COPA-DATA
do it your way

zenon tutorial

Command input

v.7.00

Contents

1. Welcome to COPA-DATA help

GENERAL HELP

If you miss any information in this help chapter or have any suggestions for additions, please feel free to contact us via e-mail: documentation@copadata.com (mailto:documentation@copadata.com).

PROJECT SUPPORT

If you have concrete questions relating to your project, please feel free to contact the support team via e-mail: support@copadata.com (mailto:support@copadata.com)

LICENSES AND MODULES

If you realize that you need additional licenses or modules, please feel free to contact the sales team via e-mail: sales@copadata.com (mailto:sales@copadata.com)

2. Command input

Command input serves primarily for the secured switching of data points in energy technology. 'Secured' means that there is a check whether the switching operation is allowed, according to the configured interlocking condition and the dynamically updated topology (no relation to the zenon network topology). The configuration of the topology is performed via the ALC.

These conditions determine whether the switching action is allowed at the moment (No interlocking condition is active), whether it is forbidden (non-unlockable condition is active) or whether it can be performed after unlocking (unlockable interlocking is active).

A data point of the command input always consists of 2 data points. The response variable and the command variable. The action is performed on one of these data points.

User actions are performed via a context menu or with the screen Command. Specific control elements are provided for this. This screen also contains a procedure which allows unlocking, two-step execution and locking. The screen can be activated via screen element, function or context menu.

The synchronization of the execution of the command input is performed via an access object which is runtime-monitored and updated cyclically in the driver and which is associated to the response variable. The activation of this object is indicated by the status 'Select' at the response variable.

The access objects are created on demand and associated with the creator on the Runtime side. Managing these objects in the driver makes sure that there is only one active access object per response variable.

The command group can be exported and imported as well as exchanged and duplicated on the interlocking level. Command interlockings and general interlockings are stored in a common file (`verriegel.bin`/`verrieg.cmp`) and are also managed together. This is why it is not possible to have a general interlocking and a command group with the same name.

3. Definition of terms

3.1 Command screen

Screen of type 'Command input' which serves for the interaction with the user during command execution. Contains tailor-made functionality and screen elements for commands.

3.2 Command group

This type of interlocking contains a configurable number of actions with their static interlockings, as well as the formula variables available for the static interlockings.

3.3 Response variable

As the name indicates, the current or resulting value of a switching action ("command input action" called "action" from now on) is reported back via this data point. But it can also be the actual data point on which an action is executed.

Is replaced

3.4 Command variable

The actions 'single command', 'double command' and 'setpoint input' are performed on this data point. You can activate SBO (select before operate) for these actions.

Is replaced

3.5 Command input action

The action defines the type of command to be executed, with its parameters and the static command interlockings. This can be the writing of a value or a status.

3.6 Action variable

In order to have a consistent term for the data point which is manipulated by an action, this data point is called action variable. Actually, this can also be the response variable or command variable.

3.7 Add-on variable

The add-on variable is the data point which is associated with the command group and which is used for loading the screen of type 'Command'.

There must always be a command group configured for the add-on variable. The response variable and the action variable are then determined from the command group.

However, this does not mean that the variable must be used for the command group. If the used command group defines the response variable absolutely and if the action variables are defined absolutely, it is also possible to not use the add-on variable in the command group. Replacement does not work here because the data point cannot be resolved.

3.8 Condition variables

Data points that can be used for the conditions of the interlocking conditions in the formulas.

If a condition variable is used, it cannot be deleted.

If a condition variable, which is not the last defined one, is deleted, the following variables are moved forwards and the formulas are adjusted.

Is replaced

3.9 Interlocking condition

Any number of interlocking conditions can be defined for every action. Theses conditions are checked before execution of the action.

3.10 Internal interlocking conditions

These conditions are checked before action execution; the engineer cannot influence this.

e.g.: Select could not be activated or SBO was rejected.

3.11 Topological interlocking conditions

These conditions result from the current topological status. The topology is configured with ALC.

4. Power On

In this step by step example, we will configure and execute our first command input . During that, we will get to know a part of the functionality of the command input. After that we will take a closer look at special functions.

We create a new project in the Editor and include the simulator driver (if it does not exist already). We use this driver for simplicity reasons, although this will lead to some limitations regarding command input.

Please consider that you will need e.g. the IEC870 driver with an appropriate 60870 device for full support of all command input options. This driver supports COT (cause of transmisison) and SBO. This functionality is required for the watchdog timer and the SPO procedure.

> **Info**
>
> *The use of the included 'Project Wizard' is recommended. It helps you to create a basic project structure and saves you the trouble of manually defining process screens.*

4.1 Screen for command input

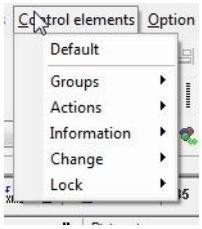
A separate screen type 'Command input' was created for the user interaction of the command input during Runtime. In this screen, the operator performs the activities necessary for command execution. This can be e.g. the unlocking of an active interlocking or the confirmaton of a two-step execution.

You can use specific control elements for this screen type, which allow all user actions necessary for command input and which visualize information about the status of the command input.

4.1.1 Create screen

Create a screen via the appropriate context menu.

Choose the type 'command input' for 'screen type'. By this, we turned the screen into a screen for command input. Next, we will add the standard control elements to the screen. We do this with the menu entry 'Add template' in the menu 'Control elements'

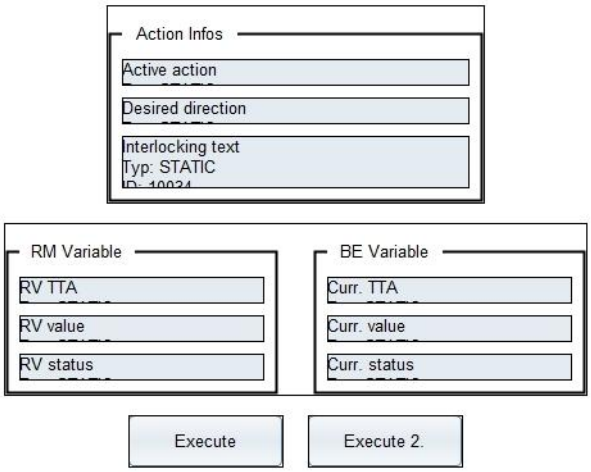


A number of control elements is added. We will look at the detailed functionality of the controls later. At the moment, this is not important.

We call the screen 'PowerOn'

> **Info**
>
> *We recommend to use a separate frame for this screen. This template should not fill the whole screen, but rather be treated like a pop up window.*

If there is not enough room for all 'Default' control elements on your selected frame, the following control elements are enough for this example:



4.1.2 REMA for the switching direction texts

The texts for the switching direction in the RT are determined via limits or REMA states. This makes it possible to use different texts for data points that has the same interlocking. The texts are shown in the context menu and in the Control **`Switching direction`**.

We configure a Multibinary Rema with the following properties:

Value	Limit value text
0	@off
1	@on
2	@diff
3	@dist
4	@none

All other settings of the states are default. We name the REMA 'Switching direction'.

4.1.3 Create data points

In order to test the command input, we need some data points. We will now create some.

Name	Drivers	Type	Offset
SW_Switch1_RM	Simulator	SPS Merker/Int	110
SW_Switch1_BE	Simulator	SPS Merker/Int	110
SW_Schalter1_CO	Simulator	SPS Merker/Int	114
SW_Switch2_RM	Simulator	SPS Merker/Int	112
SW_Switch2_BE	Simulator	SPS Merker/Int	112
SW_Schalter2_CO	Simulator	SPS Merker/Int	116

Because the Simulator driver does not know about the allocation of the command variable to the response variable, they are both configured on the same offset. This allows them to update each other, at least regarding the value.

The names are chosen in a way that allows the text _SchalterX to replace the replacement character * later on.

For the *RM and *BE data points, we allocate the REMA 'Switching direction'.

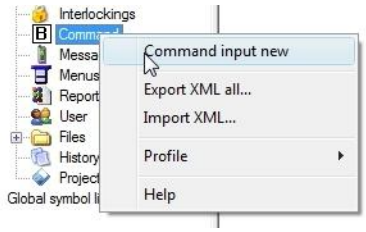
Attention

Do not forget to configure the control variable for the simulator driver with offset 0 or to set it to 0. Otherwise, the driver will run in simulation mode.

4.2 Creating a command input

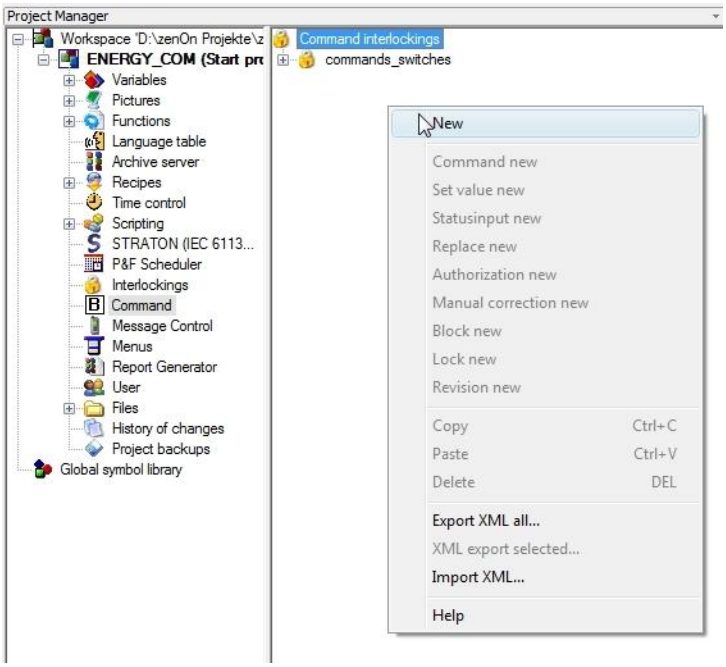
Now we create a simple command group to get to know the configuration possibilities of the command input.

Therefore, we can open the context menu in the project tree at the node 'Command input'



and create a new command group with **New command**.

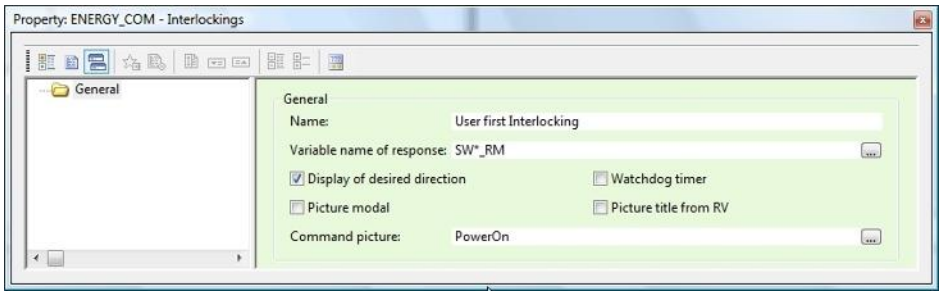
Alternatively, a command group can be created by choosing the project node in the window of the command group and then, via the context menu



and **New**.

After creating a new command input, it is inserted in the tree with the standard name 'interlocking X'. X is replaced by a number that identifies the name among all interlockings in the project, including the general interlockings.

Let us now have a look at the properties of the interlocking:



Perform the following changes of the default settings:

- Change name to the name in the screen.
- Change variable name or the response to the depicted text.
- Activate Set under construction.
- Activate Watchdog timer.
- Choose 'PowerOn' as command screen.

Explanation of the properties.

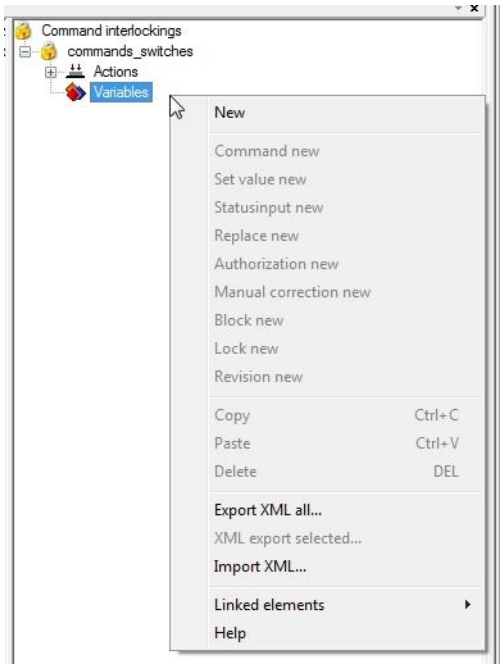
Name of the property	Description
Name	Name of the command group. Must be unique among all interlockings in the project. This name is used later with the process variable that uses this interlocking. The actual allocation is done with a unique consecutive numerical ID. The name is only used for GUI, Export and Import.
Variable name of response	This is the variable name or the mask for the replacement of the response variable. The placeholder for the replacement text is the character sequence ‚*' within a name. Only one placeholder can be used in a name. When entering the mask, it is important to take care that this name results in an existing variable name after replacement. If the variable that is used here (replaced or absolute) does not exist during compiling, the command group is not available in the RT. An according message announces this error during compiling.
Set under construction	If activated, status bit In progress (PROGRESS) is written at actions command and Manual correction. The value that the status bit is set to depends on the switching direction of the action. The status bit is set to 1 if: ▶ the Return state/switching direction of the action is ON or OFF. ▶ The response variable does not already have the value of the set switching direction. The status bit is set during the check of the interlockings and it remains on that value until the screen returns to step 1 or until it is closed. This also implies that the status remains while the watchdog timer and/or the edge delay is active. If the execution of an action is triggered by a context menu or if it is a one-step action, the status bit is set appropriately.

Laufzeitüberwachung	Determines process of the watchdog timer. Select from the drop-down list: ▸ none: no watchdog timer ▸ Response variable: The value of the response variable is used to determine if the command was successful. ▸ COT : The COT (Cause of Transmission) is used to determine if the operation was successful. For action '**Command**' the states for the cause of transmission (COT) are evaluated. If the COT procedure is not ended successfully within the timeout configured for the action, the status bit TIMEOUT is set. The monitoring is activated only if the value to be written does not equal the current value of the action variable. See also: Cause_of_Transmission_(COT)_Procedure
Screen modal	If this is active, the screen is displayed modally, independent of the setting 'Modal dialog' in the screen settings.
Screen title from RV	The identification of the response variable is shown in the screen title. This only happens when there a title was configured for the screen at the frame. Online language switchable.
Command screen	Name of the screen to be loaded if no action specific screen is defined and if the screen is not opened via the function 'Screen : Switch to'.

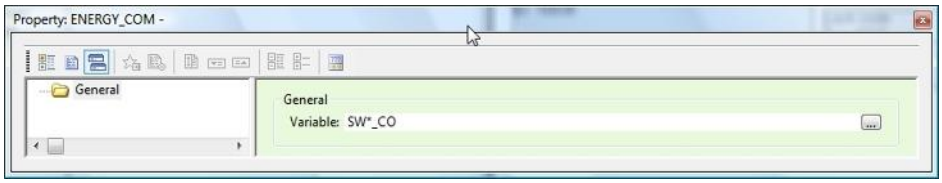
4.2.1 Defining condition variable

Any number of interlocking conditions can be defined for every action. These conditions allow for an additional restriction of the executability of the action. These conditions are defined with formulas, in which you can use the variables from the active projects. The variables available for the command group must be defined. The formula addresses these variables via the Index. The condition variable is automatically replaced if you use a ’*’ in the definition.

Select the node 'Variable' of the command group and, via the context menu,

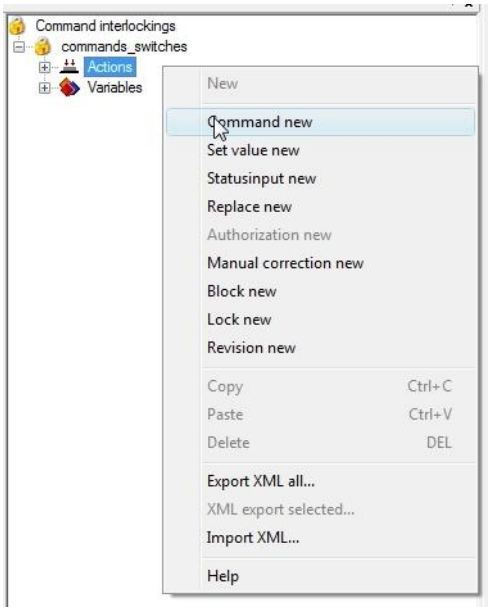


and **`New`**, add a new condition variable. Cancel the variable selection dialog. This results in the creation of an empty definition. Select this definition and enter the text `SW*_CO` for `Variable`. The settings for the condition variable are then as follows:



4.2.2 Create action

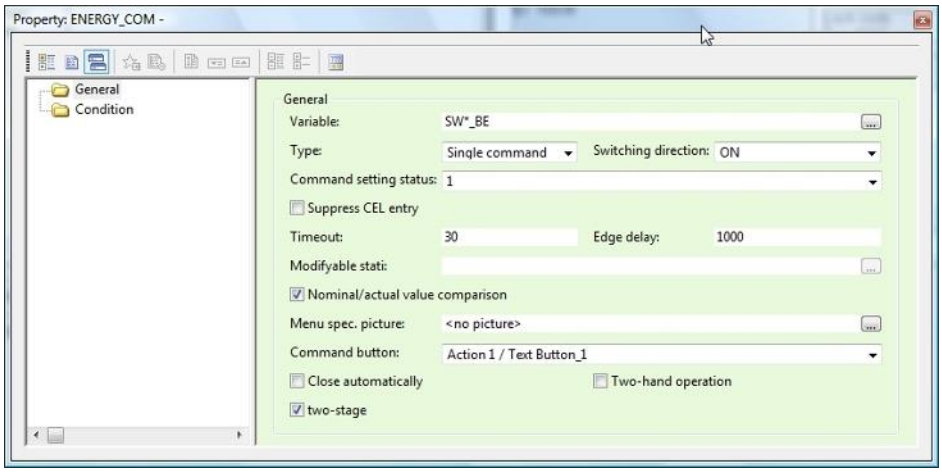
Actions are the switching commands that are possible for the command groups, with the action-specific configuration. By selecting the command group and activating the context menu



you can create the desired action. With **`Command new`**, we now create a new action 'Command'. Now we parameterize the action:

- Set Variable to SW*_BE
- Activate Nominal/actual value comparison
- Set Command button to Action 1 / Text Button_1

This is how the settings should look now:



Explanation of the properties.

Name of the property	Description
Variable	Data point, on which writing actions are performed. For some actions, this is the response variable. In this case, the field is locked. The placeholder for the replacement text is the character sequence ‚*' within a name. Only one placeholder can be used in a name. If the variable that is used here (replaced or absolute) does not exist during compiling, the action is not available in the RT. An according message announces this error during compiling. Initialization: No Allocation
Type	Determines the type of the command. Options are: single or double command. Double command: Writes the value of the setting 'Command setting status'. Single command: Like a double command, but after the edge delay, a rewrite with the value 0 is performed automatically. This rewrite is no longer runtime-monitored. Locked for all other actions. Initialization: Single command
Switching direction	Defines the expected value or the status of the response variable after action execution. Locked for the actions Block, Set value, Lock and Release. Initialization: Off
Command settings status	Defines the value that is written to the command variable during the action 'Command'. Locked for all actions except 'Single command' and 'Double command'. Initialization: 0
Suppress CEL entry	If this is active, no entry in the CEL will be made when executing an action. Initialization: Inactive
Timeout	Timeout used for watchdog timer. Only available for the actions Command and Setpoint input. Unit is seconds Initialization: 30

Edge delay	Time in milliseconds, with which the writing of 0 is delayed for a single command. Only available for the action 'Single command'. The system does not wait until the watchdog timer is ended. Initialization: 1000 ms
Modifiable stati	List of the states which can be modified with the action 'Set status'. Only available for the action 'Status input' . Initialization: None modifiable
Nominal/actual value comparison	If this is active, there will be a check whether the value of the response variable already matches the switching direction. If this is true, an unlockable interlocking variable is shown. Initialization: Inactive
Menu spec. screen	Screen of type Command, which is activated when the action is activated over the context menu of the element. If no screen is entered here, the screen entered for the command group is used. A screen that was configured, but that does not exist, causes an error message during compiling and the action is not taken over. Initialization: No Allocation
Command button	Allocation of the action to an action button in the screen, defined at the command group. If the command group is used for another screen (e.g. via function), the allocation to the action button remains nevertheless. In other words, the action is always placed on the button with the allocated action ID. If such a button is missing, the action is not available in the screen. Only the action buttons that were not allocated yet are provided in the selection list. This setting is locked for the action type 'Lock' or if no screen was allocated to the command group. Initialization: No Allocation
Close automatically	If this is active then the screen is closed automatically after action execution. Initialization: Inactive
Two-hand operation	If this is active, the Control 'Execute 2' is only active when you hold Ctrl. Only available for two-step execution.

	Initialization: Inactive
`Two-stage`	If this is active, an action is executed only after operating the Control 'Execute 2'. If not active, the action is executed after releasing the last interlocking or, if there is no upcoming interlocking, immediately. Locked for the action 'Lock'. Initialization: Active

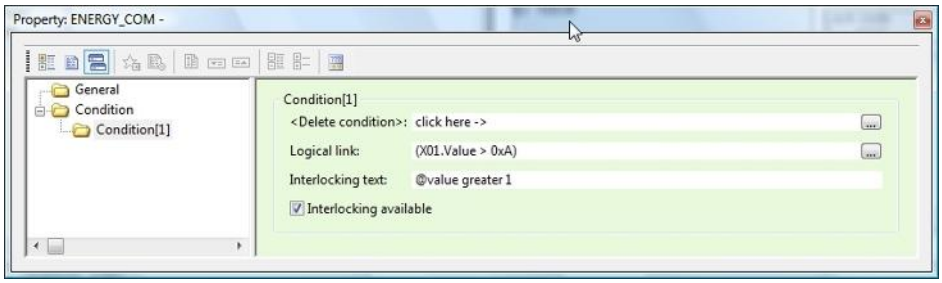
Example

We repeat this procedure and create additional actions for the switching direction 'OFF' and the actions 'Revision [On]' and 'Revision [Off]'.

4.2.3 Configuring command interlockings

We have already explained what interlocking conditions are for in the section about the configuration of the condition variable. We will now create a simple condition.

In order to do so, we select the action 'Single command: SW*_BE[On]' of our command interlocking and then the node 'Condition'. By clicking on the setting **`Condition new`** we create a new condition.



Explanation of the properties.

Name of the property	Description
Delete condition	The condition is deleted and is then no longer available.
Logical Link	Shows the formula, of which the result must be true in order to activate the interlocking. The formula can be edited with the formula editor. The formula editor is the same as the one of the Combined Element.
Interlocking text	This text is shown as the interlocking message, when the interlocking condition applies.
Unlockable	If this is active, the interlocking can be unlocked.

Adjust the settings according to the above image.

4.2.4 Assigning an interlocking to a variable

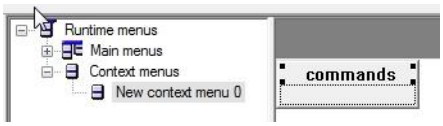
The command group must be assinged to the variables, so that it is actually used. This is done by assigning the desired command group in the setting 'Interlocking'.

We assign our command group to the following variables:

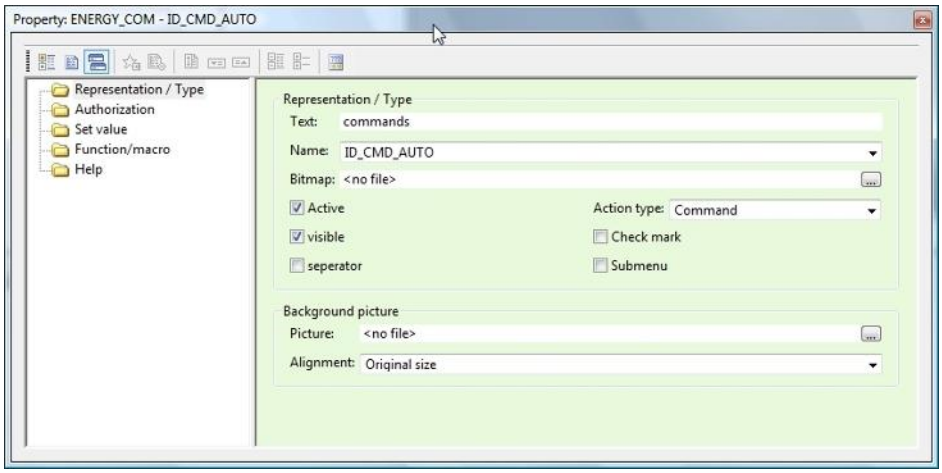
- SW_Switch1_RM
- SW_Switch1_BE
- SW_Switch2_RM
- SW_Switch2_BE

4.3 Create menu

Command input can also be activated via an element-specific context menu. For this, we create a context menu with an entry with the following settings:



Settings for menu entry:



The following two settings are important here:

Name of the property	Description
`Action type`	With the Energy license, this setting contains one more option. The value 'Command input' can be selected. This makes sure that the menu entry is assigned directly to a command input action. The assigned action is set in the setting 'Name'.
`Name`	This setting determines the action/actions to which the menu entry is assigned if the action type was set to 'Command input'.

We set these two settings to the depicted values. Start with the action type.

4.3.1 Assigning a menu

For this, we create a numerical element in the screen 'START' for all our variables SW*. If this screen was not created by the Wizard, please define it now and make sure that this screen is defined as the 'Start screen' of the project.

Info

You can see how such a screen can look in the chapter 'Start of the RT'

We assign our menu to the numerical element of SW_Schalter1_RM, SW_Schalter1_BE and SE_Schalter2_RM.

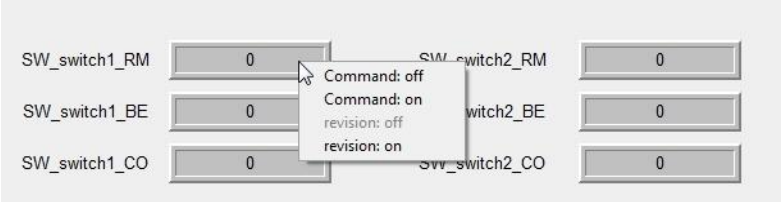
4.4 Starting the RT

Now we are ready to compile the project and to start it - if there are no error messages in the editor.

Our start screen could then look like this:

4.4.1 Execute command

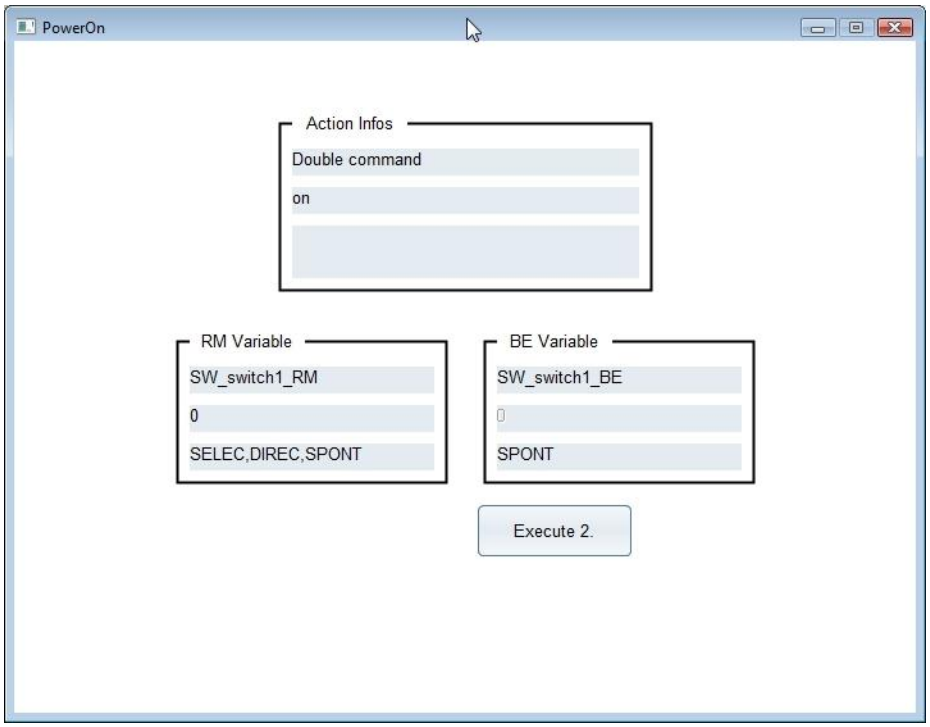
In order to execute a command, activate the context menu for the element of the variable SW_Switch1_RM.



You can see that the menu entry with the name ID_CMD_AUTO was replaced with the actions which are defined by the element of the command group of the variable (if the term "command group" is used in the following, it is always defined by this dependency).

Now select **`Command: on`** .

The system now automatically switches to the screen defined in the command group.



If you look at the screen you will notice that some of the controls that are visible in the Editor are now missing.

Control	Description
`Action 2`	There is no action assigned to the action button in the command group. Action buttons without action are hidden.
`Exit`	The screen is not modal.
`Execute`	We are not in level 1.
`Unlocking`	No interlocking condition active.

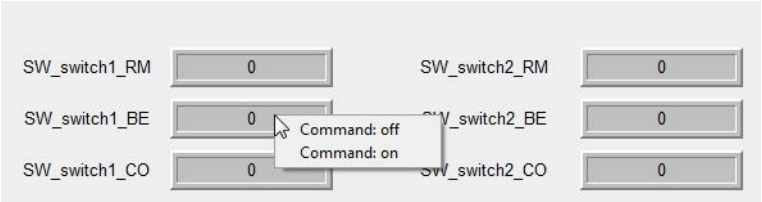
If the button **`Execute 2`** is clicked, the command execution is performed. This can be told from the fact that the command and response variable change to 1 and then immediately back to 0.

As the action was not configured to close the screen after execution, the screen changes back to step 1, where commands can be activated again.

With the action button 1, the associated action **`Command: on`** can be executed again.

4.4.2 Execute command, 2nd part

Activate the start screen and call up the context menu of the screen element of the variable SW_Switch1_BE.

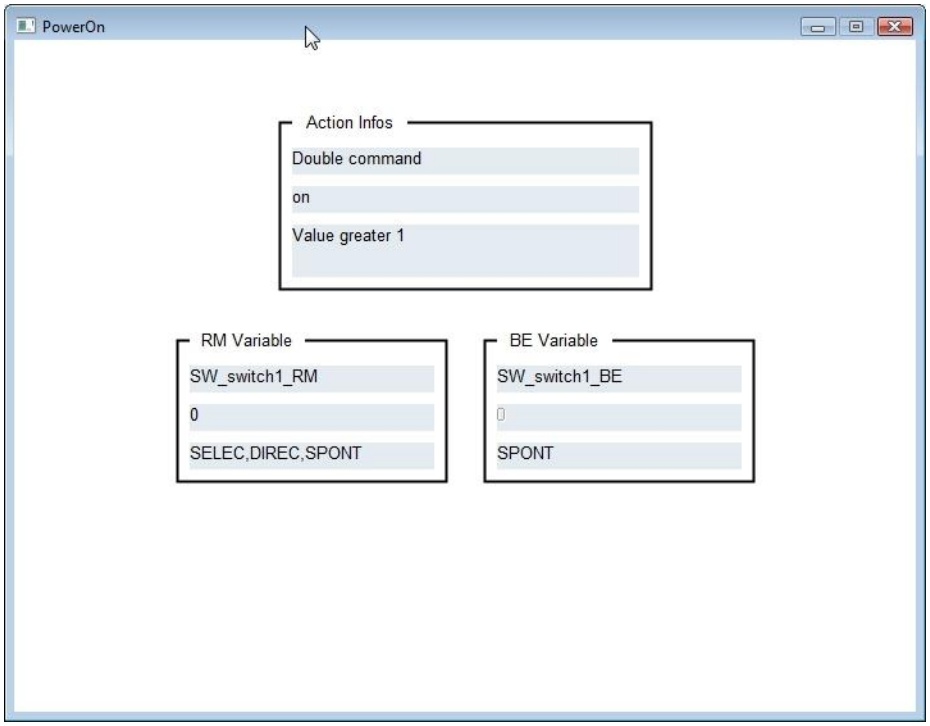


Here we can see that for the same menu and the same interlocking, only **`Command on`** and **`Command off`** is available. This is because the associated variable is the command variable. Status operations are not offered for a command variable.

4.4.3 Testing interlocking conditions

In order to activate our interlocking condition, the variable SW_Switch_CO must have a value > 10. So we perform a 'Set value' with the value 1000 for the variable.

Now we perform the action **`Command: on`** .



In the Control of the interlocking message, the interlocking text configured earlier is now displayed. The action can only be performed after the value of the interlocking has been set to '0' again.