EcoStruxure Machine Expert Twin

Logic Configurator

User Guide

EIO0000005052.00
06/2024

www.se.com

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this document are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owner.

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

Document Scope

This document describes the **Logic Configurator** feature of the EcoStruxure Machine Expert Twin and the functions it provides. For further information, refer to the separate documents provided in the EcoStruxure Machine Expert Twin online help.

Validity Note

This document has been created for the release of EcoStruxure Machine Expert Twin V2.1.

Related Documents

Document title	Reference
Cybersecurity Best Practices	CS-Best-Practices-2019-340
Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment	EIO0000004242
EcoStruxure Machine Expert Twin Getting Started - User Guide	EIO0000005022 (ENG)
EcoStruxure Machine Expert Twin How to Create Device Catalogs - User Guide	EIO0000005034 (ENG)

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Product Related Information

⚠WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

For reasons of Internet security, for those devices that have a native Ethernet connection, TCP/IP forwarding is disabled by default. Therefore, you must manually enable TCP/IP forwarding. However, doing so may expose your network to possible cyberattacks if you do not take additional measures to protect your enterprise. In addition, you may be subject to laws and regulations concerning cybersecurity.

⚠WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT NETWORK INTRUSION

- Observe and respect any and all pertinent national, regional and local cybersecurity and/or personal data laws and regulations when enabling TCP/IP forwarding on an industrial network.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Consult the Schneider Electric Cybersecurity Best Practices for additional information.

EcoStruxure Machine Expert Twin is a simulation and emulation software suite to create digital models of real machines to start the virtual design, virtual pre-commissioning, and to support co-development before building the machine – thus enabling parallel engineering of mechanical, electrical and controls work assignments.

The simulation, emulation and machine visualization functions of EcoStruxure Machine Expert Twin are intended to support you in developing your application and its configuration by simulating the behavior of the various machine or process components. These functions are not intended to substitute for, but to complement the processes of risk assessment, risk evaluation, validation, and commissioning as well as any ancillary processes, tasks, and obligations according to the applicable regulations and standards such as ISO/EN 13849 and IEC 62061. The product, though powerful, does not, nor can it, simulate every aspect of the application and its environment.

⚠WARNING

INSUFFICIENT TEST COVERAGE

- Do not use EcoStruxure Machine Expert Twin as the sole means for risk assessment, risk evaluation, validation, and commissioning as well as any ancillary processes, tasks, and obligations according to the applicable regulations and standards such as, but not limited to, ISO/EN 13849 and IEC 62061.
- Verify and validate your results on the intended equipment before placing your machine or process into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Based on the system configuration and operation, a hazard and risk analysis must be conducted for the system (for example, according to ISO 12100 or ISO 13849-1) independent of the work with EcoStruxure Machine Expert Twin. The results of this analysis must be considered when designing the machine, and subsequently applying safety-related equipment and safety-related functions. The results of your analysis may deviate from any digital models of physical machines that you may create. For example, additional safety components may be required. In principle, the results from the hazard and risk analysis have priority.

⚠WARNING

NON-CONFORMANCE TO SAFETY FUNCTION REQUIREMENTS

- Specify the requirements and/or measures to be implemented in the risk analysis you perform.
- Verify that your safety-related application complies to applicable safety regulations and standards.
- Make certain that appropriate procedures and measures (according to applicable sector standards) have been established to help avoid hazardous situations when operating the machine.
- Use appropriate safety interlocks where personnel and/or equipment hazards exist.
- Validate the overall safety-related function and thoroughly test the application.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Catalogs contain important data, parameters and operational aspects of the devices defined within. This information is subject to change over time for a variety of reasons. Therefore, it is necessary to maintain the relationship between the models you create and the catalogs you have used to do so. Version mismatches of catalogs may cause your models to operate in ways that are incongruent with the equipment they represent and may lead to errors in design and operation.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Impose a system of file name conventions that readily indicate the version of the catalogs you use and models you create.
- Create documentation that records catalog and model versions, as well as firmware versions of the equipment used in your models.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

In addition, changes to your underlying application (logic, data address, functions, I/O configurations, device types and configuration, etc.) can have serious impact on the models you have created.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Update your models every time you modify your application or change the physical hardware configuration.
- Verify that objects you have created in your models are coherent with the modifications and/or changes you have made to your application and that they are associated with the correct variables.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

It is also important to connect to the correct automation logic/motion controller in a networked, multi-controller environment.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION Verify that you have connected to the intended automation controller. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

It is important to manage the amount of data that is transmitted between your automation logic/motion controller and EcoStruxure Machine Expert Twin. Large amounts of data, or data that is not contiguous in the controller memory may impact performance of EcoStruxure Machine Expert Twin, the controller or both.

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in the information contained herein, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2023	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2020	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2021	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.

Standard	Description
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2021	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive* (*2006/42/EC*) and *ISO 12100:2010*.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Introduction

What’s in This Part

General Information

What's in This Chapter

General Information on the Logic Configurator

General Information

The **Logic Configurator** is accessible as a view within the EcoStruxure Machine Expert Twin user interface. It allows you to configure logical relations between the elements of the scene in a graphical editor. You can use the **Logic Configurator** to create flow simulations by adding logical operations and connecting them with elements of the scene. The **Logic Configurator** provides element inputs and outputs and allows you to configure the relations between them without a controller.

Example of a Flow Simulation

For a visual illustration of a flow simulation created with the **Logic Configurator**, refer to the video sequence provided in the online help of this user guide.

Functions

What's in This Part

Functions of the Logic Configurator

What's in This Chapter

Creating Logical Relations

Overview

To create logical relations between the elements of the scene, perform the following steps:

1. Create blocks in the **Logic Configurator**.
2. Create connections between the blocks.

Creating Blocks

Right-click anywhere in the **Logic Configurator** view to access the contextual menu with commands for creating blocks to establish logical relations between the scene elements.

- **Create Register Node(s)**

  For further information, refer to Creating Multiple Blocks for the Same Assembly, page 18.
- **Logic**

  For details, refer to **Logic** Blocks, page 21.
- **Comparison**

  For details, refer to **Comparison** Blocks, page 22.
- **Mathematics**

  For details, refer to **Mathematics** Blocks, page 23.
- **Boolean Delay**

  For details, refer to Boolean **Delay** Block, page 24.
- **Boolean Display**

  For details, refer to Boolean **Display** Block, page 24.
- **Counter**

  For details, refer to **Counter** Block, page 25.
- **Number Delay**

  For details, refer to Number **Delay** Block, page 25.
- **Number Display**

  For details, refer to Number **Display** Block, page 25.
- **Random**

  For details, refer to **Random** Block, page 26.
- **Timer**

  For details, refer to **Timer** Block, page 26.

For details about accessing these commands, refer to Good Practices for Working with the **Logic Configurator**, page 17.

Creating Connections

To establish a relation between blocks, follow these steps:

1. Click an output of one assembly block.
2. Hold down the left mouse button and draw a connecting line to the input of another assembly block.

The connection between two blocks is indicated by a solid line. Connections can only be established between inputs and outputs of the same type, see Types of Inputs and Outputs, page 19.

To remove a connection, follow these steps:

1. Right-click the connection in the **Logic Configurator**.
2. Click **Delete** from the contextual menu.

   NOTE: Each input can be connected to only one signal, whereas each output signal can have multiple connections.

Good Practices

Good Practices for Working with the Logic Configurator

This chapter provides tips and tricks for working with the **Logic Configurator**.

Accessing the Commands

You can access the contextual menu commands by typing the first letter of the command.

Examples:

Right-click in the **Logic Configurator** view to open the contextual menu and type:

- `t` to create a **Timer** block.
- `c` + `c` to create a **Counter** block because this is the second command in the list starting with `c`.
- `m` + `n` to create a **Number** block. It serves as a shortcut to use the command **Mathematics > Number**.

Creating Multiple Blocks for the Same Assembly

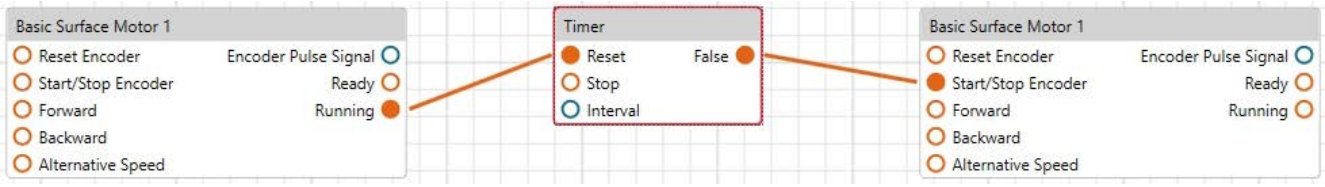
To create a clear arrangement of blocks and connecting lines for the same assembly or motor, you can use multiple blocks. To achieve this:

1. Select the assembly or motor in the scene.
2. Right-click anywhere in the **Logic Configurator** view and click **Create Register Node(s)** several times until the desired number of blocks is created.

   NOTE: If you create multiple instances of the same assembly or motor and you establish multiple connections to the same input, the block uses the last updated input value as the default input.

Example:

To establish clear visibility of connecting lines when representing an output signal located on the right-hand side of the assembly block (**Basic Surface Motor 1** in this example) connecting to another block located on the left-hand side (**Timer** in this example), it is a good practice to create an additional instance of the block (**Basic Surface Motor 1** in this example) and position it on the left-hand side (of the **Timer** block in this example).

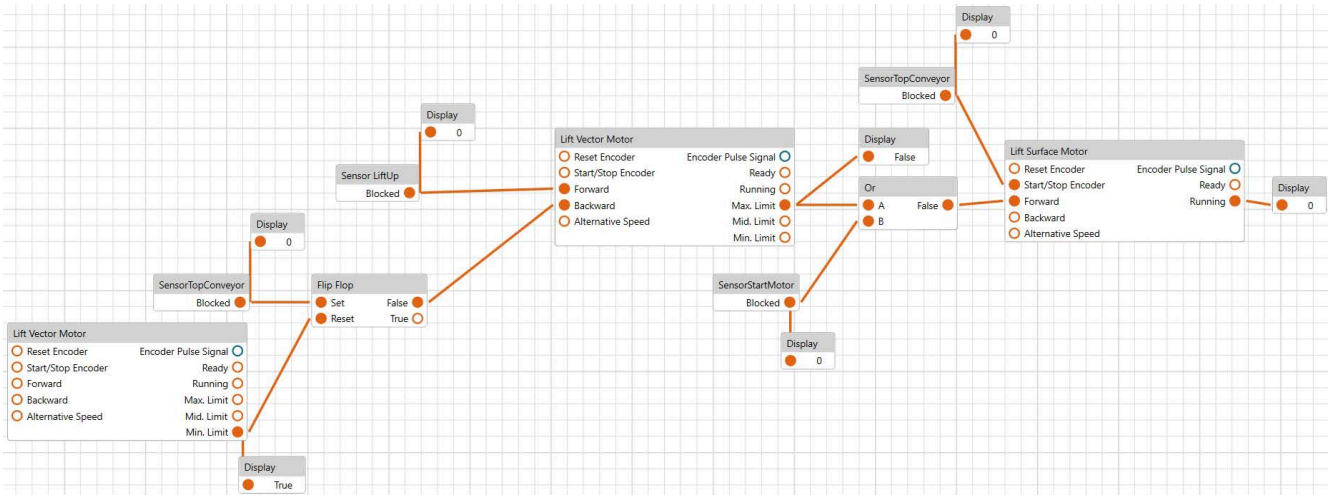


Displaying Boolean and Numerical Values

The values of Boolean and numerical inputs and outputs are, by default, not displayed for all blocks in the **Logic Configurator**. To make these values visible, create the following blocks and connect them to the inputs or outputs:

- **Boolean Display** blocks to visualize Boolean outputs.
- **Number Display** blocks to visualize numerical outputs.

For more information, refer to Types of Inputs and Outputs, page 19.



Managing Blocks

To create blocks, right-click in the **Logic Configurator** and select a command from the contextual menu to create a specific block.

To remove a block, right-click the block and use the **Delete** command from the contextual menu or select the block and press the **Delete** key on your keyboard.

Configuring Blocks

You can configure a block in two different ways:

- Select the block in the **Logic Configurator** and configure the corresponding parameters in the **Properties** view on the right-hand side of the EcoStruxure Machine Expert Twin window.
- Create other blocks, for example, **Mathematics** blocks, and establish relations between the blocks within the **Logic Configurator**.

Types of Inputs and Outputs

In the blocks of the **Logic Configurator**, inputs and outputs of distinct types are depicted using different colors. You can only establish connections between inputs and outputs of the same type.

Value type	Color of connecting lines and points
Boolean values	orange
Numerical values	blue

NOTE: String values are not supported by the **Logic Configurator**.

Functions of the Blocks

What's in This Chapter

Functions of the Blocks

Overview

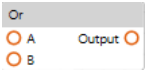
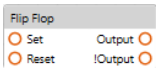
The blocks you can insert in the **Logic Configurator** provide different functions. This chapter provides further information on available blocks.

Logic Blocks

Logic blocks are logical operators that perform operations using Boolean input values.

NOTE: If either input *A* or *B* is not connected, it is automatically interpreted as FALSE.

The following **Logic** blocks are available:

Block	Description
Not 	Performs a logical negation of the operand *A*: *A* = TRUE => *Output* = FALSE *A* = FALSE => *Output* = TRUE
And 	Performs a logical AND of two operands *A* and *B*: *A* = TRUE, *B* = TRUE => *Output* = TRUE Other combinations of *A* and *B* => *Output* = FALSE
Nand 	Performs a negated logical AND of two operands *A* and *B*: *A* = TRUE, *B* = TRUE => *Output* = FALSE Other combinations of *A* and *B* => *Output* = TRUE
Nor 	Performs a negated logical OR of two operands *A* and *B*: *A* = FALSE, *B* = FALSE => *Output* = TRUE Other combinations of *A* and *B* => *Output* = FALSE
Or 	Performs a logical OR of two operands *A* and *B*: *A* = FALSE, *B* = FALSE => *Output* = FALSE Other combinations of *A* and *B* => *Output* = TRUE
XNor 	Performs a negated exclusive OR of two operands *A* and *B*: *A* = FALSE, *B* = TRUE => *Output* = FALSE *A* = TRUE, *B* = FALSE => *Output* = FALSE Other combinations of *A* and *B* => *Output* = TRUE
Xor 	Performs an exclusive OR of two operands *A* and *B*: *A* = FALSE, *B* = TRUE => *Output* = TRUE *A* = TRUE, *B* = FALSE => *Output* = TRUE Other combinations of *A* and *B* => *Output* = FALSE
Flip Flop 	Provides two outputs: *Output* and negated *!Output*. A *Set* state is retained until a *Reset* input becomes TRUE. See the **Flip Flop** input and output signals table hereafter.

Flip Flop input and output signals:

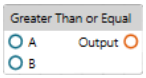
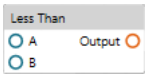
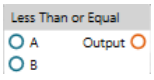
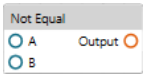
Input signals		Output signals		Result
Set	***Reset***	***Output***	***!Output***	
FALSE	FALSE	*Output*	*!Output*	If both input signals are FALSE, the output signals do not change.
FALSE	TRUE	FALSE	TRUE	A reset is performed: *Output* = FALSE *!Output* = TRUE The state is retained.
TRUE	FALSE	TRUE	FALSE	A set is performed: *Output* = TRUE *!Output* = FALSE The state is retained.
TRUE	TRUE	FALSE	FALSE	If both input signals are TRUE, an invalid state is detected and both output signals are set to FALSE.

Comparison Blocks

Comparison blocks are logical operators that perform mathematical comparison operations using numerical input values. The output is presented as a Boolean value.

NOTE: If either input *A* or *B* is not connected, it is automatically interpreted as 0.

The following **Comparison** blocks are available:

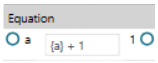
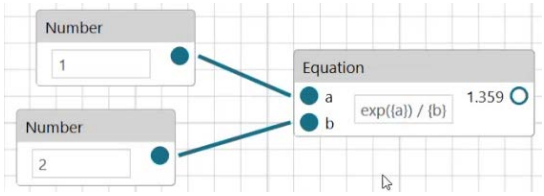
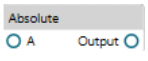
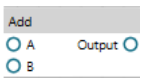
Block	Description
Equal 	If *A* is equal to *B*, the output is TRUE. *A* = *B* => *Output* = TRUE
Greater Than 	If *A* is greater than *B*, the output is TRUE. *A* > *B* => *Output* = TRUE
Greater Than or Equal 	If *A* is greater than or equal to *B*, the output is TRUE. *A* >= *B* => *Output* = TRUE
Less Than 	If *A* is less than *B*, the output is TRUE. *A* < *B* => *Output* = TRUE
Less Than or Equal 	If *A* is less than or equal to *B*, the output is TRUE. *A* <= *B* => *Output* = TRUE
Not Equal 	If *A* is not equal to *B*, the output is TRUE. *A* <> *B* => *Output* = TRUE

Mathematics Blocks

Mathematics blocks are logical operators that perform mathematical operations using numerical input values and presenting numerical output values.

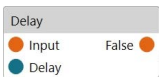
NOTE: If either input *A* or *B* is not connected, it is automatically interpreted as 0.

The following **Comparison** blocks are available:

Block	Description
Equation 	The **Mathematics > Equation** block allows you to enter functions in the text box of the block. These functions act on the input variables as shown in the following example: `exp({a}) / {b}`  To create a variable, enter text surrounded by curly brackets. For example, `{MyVariable}` creates the variable *MyVariable*. Input variables are case-sensitive. `{PI}` representing the constant value π = 3.14159 is supported. The following operators are supported: • `+` (addition) • `-` (subtraction) • `*` (multiplication) • `/` (division) • `%` (remainder) • `^` (power) • `!` (factorial) • `sqrt` (square root) • `sin` (sine) • `asin` (arcsine) • `cos` (cosine) • `acos` (arccosine) • `tan` (tangent) • `atan` (arctangent) • `rad` (convert degrees to radians) • `deg` (convert radians to degrees) • `ceil` (ceiling) • `floor` (floor) • `round` (rounding) • `exp` (e raised to power) • `abs` (absolute)
Number 	The **Number** block allows you to enter a numerical input.
Absolute 	The absolute value `| x |` of a numerical value `x` is the non-negative value of `x` without regard to its sign.
Add 	The numerical input values are added. *Output = A + B*

Block	Description
Ceiling 	The input real number `x` is rounded to the smallest integer that is greater than or equal to `x` (also known as the least or smallest integer function). Examples: Input *A* = `9.51` => *Output* = `10` Input *A* = `9.49` => *Output* = `10`
Divide 	The numerical input value *A* is divided by value *B*. *Output* = *A* / *B*
Floor 	The input real number `x` is rounded to the greatest integer that is less than or equal to `x` (also known as the greatest integer function). Examples: Input *A* = `9.51` => *Output* = `9` Input *A* = `9.49` => *Output* = `9`
Modulo 	The remainder or signed remainder of a division is returned. *Output* = *A* Mod *B*
Multiply 	The numerical input values are multiplied. *Output* = *A* * *B*
Round 	The real number `x` is rounded to the nearest integer. Examples: Input *A* = `9.51` => *Output* = `10` Input *A* = `9.49` => *Output* = `9`
Subtract 	The numerical input value *B* is subtracted from the numerical input value *A*. *Output* = *A* - *B*

Boolean Delay Block



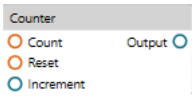
Run the **Boolean Delay** command to add a **Delay** block. It allows you to add a delay time (in s) defined with the *Delay* input after the input is received before the Boolean output is set.

If the *Delay* input is not connected, it is by default set to 1 s.

Boolean Display Block

Run the **Boolean Display** command to add a **Display** block. It allows you to debug and visualize the values of Boolean outputs.

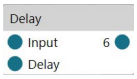
Counter Block



Inputs	Description
Count	When a rising edge is detected at the *Count* input, the *Output* value increases by the numerical value provided as *Increment* input.
Reset	When a rising edge is detected at the *Reset* input, the *Output* value is reset to 0.
Increment	Numerical input value by which the *Output* value is incremented when a rising edge is detected at the *Count* input. If the *Increment* input is not connected, it is interpreted as 1 by default.

Outputs	Description
Output	Numerical value that serves as a counter output.

Number Delay Block

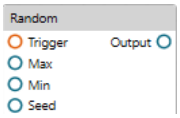


Run the **Number Delay** command to add a **Delay** block. It allows you to add a delay time (in s) defined with the *Delay* input, which sets the numerical output.

Number Display Block

Run the **Number Display** command to add a **Display** block. It allows you to debug and visualize the values of numerical outputs.

Random Block



Inputs	Description
Trigger	When a rising edge is detected at the *Trigger* input, a random numerical *Output* value is generated based on the rules defined using the *Max*, *Min*, *Seed* inputs.
Max	This numerical input value defines the maximum threshold of the value range for the random output value. This input value must be available to generate a random numerical *Output* value of type integer (**Int**, see **Properties** of a **Random** block).
Min	This numerical input value defines the minimum threshold of the value range for the random output value. This input value must be available to generate a random numerical *Output* value of type integer (**Int**, see **Properties** of a **Random** block).
Seed	This numerical input value defines the initial value for the random output value.

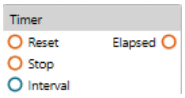
Outputs	Description
Output	Generates a random numerical value.

Properties of a Random block:

The following parameter is available in the **Properties** view when a **Random** block is selected:

Parameter	Description
RandomMode	Select the type of numerical values that are generated as *Output* values: • **Int**: Output values of type integer are generated, for example, `2`. To use this option, a *Max* and a *Min* input value must be available. • **Double**: Output values of type double are generated, for example, `0.596`. Default value: **Double**

Timer Block



Inputs	Description
Reset	When a rising edge is detected at the *Reset* input, the timer is reset.
Stop	When a rising edge is detected at the *Stop* input, the timer is stopped (without being reset).
Interval	This numerical input value in seconds defines the time interval for the timer. Default value: 1 s This input value is also available in the **Properties** view when the **Timer** block is selected. You can edit the value in the **Properties** view or you can connect a numerical **Number** block to this input. Within this time interval, the **Elapsed** output switches from FALSE to TRUE as configured with the **Duty Cycle** parameter in the **Properties** view.

NOTE: Start the physical simulation for the **Timer** to become effective.

Outputs	Description
Elapsed	The Boolean output switches from FALSE to TRUE within the *Interval* time as configured with the **Duty Cycle** parameter in the **Properties** view. By default, the output is 0.5 s FALSE and 0.5 s TRUE with *Interval* = 1 s and **Duty Cycle** = 50%.

Properties of a Timer block:

The following parameters are available in the **Properties** view when a **Timer** block is selected:

Parameter	Description
Interval	Time interval for the timer. Default value: 1 s Edit this value or connect a numerical **Number** block to the *Interval* input of the **Timer** block to define a time interval within which the **Elapsed** output switches from FALSE to TRUE as configured with the **Duty Cycle** parameter.
Duty Cycle	Configures a percentage of the **Interval** time. After this percentage of the **Interval** time has elapsed, the **Elapsed** output switches from FALSE to TRUE. Default value: 50%
Auto Reset	By default, this option is selected and the timer is automatically reset and restarted. Clear the check box to start the timer only upon a rising edge at the *Reset* input.

Example:

Interval = 10 s

Duty Cycle = 80%

With these settings, the *Elapsed* output is set to FALSE for 2 seconds and to TRUE for 8 seconds within the interval of 10 seconds.

Glossary

D

digital twin:

A digital twin refers to a virtual representation or digital replica of a physical object, system, or process. It is a digital counterpart that simulates the behavior, characteristics, and performance of its physical counterpart in real-time or historical contexts. The concept of a digital twin allows for the integration of the physical and digital worlds, enabling organizations to monitor, analyze, and optimize the performance of their assets or processes.

EcoStruxure Machine Expert Twin provides features for visualization, simulation, and emulation of machines and automation lines throughout the complete lifecycle.

E

emulation:

Based on the *ISO 24765-2017 International Standard - Systems and software engineering--Vocabulary*, emulation is defined as the use of a data processing system to imitate another data processing system, so that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated system.

M

Model view: In EcoStruxure Machine Expert Twin, the **Model** view provides the graphical representation of the scene.

P

physical simulation: The physical simulation is a software library that is designed to simulate and model physical systems in a computer-generated environment. It is used to create realistic and dynamic animations and simulations of objects, environments, and interactions between them. In EcoStruxure Machine Expert Twin the physical simulation uses mathematical algorithms to simulate physical phenomena, such as gravity, friction, and collision detection.

project: An EcoStruxure Machine Expert Twin project file is saved with the extension **.experior*. It contains the information about assemblies, connections, loads, settings.

S

scene: In the EcoStruxure Machine Expert Twin context, a scene is a representation of a set of assemblies interacting with loads.

simulation:

Based on the *ISO 24765-2017 International Standard - Systems and software engineering--Vocabulary*, simulation describes two concepts:

- A model that behaves or operates like a given system when provided a set of controlled inputs.
- The use of a data processing system to represent selected behavioral characteristics of a physical or abstract system.

In the context of this manual, the term simulation is used whenever it is referred to modeling physical systems in EcoStruxure Machine Expert Twin.

STEP: (STandard for the Exchange of Product model data) This ISO 10303 standard specifies a standard file format for 3-D models and allows for product data exchange across different platforms.

U

URDF: (unified robotics description format) A special type of eXtensible Markup Language (XML) file that includes the physical description of a robot and contains information on the mechanical structure, joints, 3-D modelling graphics, motors and colliders. URDF files are provided by numerous robotic manufacturers for download. EcoStruxure Machine Expert Twin allows importing URDF files for integrating third-party robots into a project without manual programming.

Index

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

EIO0000005052.00