

SCRIPTS CONFIGURATION

TABLE OF CONTENTS

1. ADDING THE SCRIPT TO THE CONFIGURATION	1
2. GRAPHIC MODE SCRIPT EDITION	4

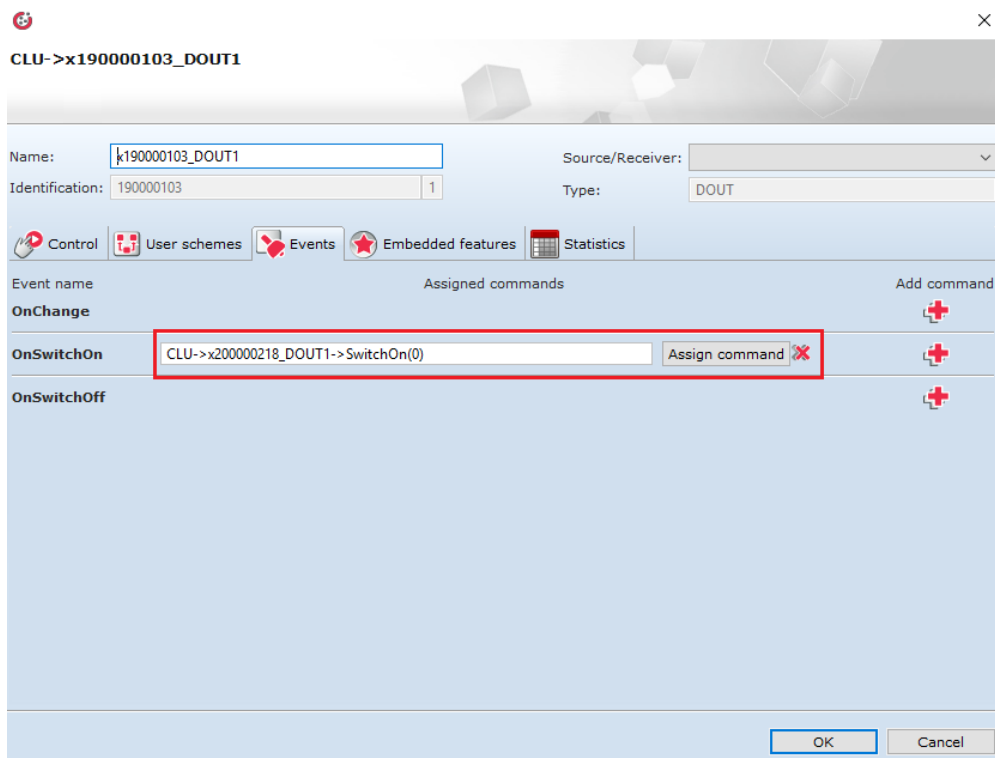
1. ADDING THE SCRIPT TO THE CONFIGURATION

1. Scripts Description:

- a. The configuration program Object manager is equipped in a tool for creating the scripts which can be used for checking the conditions or, for example, performing numerous actions assigned to one script (for example switching several relay outputs in one moment, using one instruction)
- b. Scripts in Object Manager program can be created in two modes:
 - Graphic – by the use of ready-to-use components
 - Textual – by the use of LUA script language's commands¹

ATTENTION! While creating the configuration by adding methods to particular events, assigned commands use LUA script language, therefore you can copy them to the script that is being created in textual mode.

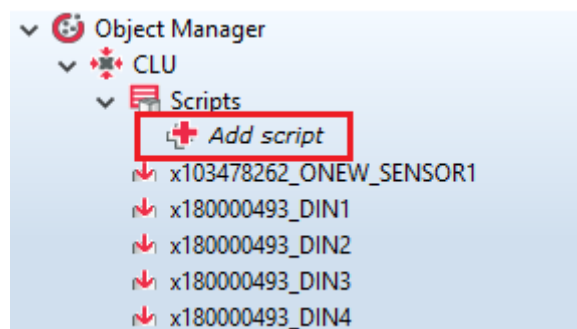
¹ In order to use the textual mode we recommend familiarizing oneself with the basic commands of the LUA script language. Information concerning the script language may be found at the address <http://www.lua.org>



ATTENTION! Creating the script in graphical mode enables switching to textual mode, which enables the return without applying the changes, whereas creating the script in a textual mode does not allow to switch to graphical mode.

2. Adding the Script

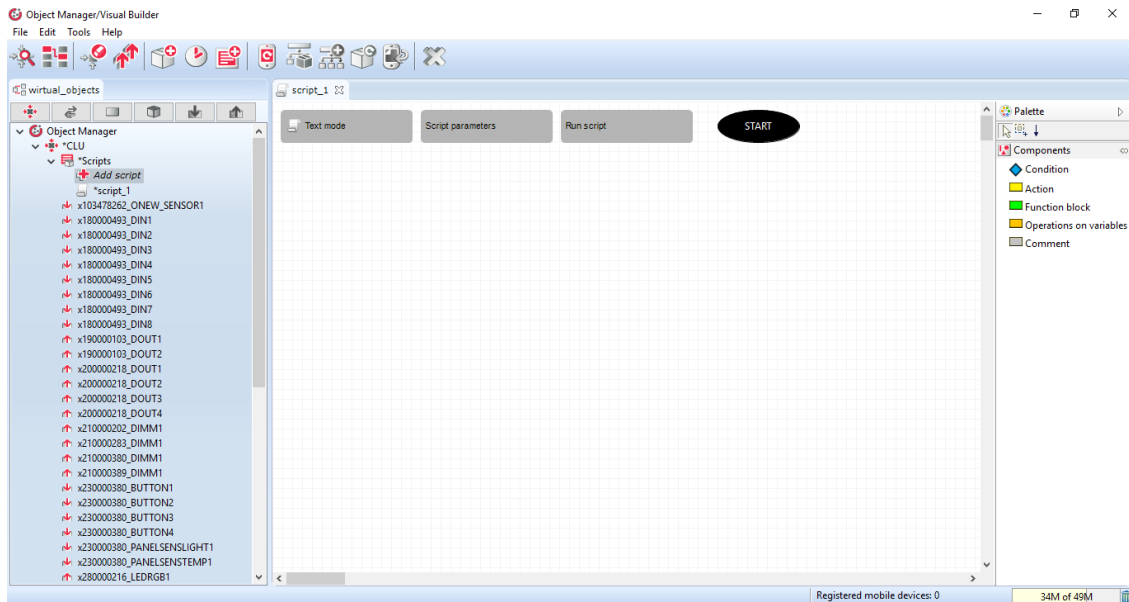
- To add a new script, run Object Manager, open the existing project or create a new one
- From the list of CLU modules, choose *Add script* option or use the button from the main menu tray



- c. In a window which appears on the screen, enter the name of a script

ATTENTION! The names of virtual objects cannot include spaces.

- d. Confirm your choice with OK button
e. The created script will appear on the module list at the left side of the program window
f. In the right part of the window, the tool Script builder will appear



- g. The window consists of an edition window and components available to create scripts in graphical mode
- h. The edition window consists of four elements:
- Mode-switching button: *Text view/Graphic view*
 - Script settings button – including settings for creating script parameters and the change of script name
 - Launch the script button – enabling launching a script from the position of Object Manager program.
 - The block of the initial point of the script in graphical mode – the connection of script blocks begins here.
- i. Components Palette consisting of:
- The choice of mouse usage mode – enabling catching and dragging elements, selecting an area (for moving or deleting the larger number of elements) and the mode of adding an arrow connecting the added blocks to the script edition window (in the moment of an edition of block connections)
 - *Condition block* – serves for checking the condition specified in the block, for creating scripts, depending on the condition of any variable or input/output
 - *Action block* – serves for invoking methods connected to module configuration and virtual objects added to the project or to invoke a script.
 - *Functional block* – serves for applying a delay in performing further part of a script

- *Operations on variables block* – serves for performing mathematical operations or assigning a particular value to the variable or reassigning the values described by module and object features used in the configuration

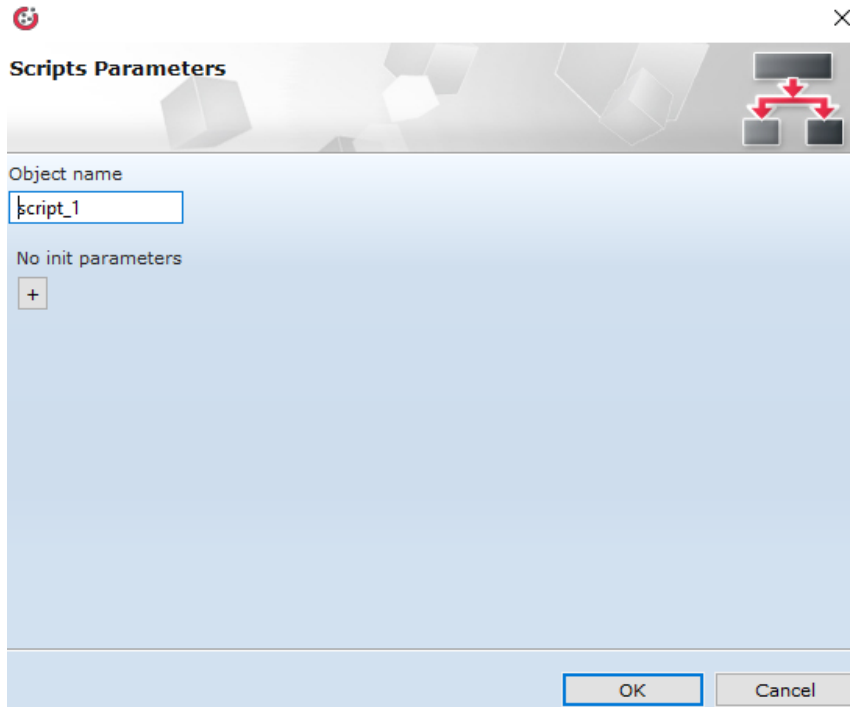
ATTENTION! Operations on variables may be performed only when there are script parameters or user's features on CLU are created. The block enables various operations, depending on the type of a given variable.


- *Comment block* – enabling inserting comments to the script

2. GRAPHIC MODE SCRIPT EDITION

1. Script Parameters

- Script parameters enable the creation of 3 types of variables: *string*, *boolean* and *number* (textual, logical and numeral) which may be used in the script
- In case of declaration of variables, while invoking the script, the values of these variables should be given in order to launch the script with specified data.
- To create script parameter, while having a script opened, click Script parameters button

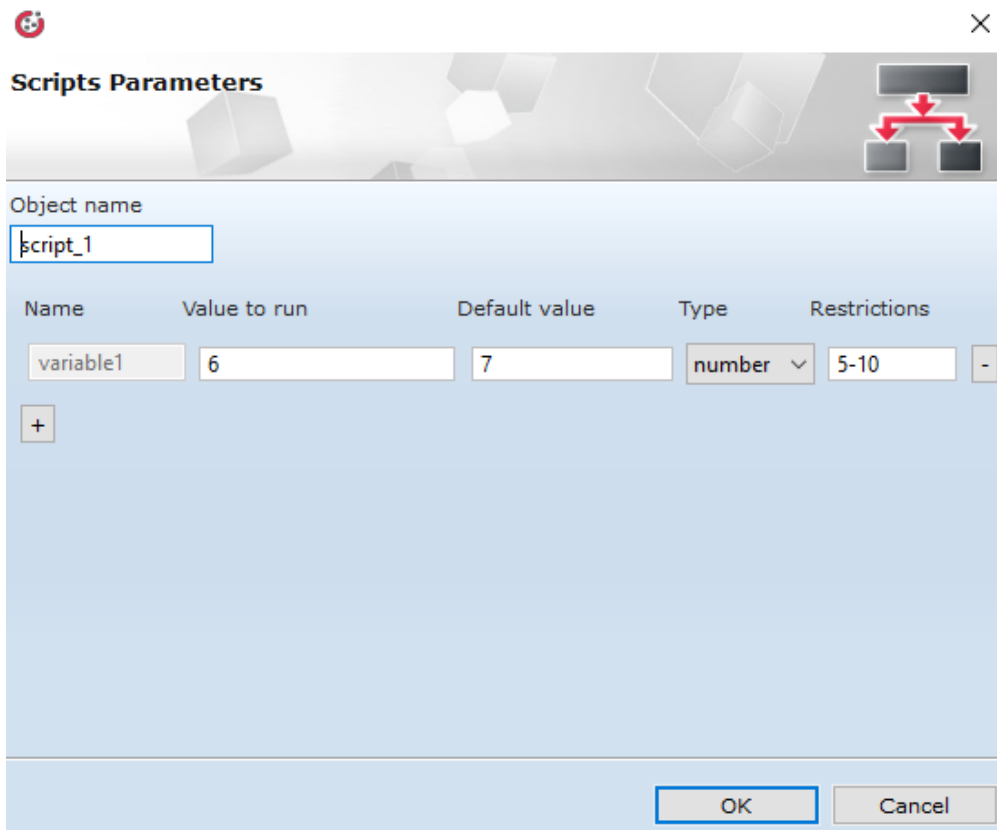


- Use  button and type the name of created variable

ATTENTION! The variables names must not include spaces.

- Click OK

- f. The created variable will appear at the variables list
- At the left side, its *Name* is being displayed
 - *Value to run* – this is a value, which is accepted to launch the script by the use of Launch the Script button (look: point 2h) in Object Manager Program
 - *Default value* – this is a value, which is accepted when we did not set any other value invoking the script and by the first launching.
 - *Type* – gives the possibility to choose one of three types of variables
 - *Restrictions* – enables setting the limits for the minimal and maximal value in case of a variable of number type. Setting the limits causes that launching the script with the value extending its scope results in launching the script, where the parameter will adopt the default value. We also type the limits in x-y mode, where x and y are the minimal and maximal limit values.





Scripts Parameters

Object name
script_1

Name	Value to run	Default value	Type	Restrictions
variable1	6	7	number	5-10

+

OK Cancel

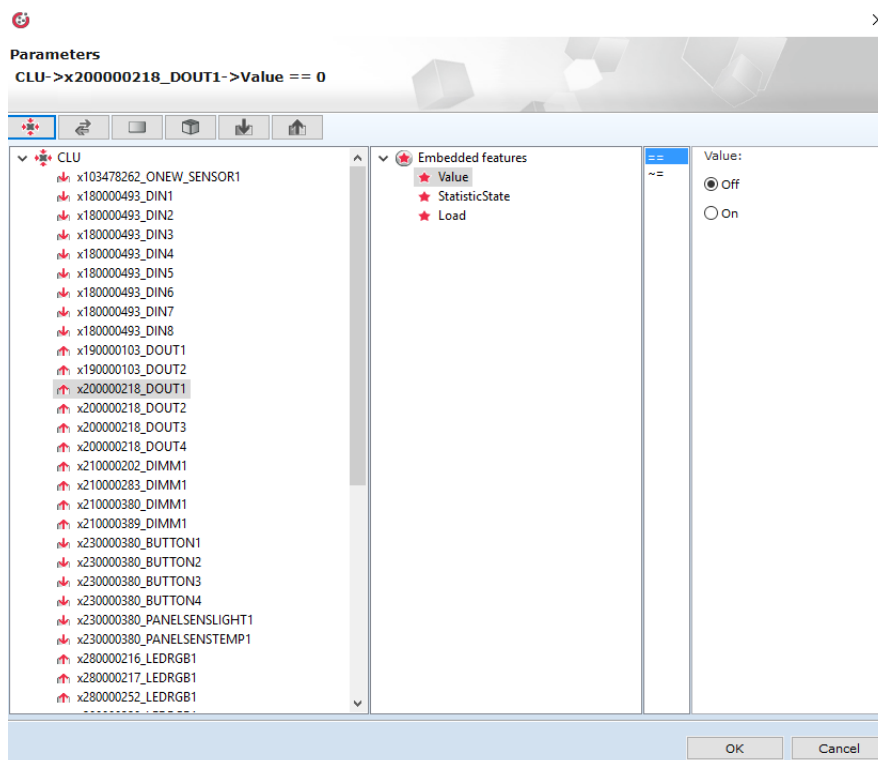
- g. Having completed adding variables, click on the OK button and continue creating the script or submit configuration to CLU
- h. In order to add a new variable, click again on 
- i. In order to delete a variable click on the right side from the variable 

2. Changing the Script name

- a. In order to change the script, click on the *Script Parameter* button (as shown on the picture above), change the name in the *Object Name* window, click OK and continue creating the script or submit configuration to CLU

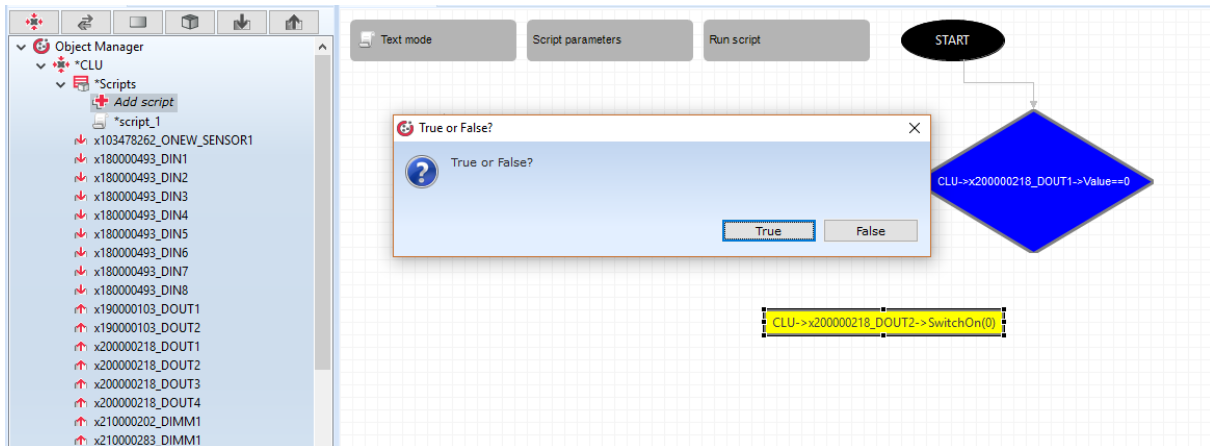
3. Creating the script

- In order to add elements from the pallet of components to the script drag them to the fields of script editing
- Having dragged the element, its connection with the previous block (or START block) is made automatically.
- Each added block to the window of script edition may be moved or its dimensions may be altered. To do so from the pallet select the cursor and click on the component you are interested in added to the script, drag it either by dragging its corners or edges change its dimensions.
- All blocks added to the script may be combined on both sides-one entry connection from the preceding block and one exit connection with the subsequent block. This does not concern the conditional block which has two exit connections-for the fulfilled and unfulfilled condition.
- Adding *Condition block*
 - To add conditional block click and drag it from the palette of components
 - You will see a window of selection of the condition on the screen, based on the inbuilt features of the connected modules, script parameters, inbuilt features of the virtual objects added to the configuration or user features in CLU

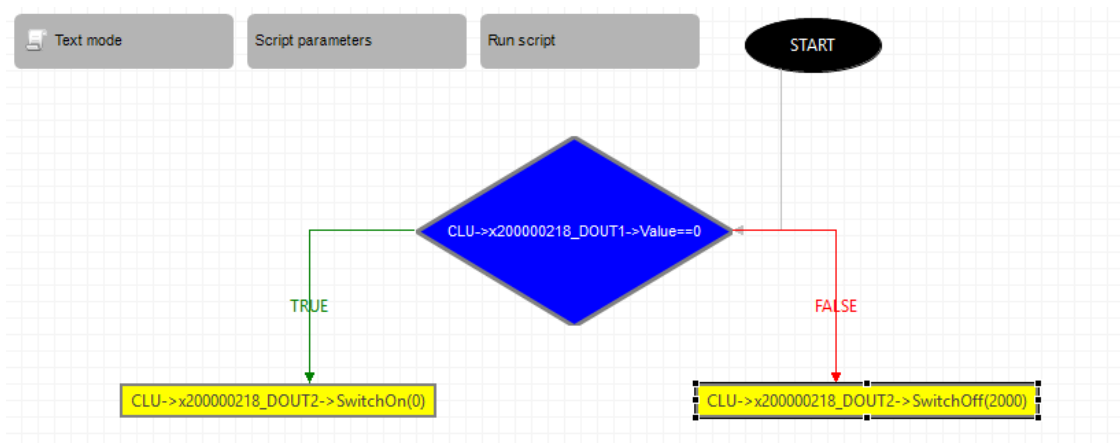


- Post entering the condition and confirming with an OK button, the conditional block will be added to the script

- Post dragging another block to the script edition window the programme will ask you whether the subsequent block is to be added after fulfilment or non-fulfilment of the condition



- Having choosing Truth or False option the block will be connected with the condition block
- After adding the subsequent block it will be by default connected with the previously added block. To perform the connection with condition block for non-fulfilment of condition (as on the shown picture) delete the connection of newly added block indicating with an arrow which one you wish to remove and after selecting the connecting arrow from the pallet, click on the condition block and newly opened condition.

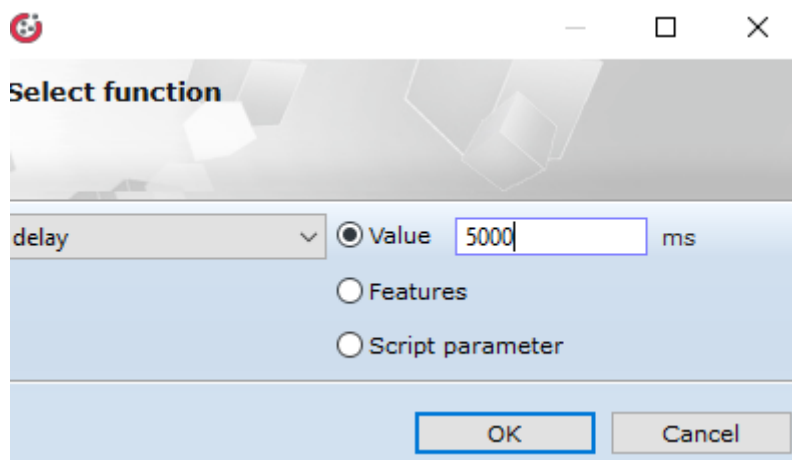


- Tin order to change *TRUE* and *FALSE* use double-click on one of the arrows existing in condition
- f. Adding *Action* block
- In order to add an Action block drag it to the script edition window which will automatically connect to the previously added element and a window of indication of method we wish to evoke will appear on the screen

- In the open window select the method in the same way as they are assigned to events of modules connected to the system
- OK Confirm with OK button

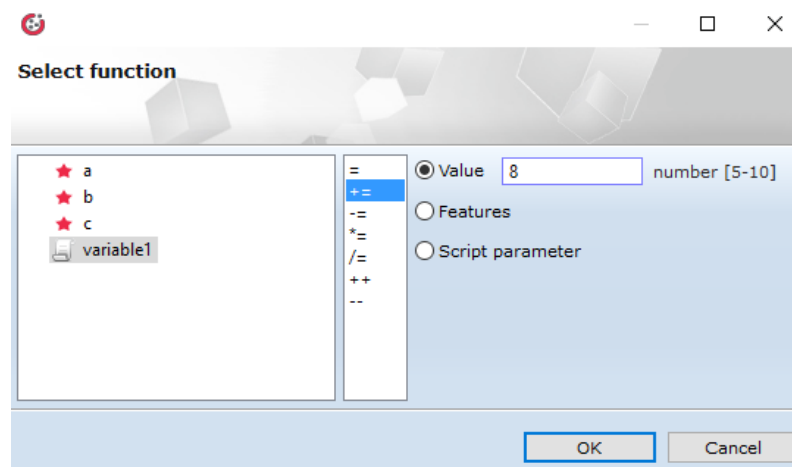
g. Adding *Functional* block

- To add delay in performance of the script, drag the component *Functional Block*
- Select *delay* from the drop down list in the window which will appear on the screen
- Enter the value of time delay in milliseconds directly, with the help of the inbuilt feature or the script parameter



h. Adding *Operations on variables* block

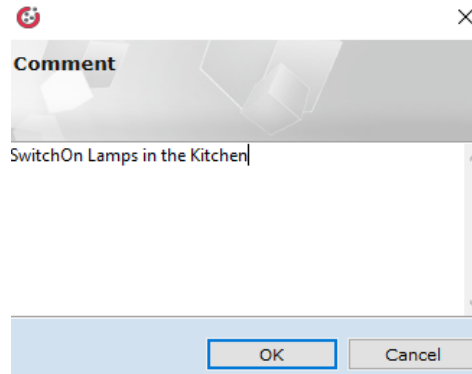
- To perform operations on variables in the script, drag the block to the window of script edition
- On the screen a window of selection of the user feature or script parameter will appear to which you may assign a specific value directly, by means of an inbuilt feature or script parameter.



i. Adding *Comment* block

- To add a comment drag the block to the window of script edition and enter any text of the comment you wish to place in the script in the window

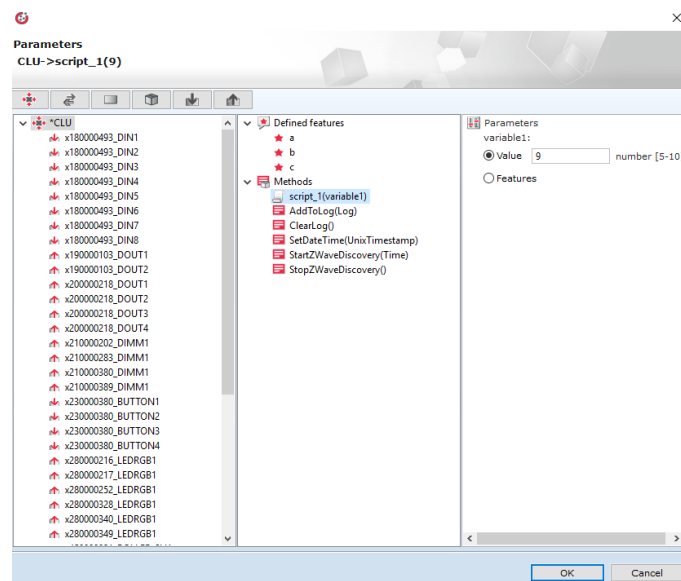
- The comment is not connected with an arrow of completion in graphic mode with any block, you may move it around freely in the script window and place it next to the manual in order to comment on it.



- Click on OK to confirm your selection
- Having completed creating the script submit the configuration to CLU

4. Launching the Script

- You may launch the script in a several ways:
 - Using the *Launch Script* button-in this mode, if you have created the script parameters, the variable will assume *Value for launching*
 - Assigning the method to specific event, click on CLU, select the script name and if you have created the script parameters, enter the demanded value which the parameter is to assume upon launching the script. If you don't enter any value the script will be launched assuming Default value entered upon adding a variable as values of parameters.



- Using Action component in another script according to the same rules as in the previous point