

VIRTUAL OBJECT – PID CONTROLLER MANUAL

Object Manager Configuration

TABLE OF CONTENT

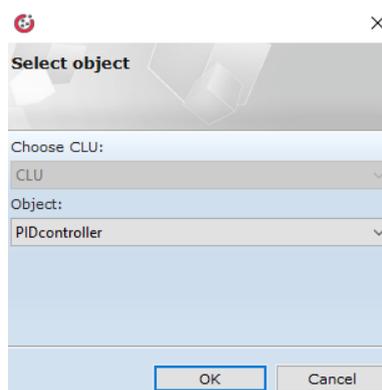
1. **Dodawanie Obiektu Wirtualnego** _____ Błąd! Nie zdefiniowano zakładki.
2. **Konfiguracja Obiektu Wirtualnego** _____ Błąd! Nie zdefiniowano zakładki.

1. ADDING THE VIRTUAL OBJECT TO THE CONFIGURATION

1. Create Object Manager configuration software
2. Create a new project and perform *CLU Discovery* Open an existing project and communicate with CLU
3. From the main menu select option Add *CLU virtual object*



4. Select object *PIDcontroller*



5. Confirm with OK button
6. Enter a new name of PID Controller and confirm with OK button

ATTENTION! Names of virtual objects cannot contain spaces.

7. After creating an object, configuration window of virtual object PID controller will appear on the screen-it has 3 tabs
8. Open the tab Inbuilt features and enter the value features SetPoint(see: point 2e)
9. You may close the window and submit configuration at the stage of created virtual object which does not steer any device or else proceed to the subsequent step Configuration of Virtual Object

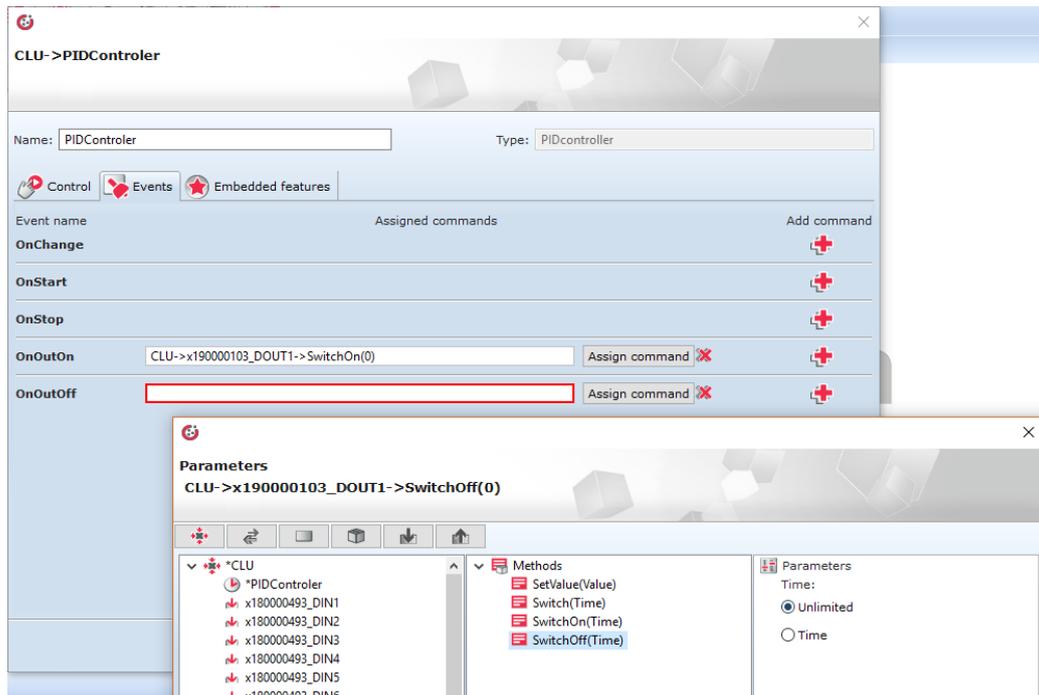
2. VIRTUAL OBJECT CONFIGURATION

1. Basic Configuration

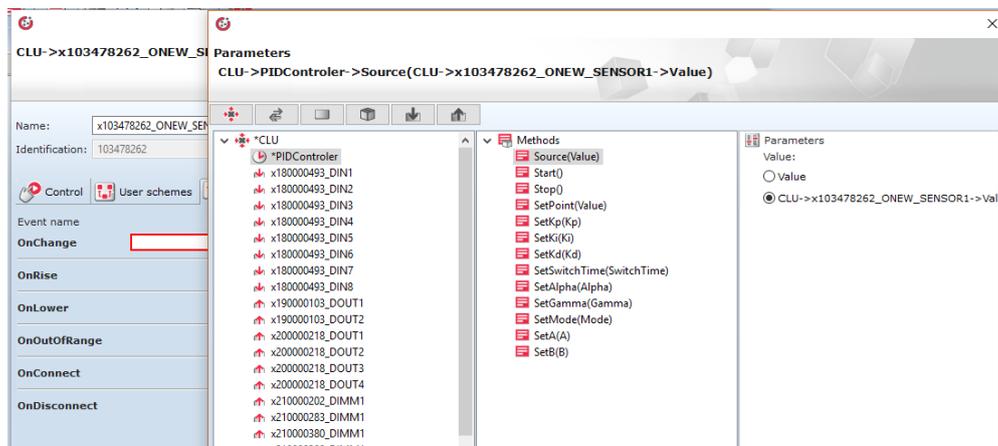
- a. PID Controller enables connecting a device which is to be connected and disconnected on the basis of a sensor in line with the operations of PID controller which turns on the output of the virtual object assigned to real contacts and bearing in mind inertias, it switches them off sufficiently in advance-for example in case of floor heating
- b. Once the window of virtual object appears, after its creation, open the tab Embedded features and enter the initial value of the feature *SetPoint* (see: point 2e) - in case of floor heating this needs to be the temperature value which we wish to obtain in the area of the sensor
- c. In the tab Embedded features select the automatic mode - Auto- Kaczmarz in feature *Mode* (see point 2i)

ATTENTION! In case of selecting manual mode, select the parameters of controller in accordance with your expectations

- d. In case of selecting automatic mode select parameters in accordance with the Kaczmarz algorithm- features A and B or leave the suggested values
- e. Proceed to tab Events and to events *OnOutOn* and *OnOutOff* (see: point 3e) assign methods of switching on and off of relay output (by clicking on ), to which in case of the example the floor heating is connected



- Close the configuration window of the controller with an OK button
- Double click on the sensor on the list of modules connected to CLU and detected by OM (in case of steering the floor heating-temperature sensor)
- Go to tab Events and to event *OnChange* set it by clicking on  method Source of PID controller with assigning the values of this sensor-assigning will cause that upon each change of temperature sensor the virtual object will obtain current data. Close the configuration window by clicking OK



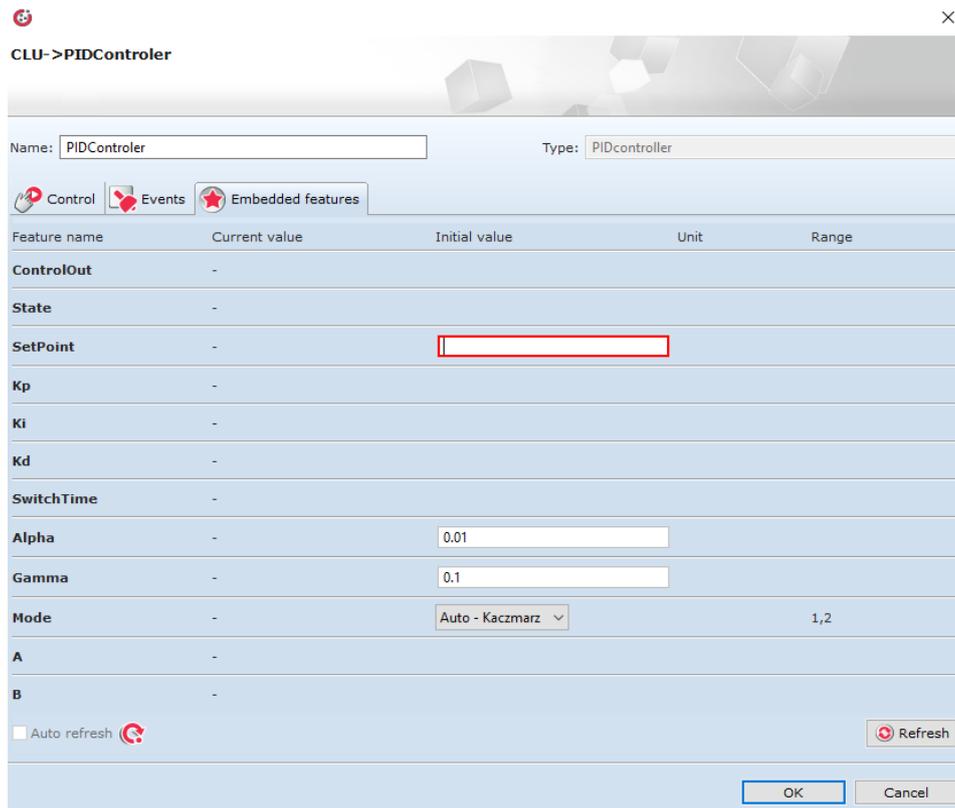
- Assign the same method to the event *OnInandt* CLU, so that the controller after launching the system obtained the information of current temperature-for this purpose click on CLU on the list of available modules, proceed to tab Events on click on  and proceed as in the previous point

- g. Submit configuration to CLU (to do more detailed configuration you may follow the below instructions)
- h. After sending the configuration make sure that the controller is in operational mode (feature State is set to value 1 - see: point 6)

2. Embedded Features Tab

- a. The tab enables preview of values specifying the virtual object
- b. After creating configuration you should enter the initial values of the virtual object in order to set up its operations
- c. Feature ControlOut shows information on connecting the output of PID controller where 1 is a connected outlet and 0 is inactive outlet
- d. Feature State specifies the status of virtual object. State of value 1 indicates that the object is analysing data indicated in the feature Source and in line with the settings of the controller it connects and disconnects the output
- e. Feature SetPoint stores information on the values which the PID Controller is to reach-for example the value of temperature
- f. Features Kp, Ki, Kd are the parameters of strengthening proportional-integrating-derivative terms specifying the operations of the controller-in manual mode they must be indicated by the user-in automatic mode they are not considered
- g. Feature SwitchTime in automatic mode is selected by the controller via calculation of time of connection specifying the cycle of controller's work.
- h. **NOTICE!** With the selected automatic mode the change of its value is not possible.
- i. Feature Alpha and Gamma involves parameters in Kaczmarz algorithm responsible for securing against neutralization of denominator and dynamics of changes of estimating the parameters a and b
- j. Feature Mode defines the mode of operations of PID controller:
 - Automatic - selecting parameters of the controller individually on the basis of conditions in a given room- after launching the work of controller (method Start) with the created basic configuration (see: point 1), controller connects the relay output and performs measurement of time of connection- in automatic mode it performs it in each cycle and conducts changes It's work is based on selection of features A and B responsible for the work of controller in line with the Kaczmarz algorithm
 - Manual- this mode requires setting up the parameters of controller-features Kp, Ki, Kd and feature SwitchTime responsible for the working cycle of the controller

- k. Features A and B specify the work of controller in automatic mode. They stand for parameters a and b in Kaczmarz algorithm. Their change is impossible during the work of the controller-when feature State is equal to 1. Their values are of importance to the controller set in the manual mode



CLU->PIDControler

Name: PIDControler Type: PIDcontroller

Control Events Embedded features

Feature name	Current value	Initial value	Unit	Range
ControlOut	-			
State	-			
SetPoint	-			
Kp	-			
Ki	-			
Kd	-			
SwitchTime	-			
Alpha	-	0.01		
Gamma	-	0.1		
Mode	-	Auto - Kaczmarz		1,2
A	-			
B	-			

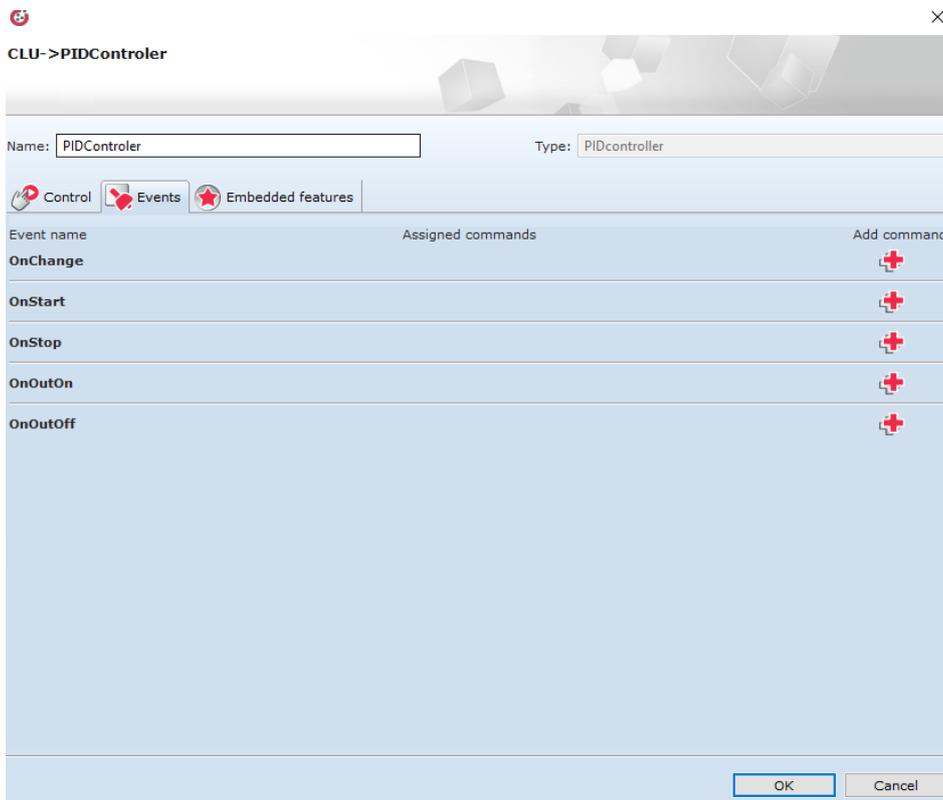
Auto refresh Refresh

OK Cancel

3. Events Tab

- Virtual Object PID Controller reacts to 5 events to which you can assign a specific action which they are to carry out after the occurrence of events
- Event *OnChange* is triggered at the time of change of state of the exit of virtual object assigned to the feature *ControlOut*
- Event *OnStart* is triggered at the time of change of feature State from 0 to 1 - launching of schedule operations
- Event *OnStop* is triggered at the time of changing of feature State from 1 to 0 - ceasing the schedule operations
- Event *OnOutOn* is triggered at the time of changing the status of feature *ControlOut* from 0 to 1 - connecting the output of virtual object
- Event *OnOutOff* is triggered at the time of changing the status of feature *ControlOut* from 1 to 0 - switching off the output of virtual object

- g. It is possible to add actions to events by pressing  upon a given event so that they are triggered at the time of occurrence of a given event

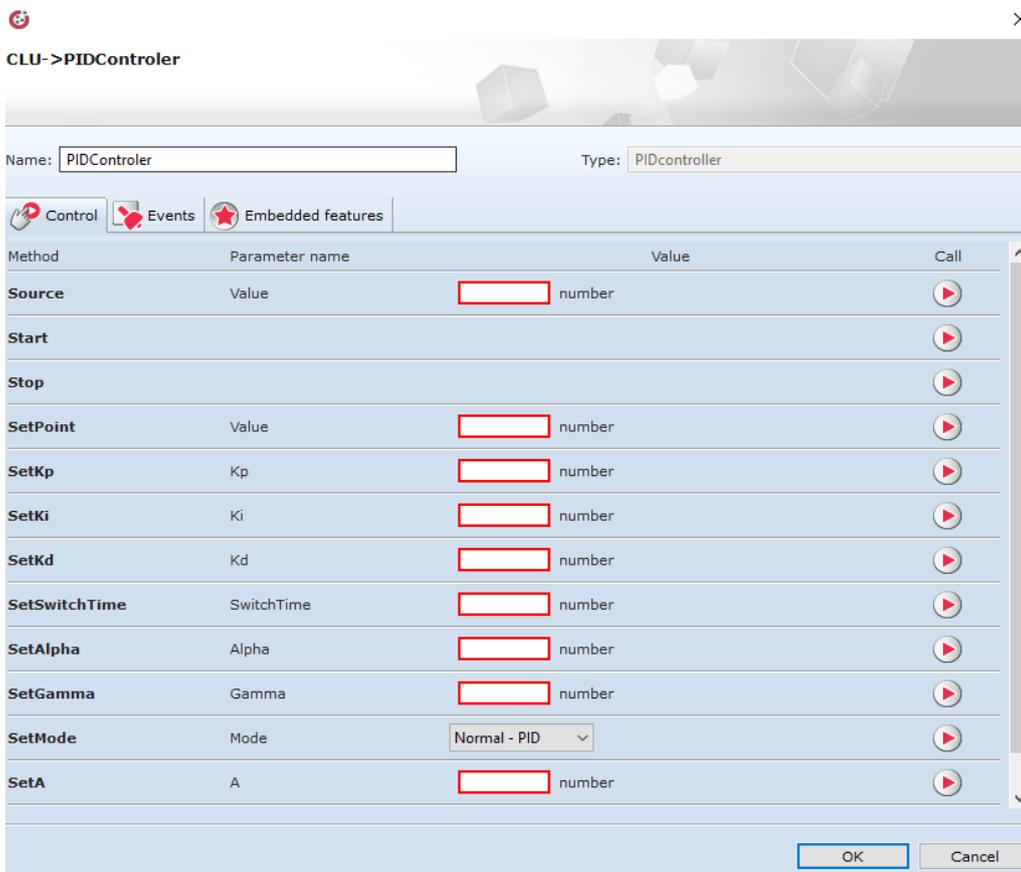


4. Methods Tab

- This tab contains 13 methods which may be set as actions for occurrence of specific events or which may be triggered by clicking the button  upon a given method. These features trigger changes of values of inbuilt features of virtual object responsible for the work of controller.
- Method *Start* is used for launching actions of the virtual object. Evoking the method sets up the feature State of the object on value1
- Method *Stop* is used for ceasing the work of virtual object. This method sets up the value of feature State to 0
- Method *SetPoint* is targeted at setting up the target values which PID controller is to reach (i.e. temperature). Evoking the method causes a change in feature *SetPoint*
- Method *SetKp* s is used to set up the proportional parameter of PID controller. Evoking the method triggers a change in feature *Kp*
- Method *SetKi* serves the function of setting up the integrating PID controller parameter. Evoking the method causes a change in feature Ki

- g. *SetK* method is used to set the derivative PID controller parameter. Evoking the method causes a change in feature Kd
- h. Methods *SetSwitchTime*, *SetAlpha*, *SetGamma*, *SetMode*, *SetA*, *SetB* are used to set up appropriate features *SwitchTime*, *Alpha*, *Gamma*, *Mode*, *A* i *B*

ATTENTION! If, after evoking the methods, CLU is reset, features will be set as during the initial set up and submission of configuration to CLU



- 5. After completing the configuration, send it to the CLU



- 6. Make sure that features State of all created controllers are equal to 1. In order to check, double click on the virtual object of PID Controller on the list of modules and proceed to tab Features. Otherwise, launch it by means of method Start in each of the objects (see: point 4b)

CLU->PIDControler

Name: Type:

Control Events Embedded features

Feature name	Current value	Initial value	Unit	Range
ControlOut	0			
State	0			
SetPoint	35	<input type="text" value="35"/>		
Kp	0.30			
Ki	0.03			
Kd	0.10			
SwitchTime	60			
Alpha	0.01	<input type="text" value="0.01"/>		
Gamma	0.10	<input type="text" value="0.1"/>		
Mode	2	<input type="text" value="Auto - Kaczmarz"/>		1,2
A	0.90			
B	0.10			

Auto refresh 