDeviation	REAL	Standard deviation
error	Bool	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-20: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	At input “variableArray”, the actual parameter is not an array.	-
1	16#8201	The data type of the array elements is not supported (see function principle).	-
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

With the input “variableArray”, a random-size array is connected. After a data type request in the block, the arithmetic mean is calculated from the values. Subsequently, the standard deviation is calculated and both are output.

Note Note: Only the data types Int, UInt, DInt, UDInt, USInt, SInt and Real are supported.

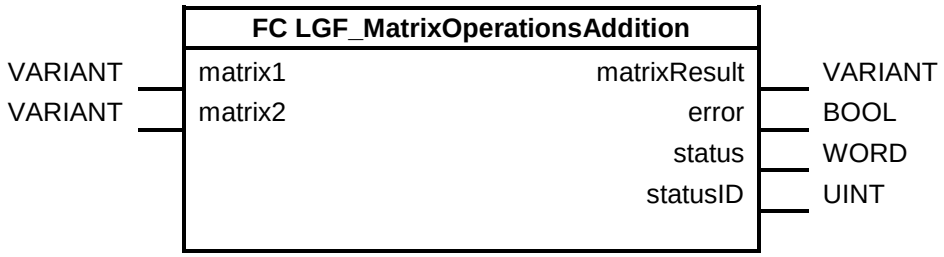
3.4.2 FC LGF_MatrixAddition

Short description

This block adds two equally sized “REAL” type matrixes.

Block

Figure 3-12: FC LGF_MatrixAddition



Input parameters

Table 3-21: Input parameters

Parameter	Data type	Description
matrix1	VARIANT	First summand (matrix)
matrix2	VARIANT	Second summand (matrix)

Output parameters

Table 3-22: Output parameters

Parameter	Data type	Description
matrixResult	VARIANT	Total (matrix)
error	Bool	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-23: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	All input and output matrixes need to have the same number of columns and rows.

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block adds two matrixes of variable size. The individual fields of the two incoming matrixes are read, added and then output in the “matrixResult” matrix.

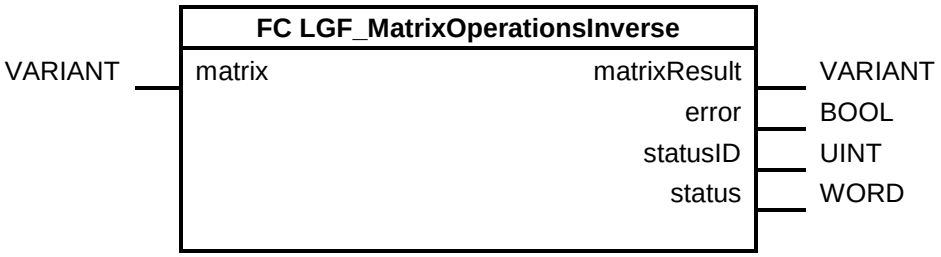
Note Please note, that all input and output matrixes must have the same number of columns and rows.

3.4.3 FC LGF_MatrixInverse

This block inverts a “REAL” type matrix.

Block

Figure 3-13: FC LGF_MatrixOperationsInverse



Input parameters

Table 3-24: Input parameters

Parameter	Data type	Description
matrix	VARIANT	Inverting matrix

Output parameters

Table 3-25: Output parameters

Parameter	Data type	Description
matrixResult	VARIANT	Resulting matrix
error	Bool	0: no errors 1: error in block, “statusID” outputs error source, “status” outputs error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-26: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	Formal parameter at input “matrix” has the wrong size.	Matrix must be square.
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	All input and output matrixes need to have the same number of columns and rows.

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block inverts a matrix of any size according to the Shipley-Coleman procedure.

Note Please note that the matrix must be square. This means, the number of rows must be equal to the number of columns.

The output matrix must be equally large as the input matrix.

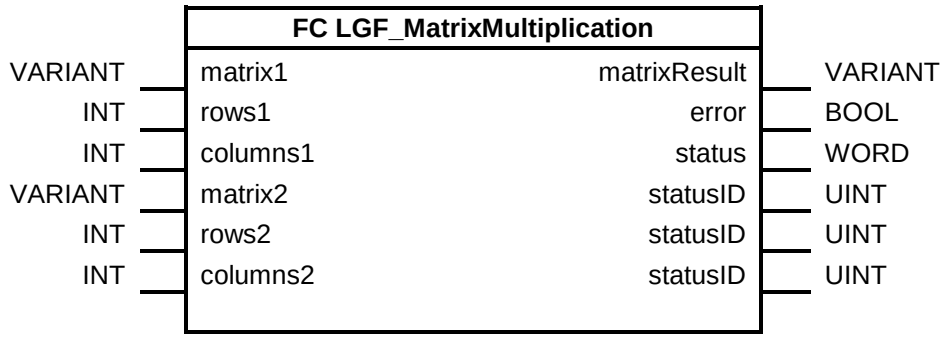
3.4.4 FC LGF_MatrixMultiplication

Short description

This block multiplies two “REAL” type matrixes.

Block

Figure 3-14: FC LGF_MatrixMultiplication



Input parameters

Table 3-27: Input parameters

Parameter	Data type	Description
matrix1	VARIANT	First factor: matrix to be multiplied
rows1	INT	Number of rows of the first matrix.
columns1	INT	Number of columns of the first matrix.
matrix2	VARIANT	Second factor: matrix to be multiplied
rows1	INT	Number of rows of the second matrix.
columns1	INT	Number of columns of the second matrix.

Output parameters

Table 3-28: Output parameters

Parameter	Data type	Description
matrixResult	VARIANT	Product: Resulting matrix
error	Bool	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-29: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	The number of columns of the first matrix does not correspond to the number of rows of the second matrix.	-
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	Check the matrixes and the respective rows and columns.

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

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

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

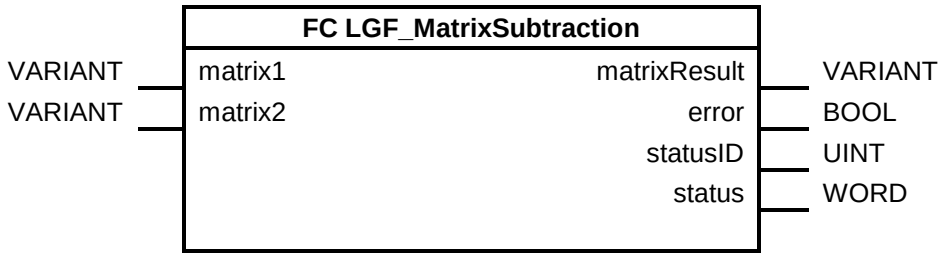
The size of the output matrix results from the largest number of rows and columns of the input matrixes.

3.4.5 FC LGF_MatrixSubtraction

This block subtracts a “REAL” type matrix from another matrix.

Block

Figure 3-15: FC LGF_MatrixSubtraction



Input parameters

Table 3-30: Input parameters

Parameter	Data type	Description
matrix1	VARIANT	Minuend: “matrix2” is subtracted from this matrix.
matrix2	VARIANT	Subtrahend: This matrix is subtracted from “matrix1”.

Output parameters

Table 3-31: Output parameters

Parameter	Data type	Description
matrixResult	VARIANT	Difference: Resulting matrix
error	Bool	0: no errors 1: error in block, “statusID” outputs error source, “status” outputs error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-32: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	All input and output matrixes need to have the same number of columns and rows.

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block subtracts two matrixes of variable size. The individual fields of the two matrixes are read, subtracted and then output in the “matrixResult” matrix.

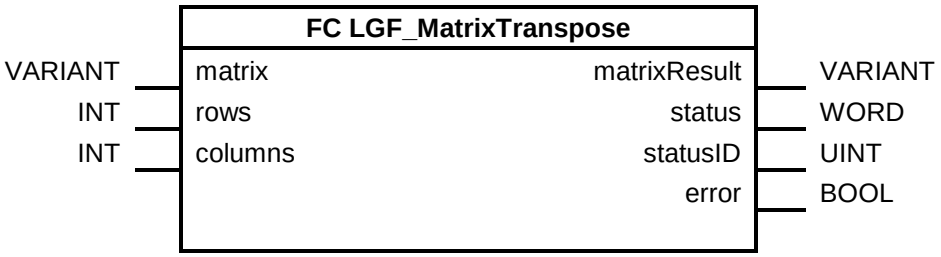
Note Please note, that all input and output matrixes must have the same number of columns and rows.

3.4.6 FC LGF_MatrixTranspose

This block transposes a matrix.

Block

Figure 3-16: FC LGF_MatrixTranspose



Input parameters

Table 3-33: Input parameters

Parameter	Data type	Description
matrix	VARIANT	Matrix to be transposed.
rows	INT	Number of rows of the matrix to be transposed.
columns	INT	Number of columns of the matrix to be transposed.

Output parameters

Table 3-34: Output parameters

Parameter	Data type	Description
matrixResult	VARIANT	Resulting matrix.
error	Bool	0: no errors 1: error in block, “statusID” outputs error source, “status” outputs error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-35: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	All input and output matrixes need to have the same number of columns and rows.

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block transposes a matrix of any size. The rows and columns are swapped (mirrored)

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

3.4.7 FB LGF_MinMaxHistory

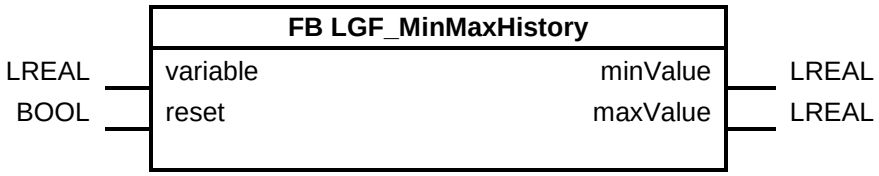
Short description

For each call, this block reads a value of a variable and outputs the maximal and minimal value read since the first call.

The evaluation can be reset if necessary. The block supports the data type LREAL.

Block

Figure 3-17: FB LGF_MinMaxHistory



Input parameters

Table 3-36: Input parameters

Parameter	Data type	Description
variable	LREAL	Variable whose value is checked for minimum and maximum.
reset	BOOL	The block is reset and the evaluation starts from the beginning.

Output parameters

Table 3-37: Output parameters

Parameter	Data type	Description
minValue	LREAL	Minimum value since the first call or since activating the “reset” input.
maxValue	LREAL	Maximum value since the first call or since activating the “reset” input.

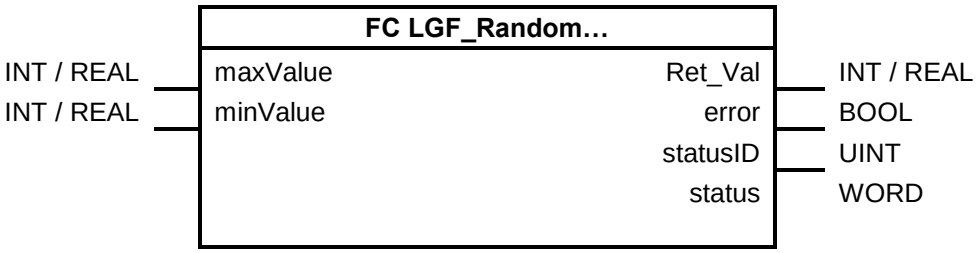
3.4.8 FC LGF_RandomINT / FC LGF_RandomReal

Short description

This block generates a “random” value between a defined maximum and minimum value. The random number has the data type INT / REAL.

Block

Figure 3-18: FC LGF_RandomINT / FC LGF_RandomReal



Input parameters

Table 3-38: Input parameters

Parameter	Data type	Description
maxValue	INT / REAL	Defines the upper limit value of the random number.
minValue	INT / REAL	Defines the lower limit value of the random number.

Output parameters

Table 3-39: Output parameters

Parameter	Data type	Description
Ret_Val	INT	Random number
error	Bool	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-40: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	“minValue” is larger than “maxValue”.	-
2	-	Error/status of subordinate block “RD_SYS_T”.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block generates random values between the given “minValue” value and “maxValue” value. This random value is output via the “Ret_Val”.

Background information

In order for the generated value to turn out as random as possible, two important program sections are run through. Each call of the block starts with a new standard value. The Nano seconds of the current time are used as start value. To prevent the output value from being constantly rising, the ten thousands, thousands, hundreds, tens and single units are swapped, partly also exchanged. This process depends on the current second of the system time of the controller.

3.4.9 FB LGF_SearchMinMax

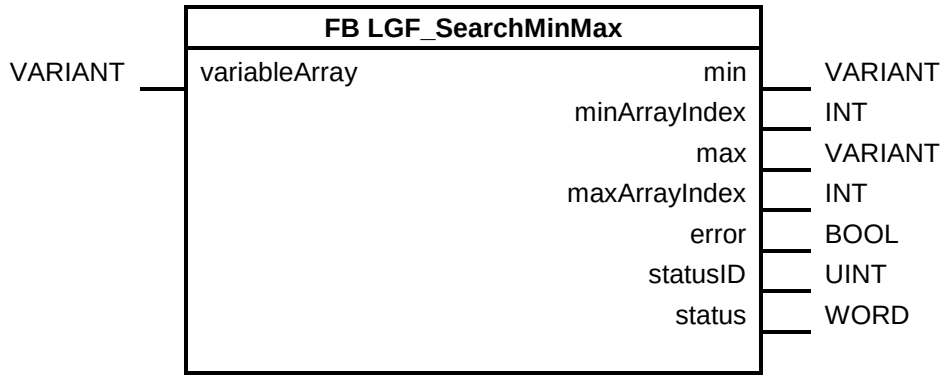
Short description

In an array, this block searches the maximal and minimal value as well as 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

Figure 3-19: FB LGF_SearchMinMax



Input parameters

Table 3-41: Input parameters

Parameter	Data type	Description
variableArray	VARIANT	Array, in whose fields the maximum and minimum is searched.

Output parameters

Table 3-42: Output parameters

Parameter	Data type	Description
minValue	VARIANT	Smallest detected value.
minArrayIndex	INT	Start index of the arrays plus minArrayIndex yields the array index of the smallest value. The index starts with 0.
maxValue	VARIANT	Largest detected value.
maxArrayIndex	INT	Start index of the arrays plus maxArrayIndex yields the array index of the largest value. The index starts with 0.
error	Bool	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-43: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	“minValue” is larger than “maxValue”.	-
1	16#8201	The data type of the array elements is not supported.	Only the data types Int, UInt, DInt, UDInt, USInt, SInt and Real are supported.
1	16#8202	The elements of the array do not have the same data type as the ouptuts “minValue” and “maxValue”.	
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

An array of any size is connected via the “variableArray” input. After a data type request in the block, the elements are copied successively into a variable of the appropriate type and compared. The smallest and largest value, as well as their respective index in the array are output.

Note For several equal minimum and maximum values, the index of the first minimum or maximum value is output.

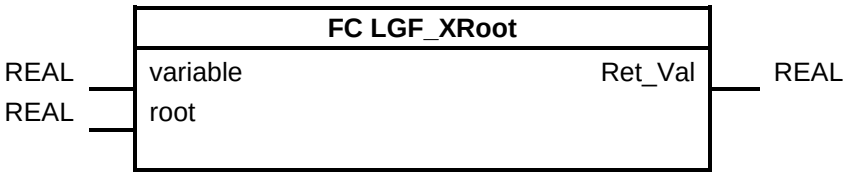
3.4.10 FC LGF_XRoot

Short description

This block calculates the x[th] root of a numerical variable.

Block

Figure 3-20: FC LGF_Random



Input parameters

Table 3-44: Input parameters

Parameter	Data type	Description
variable	REAL	Variable to be calculated from the root.
root	REAL	Root (e.g. 3 as 3rd root)

Output parameters

Table 3-45: Output parameters

Parameter	Data type	Description
Ret_Val	REAL	Output of the result

Mode of Operation

The block calculates the n[th] root of a number. To implement this function, the following formula is expanded.

$$number = e^{log_e(number)}$$

This results in:

$$Ret\_Val = \sqrt[root]{number} = number^{\frac{1}{root}} = (e^{log_e(number)})^{\frac{1}{root}} = e^{\ln(number) * \frac{1}{root}}$$

In STEP 7 (TIA Portal) the function “EXP” corresponds to $e^{(\ldots)}$ and die function “LN” to ln(…).

This results in the following formula:

$$Ret\_Val = EXP((1/root) * \mathrm{LN}(number))$$

3.5 Data handling

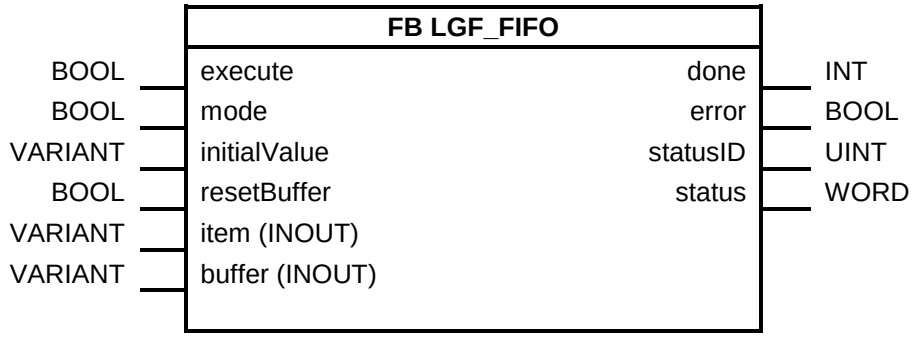
3.5.1 FB LGF_FIFO

Short description

This block stores incoming jobs/data and outputs the oldest not yet processed job.

Block

Figure 3-21: FB LGF_FIFO



Input parameters

Table 3-46: Input parameters

Parameter	Data type	Description
execute	BOOL	Requesting a run.
mode	BOOL	Selecting the mode. TRUE: writing the value from “item” into “buffer” FALSE: reading the value from “buffer” and output at “item”
initialValue	VARIANT	Value for initializing the ring buffer (mostly: 0)
resetBuffer	BOOL	Emptying and initializing the ring buffer.
item	VARIANT	Value returned from the ring buffer or to be written to the ring buffer.
buffer	VARIANT	Ring buffer (array of …)

Output parameters

Table 3-47: Output parameters

Parameter	Data type	Description
done	INT	Error information
error	Bool	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	Word	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-48: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8001	The ring buffer is empty.	-
1	16#8002	The ring buffer is full.	-
1	16#8200	An array is not pending at the “buffer” input.	-
1	16#8201	The data type of the InOut parameter “item” does not correspond to the data type of the array elements of the “buffer” input.	-
1	16#8202	The data type of the “initialValue” input does not correspond to the data type of InOut parameter “item”.	-
1	16#8601	The “nextEmptyItemIndex” variable does not lie within the array limits.	-
1	16#8602	The “firstItemIndex” variable does not lie within the array limits.	-
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT”.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

In order to start the block, a positive edge must be pending at the “execute” input.

To request the next read job to be processed, value “FALSE” must be pending at the “mode” input. If this is the case, the next processed job/data is output at InOut parameter “item”, and this field is replaced by the value at the “InitialValue” parameter in the ring buffer.

To save a new write job in the ring buffer, value “TRUE” must be pending at the “mode” input. If this is the case, the value pending at the InOut parameter “item” or job will is stored in the ring buffer at the next free location.

If value “TRUE” is pending at input “resetBuffer” during a rung, all fields in the ring buffer are reset to the value given at input “InitialValue2. Subsequently, the ring buffer can be filled up with jobs/data again.

3.5.2 FB LGF_ShellSortInt / LGF_ShellSortUInt / ShellSortReal

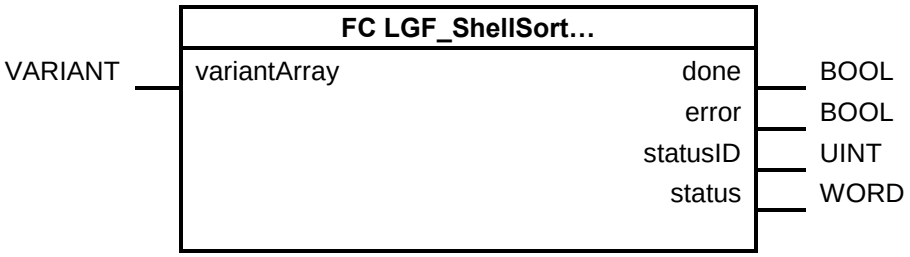
Short description

This block sorts an array with any number of elements (max. 1000) rising and is available for three different data types:

- Array of type “Int”: LGF_ShellSortInt
- Array of type “UInt”: LGF_ShellSortUInt
- Array of type “Real”: LGF_ShellSortReal

Block

Figure 3-22: FC LGF_ShellSort…



Input parameters

Table 3-49: Input parameters

Parameter	Data type	Description
variantArray	VARIANT	Array to be sorted.

Output parameters

Table 3-50: Output parameters

Parameter	Data type	Description
done	INT	Error information
error	BOOL	0: no errors 1: error in block, “statusID” outputs the error source, “status” outputs the error code.
statusID	UINT	“statusID” outputs the ID of the block, that reports the status. See table below.
status	WORD	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-51: Status/error codes

statusID	Status	Meaning	Remedy / notes
1	16#7000	Initial value	-
1	16#0000	No faults	-
1	16#8200	Actual parameter at input “variantArray” only has one element.	At input “variantArray” you interconnect an array with at least two elements.
1	16#8201	Actual parameter at input “variantArray” is not an array.	
1	16#8202	Actual parameter at input “variantArray” does not have the suitable data type.	Interconnect an array with the correct data type at the “variantArray” input. LGF_ShellSortInt: Array of type Int LGF_ShellSortUInt: Array of type UInt LGF_ShellSortReal: Array of type Real
1	16#8203	Actual parameter at the “variantArray” input has too many elements.	By default, you can have array sorted with up to 1000 elements.
2	-	Error/status of subordinate block “MOVE_BLK_VARIANT” when reading the array.	-
3	-	Error/status of subordinate block “MOVE_BLK_VARIANT” when writing the array.	-

Note If “statusID” > 1, all values of output “status” origin directly from the called instructions (see table output parameter). In this case you fetch the information from the TIA Portal Online help on the respective instructions.

Mode of Operation

The block sorts according to the Shell sorting procedure. Please note that the execution time of the block strongly depends on how many elements the array to be sorted has. The following overview shows some measured values of the block depending on the number of array elements.

Table 3-52: 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 not divided between several PLC cycles. The execution time hence directly affects the PLC cycle time. Please note this behavior in your project regarding the used controller and, if necessary, adjust the monitoring time to the controller.

3.6 Converter operations

3.6.1 FC LGF_BCDToGray

Short description

This block turns a BCD-encoded value into a Gray-encoded value.

Block

Figure 3-23: FC LGF_BCDToGray

Input parameters

Table 3-53: Input parameters

Parameter	Data type	Description
variableBCD	DWORD	BCD-encoded value.

Output parameters

Table 3-54: Output parameters

Parameter	Data type	Description
Ret_Val	DWORD	Gray-encoded value.

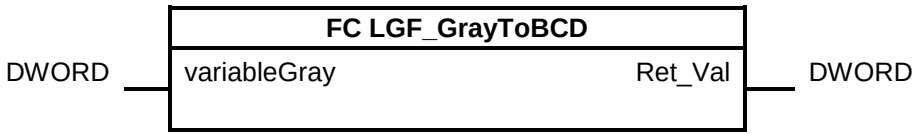
3.6.2 FC LGF_GrayToBCD

Short description

This block turns a Gray-encoded value into a BCD-encoded value.

Block

Figure 3-24: FC LGF_GrayToBCD



Input parameters

Table 3-55: Input parameters

Parameter	Data type	Description
variableGray	DWORD	Gray-encoded value.

Output parameters

Table 3-56: Output parameters

Parameter	Data type	Description
Ret_Val	DWORD	BCD-encoded value.

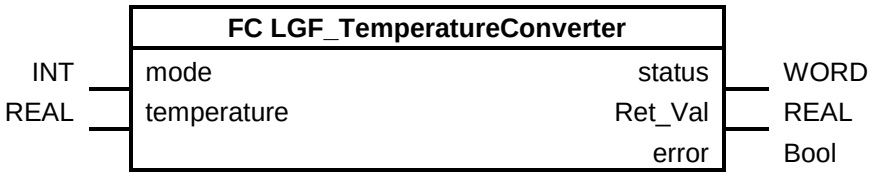
3.6.3 FC LGF_TemperatureConverter

Short description

This block turns °Celsius into °Fahrenheit or Kelvin as well as °Fahrenheit into Kelvin and vice versa.

Block

Figure 3-25: FC LGF_TemperatureConverter



Input parameters

Table 3-57: Input parameters

Parameter	Data type	Description
mode	INT	Mode 1. °Celsius into °Fahrenheit 2. °Fahrenheit into °Celsius 3. °Celsius into Kelvin 4. Kelvin into °Celsius 5. °Fahrenheit into Kelvin 6. Kelvin into °Fahrenheit
temperature	REAL	Temperature to be converted

Output parameters

Table 3-58: Output parameters

Parameter	Data type	Description
Ret_Val	INT	Converted temperature
error	BOOL	0: no errors 1: error in the block
status	WORD	“status” outputs the status/error code (see table below).

Status and error displays

Table 3-59: Status/error codes

Status	Meaning	Remedy / notes
16#7000	Initial value	-
16#0000	No faults	-
16#8200	No correct mode at input “mode”.	See description of the input parameters

3.7 Signal generators

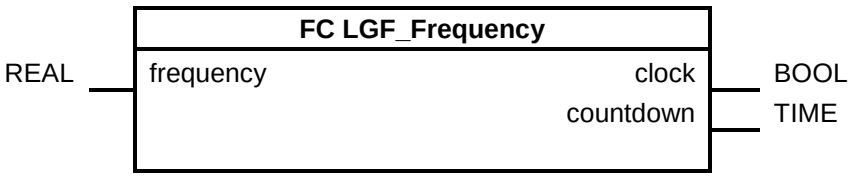
3.7.1 FB LGF_Frequency

Short description

The block generates a signal which, depending on a defined frequency, changes between the values “0” and “1”.

Block

Figure 3-26: FC LGF_Impulse



Input parameters

Table 3-60: Input parameters

Parameter	Data type	Description
frequency	REAL	Cycle frequency in Hz

Output parameters

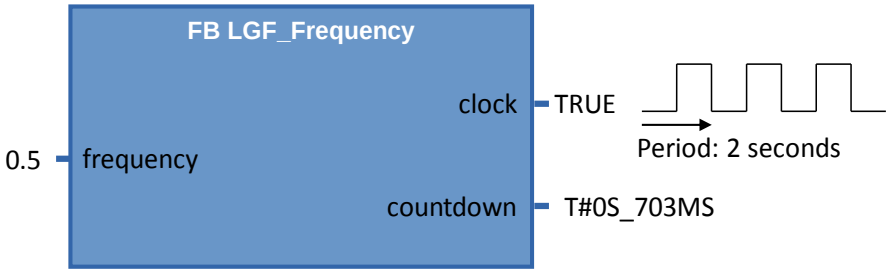
Table 3-61: Output parameters

Parameter	Data type	Description
clock	BOOL	Output changes with defined frequency
countdown	TIME	Remaining time of the current state of “clock”

Mode of Operation

The “clock” output is a Boolean value that toggles in the desired frequency. The “countdown” output outputs the remaining time of the current state of “clock”. If the desired frequency is smaller or equal to “0.0”, then output “clock” = FALSE and “countdown” = “0.0”.

Figure 3-27



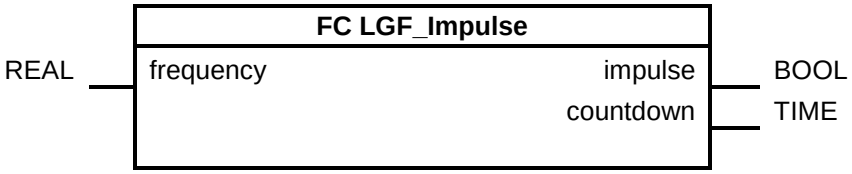
3.7.2 FC LGF_Impulse

Short description

This block generates pulses in a given frequency. The pulse is always pending for one (control) cycle.

Block

Figure 3-28: FC LGF_Impulse



Input parameters

Table 3-62: Input parameters

Parameter	Data type	Description
frequency	REAL	Clock speed

Output parameters

Table 3-63: Output parameters

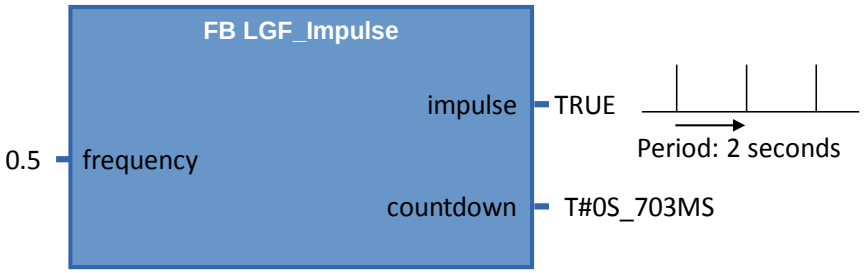
Parameter	Data type	Description
impulse	BOOL	Signal with pulses
countdown	TIME	Time until the next pulse

Mode of Operation

The block generates pulses at output “impulse” with the frequency “frequency”.

The frequency is realized via the “LGF_Frequency” block. If a negative edge is pending at output “clock” of the “LGF_Frequency” block, a pulse is generated at output “impulse”.

Figure 3-29



Note In order for the block to work, block “LGF_Frequency” must additionally be adopted into the STEP7 project.

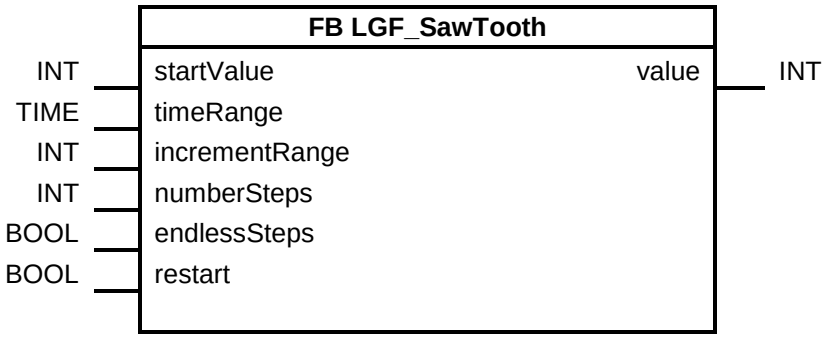
3.7.3 FB LGF_SawTooth

Short description

The block generates a sawtooth-shaped signal curve. Each sawtooth consists of a defined number of steps (increments).

Block

Figure 3-30: FB LGF_SawTooth



Input parameters

Table 3-64: Input parameters

Parameter	Data type	Description
startValue	INT	Start value at which the signal starts.
timeRange	TIME	Time after which output parameter “value” is incremented
incrementRange	INT	Size of the jump from one increment to the next.
numberSteps	INT	Number of increments per sawtooth. (In the case of an endless sawtooth-shaped signal, this specification is not necessary)
endlessSteps	BOOL	Detail of whether an endless sawtooth-shaped signal shall be generated
restart	BOOL	Sawtooth starts again at start value “startValue”.

Note Please note, that changes at the input parameters only become effective with “restart”.

Output parameters

Table 3-65: Output parameters

Parameter	Data type	Description
value	INT	Current value of the sawtooth-shaped signal.

Mode of Operation

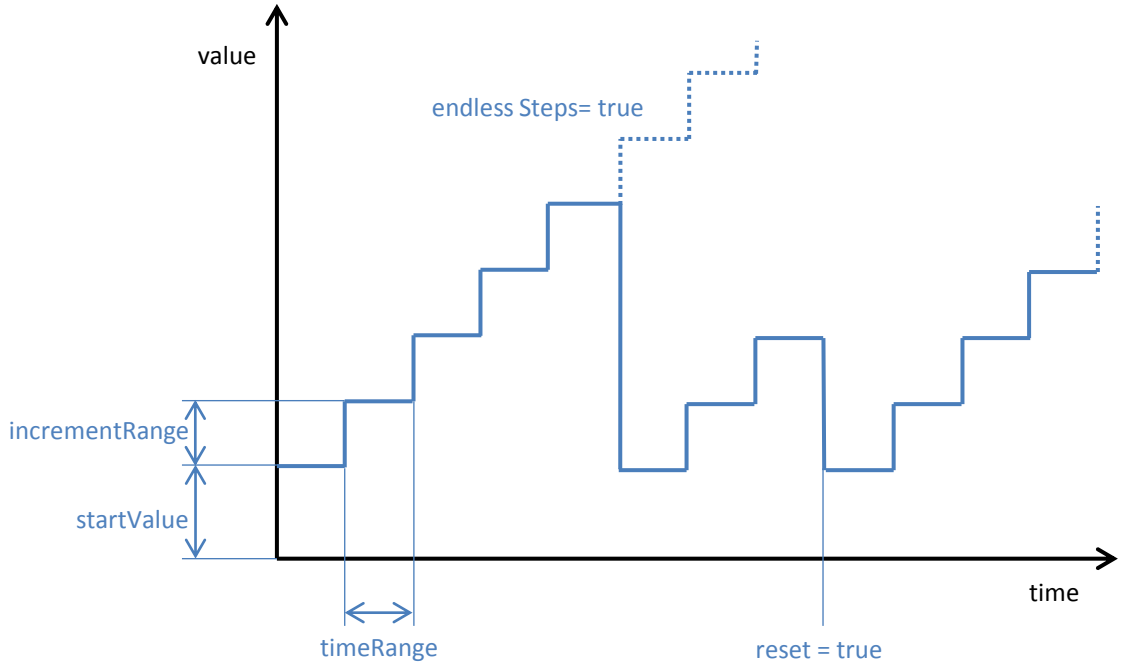
The block calculates the values for a sawtooth-shaped signal curve output at output parameter “value”. The signal starts with start value “startValue” and is added with value “increment” after each run of time interval “timeRange”. The value can also be negative.

If the “endlessSteps” variable is set to “FALSE”, the number of add processes is counted. If they exceed the value “numberSteps”, the output value “value” is reset to the start value. A new sawtooth starts. If the “endlessSteps” variable is set to “TRUE”, value “increment” is continuously added.

Note The duration of a sawtooth at “endlessSteps” on “FALSE” is calculated as follows:

Duration = “timeRange” * (“numberSteps” + 1)

Figure 3-31: Signal course of output “value”



4 Links & Literature

Table 4-1

	Topic	Title
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Download page of the entry	https://support.industry.siemens.com/cs/ww/en/view/109479728

5 History

5.1 Explanation on versioning the library

Library and library elements are maintained according to the following table:

Table 1: Definition of the version

V	1.	2.	3
	Non-compatible change	Compatible change	Error correction
	• Reducing the interfaces • Changing the interface • Incompatible expansion of the functionality	• Expansion of the interfaces • Compatible expansion of the functionality	• Bugfix

Example on versioning

Table 5-2: Example for changing the version

Library	FB1	FB2	FC1	Comment
1.0.0	1.0.0	1.0.0	1.0.0	Enable first version
1.0.1	1.0.1	1.0.0	1.0.0	Error handling 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	Compatible expansion at FB1
1.2.0	1.2.0	1.0.1	1.0.0	Compatible expansion at FB1
2.0.0	2.0.0	1.0.1	2.0.0	New functionality at FB1 and FC1
2.0.1	2.0.0	1.0.2	2.0.0	Error handling of FB2

5.2 Change log

Table 5-3

Version	Date	Modifications
V1.0.0	09/2015	First version
V1.0.1	10/2015	LGF_Astro V1.0.1 • T_ADD instruction is replaced with “+”.