Grenton Sp. z o.o.
ul. Na Wierzchowinach 3
30-222 Krakow, Poland

VIRTUAL OBJECT – TIMER MANUAL

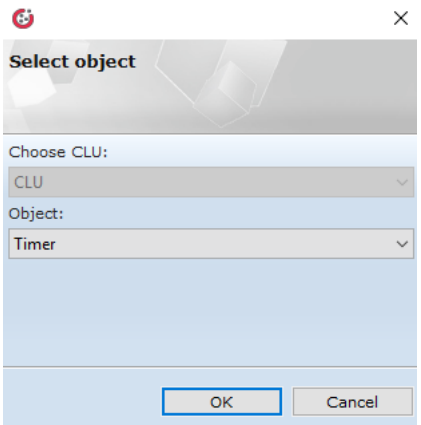
Object Manager Configuration

TABLE OF CONTENTS

1. ADDING VIRTUAL OBJECT TO THE CONFIGURATION

1. Open Object Manager CONFIGURATION SOFTWARE
2. Create a new project and execute *CLU Discovery* / open the existing project and communicate with CLU
3. From the main menu choose *Add CLU Virtual Object* option

4. Choose the Timer object



5. Enter the name of a timer and confirm with OK button

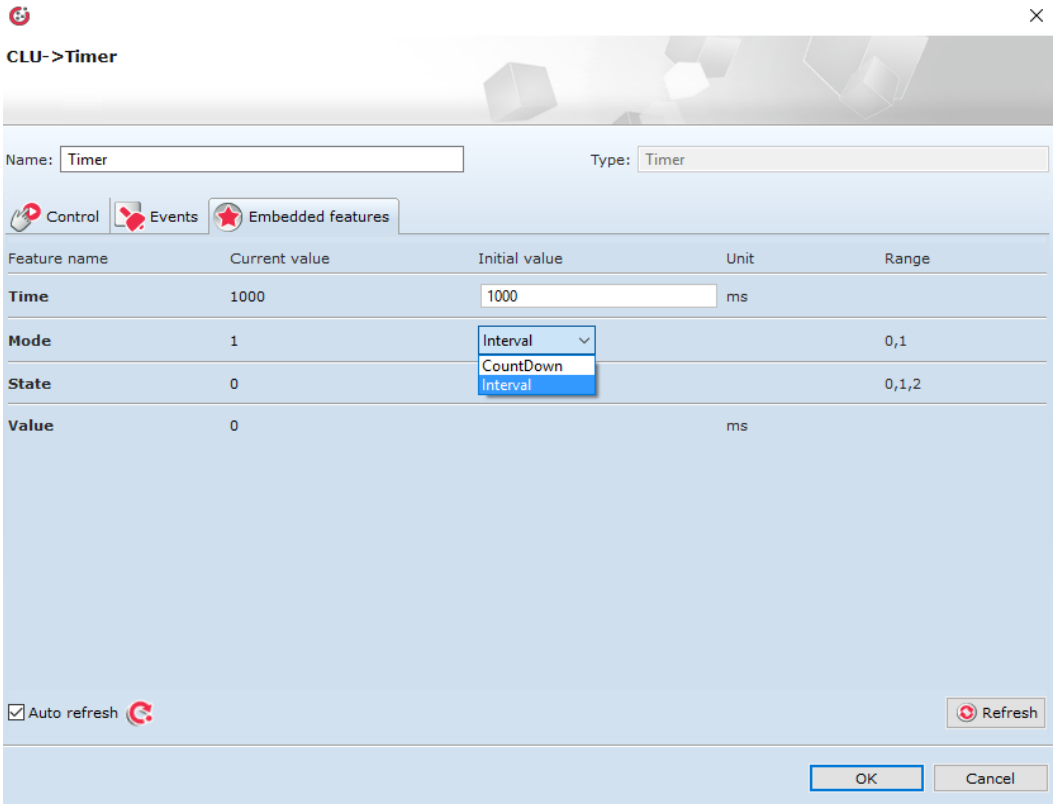
   ATTENTION! The name cannot contain spaces.
6. After creating the object on the screen, a virtual object Timer configuration window will appear on the screen. – It has three tabs
7. The window can be closed and the configuration can be sent at the stage of the created virtual object which does not control anything or proceed to step Configuration of Virtual Object

2. VIRTAUL OBJECT CONFIGURATION

1. Basic Configuration
   a. The virtual object timer is designated for multiple continuous or single countdown of the set time, after which the invocation of action is possible
   b. After the appearance of virtual object window, open Embedded features tab
   c. Fill in the initial values of Time and Mode initial values (see: point 2c and 2d)
   d. Go to the Events tab
   e. Add the desired method to the event *OnTimer* (see: point 3b)
   f. Close the timer configuration window
   g. Send the configuration (see: point 5)
   h. Open again the virtual object Timer from the list of modules connected to CLU
   i. Go to Methods tab
   j. Use the  button to Invoke *Start* method
   k. Send configuration to CLU (in order to execute more detailed configuration you can use the following tips)
   l. After sending the configuration, make sure that the schedule is in working mode (feature *State* is set on value 1 – see: point 6)

2. Embedded Features Tab
   a. The tab enables to preview the values defining the virtual object
   b. While creating the configuration you should fill the initial values of virtual object features in order to set its action.
   c. *Time* feature is responsible for the period of time which is counted after launching thermostat action.
   d. *Mode* feature defines the mode of timer work – *Interval* or *CountDown* (see: point 3f)
   e. *State* feature defines the state of the virtual object. The state with a value of 1 means that the object counts down the set time in *Time* feature. Value 0 means that the timer does not count down the time

f. *Value* feature presents the remaining time for invocation of the *OnTimer* event. Directly after launching the timer, value feature presents the same value which is entered into the Time feature, after which this value begins to decrease while counting down



3. Events Tab
   a. Virtual object reacts to 4 events, to which you can assign particular actions, which are to be executed after the occurrence of events
   b. *OnTimer* event is invoked after the end virtual object's countdown for the Time set in the feature. To the event one can assign for example the method controlling i.e. invoking of the script. To add the method of the other device or script to the event, use the button by  button
   c. *OnStart* event is invoked when the *State* feature is changed from 0 to 1 – the moment of launching schedule work
   d. *OnTime* event is invoked when the *State* feature is changed from 1 to 0 – the moment of stopping the schedule work
   e. The *OnPause* event is invoked when the *Pause* method is invoked

4. Methods Tab
   a. This tab includes 5 methods which can be set as actions for the occurrence of particular events, or it is also possible to invoke them by clicking  button by the given method
   b. Start Method serves for launching the work of the virtual object. Invoking the method sets State feature of the object to value

c. Stop Method serves for stopping the work of the virtual object. The method sets the value of a feature State to value 0

d. Pause method serves for temporary stop of virtual object countdown. After resuming the countdown by the use of Start timer continues the countdown from the paused moment

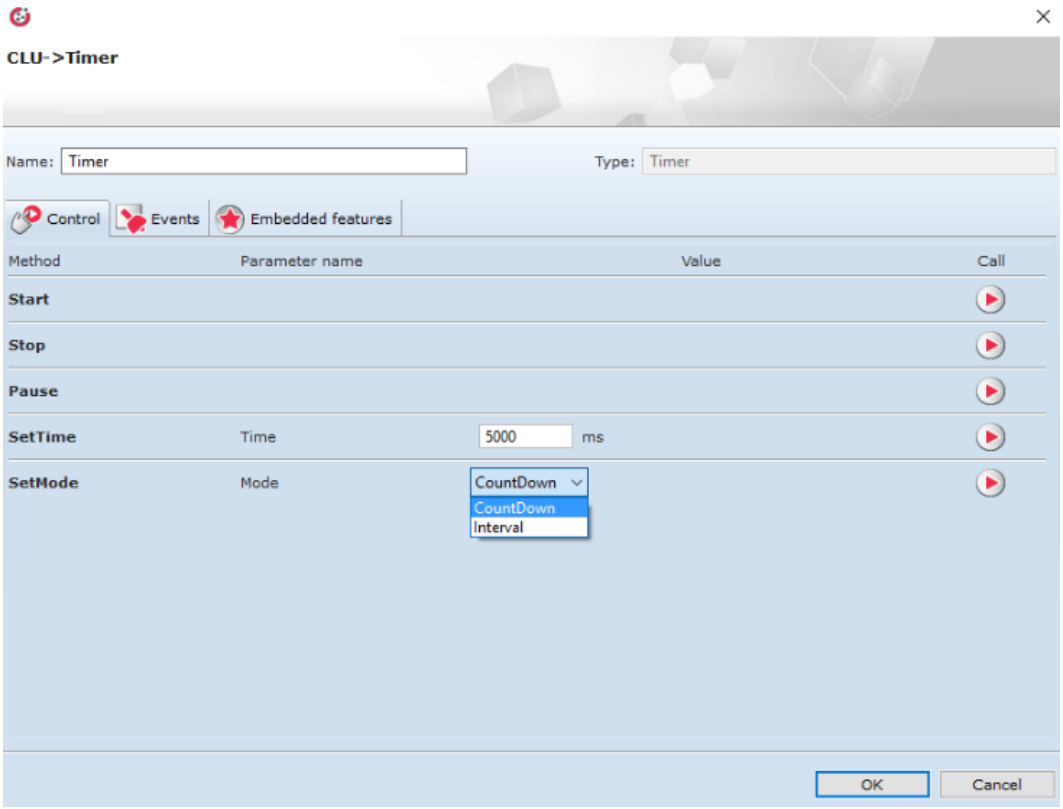
e. *SetTime* method serves for setting Time feature responsible for the length of the counted time

ATTENTION! If, after executing the method, CLU was reset, Time feature will be set as in the first execution and sending the configuration to CLU

f. *SetMode* method serves for setting timer mode:

- *Interval* – in this mode timer counts down the time after which *OnTimer* method is invoked and the timer starts the next countdown
- *CountDown* - in this mode timer counts down the time, after which *OnTimer* method is invoked and timer switches off

ATTENTION! If, after executing the method, CLU is reset, Mode feature will be set as in the initial execution and submission the configuration to CLU.



5. After finishig creating the configuration, send it to the CLU

6. Make sure that *State* features of all created timers counting down are equal to 1. To check it, double-click on virtual object Timer on the module list and go to Features tab. If not, launch it by means of the *Start* method in each schedule