Smart Home – Presence Simulator

Many people when leaving home for a longer period, for example, going on a holiday, wonder how to secure home against burglary. One option is an intrusion detection system, anti–theft shutters and monitoring. However, a thief can watch the house and closed shutters for several days in a row is a clear signal, that the house is empty. Much cheaper and very effective deterrent are light that are turned on – it will signal that someone is at home and thus scare the thief. But how to achieve such functionality and turn on lights only at certain moments of a day and imitate natural human behaviour? This is where a smart home comes into play with functionality of the presence simulator. In this tutorial, we will present how create the presence simulator using a **DOMIQ/Base** module.

1. The Concept of Presence Simulator

The main functionality of the presence simulator is lighting control, that imitates human behaviour – selected lights are being turned on in a random order and remain turned on for a random period of time. This cause the system to work in a pseudo–chaotic manner, which further increases the realism of the simulator.

The second element of the presence simulator's functionality is the control of shutters/ blinds based on timers. For example, shutters can be closed at sunset and raised at sunrise. External lighting can be controlled in the similar way. You can also use garden watering as a part of the presence simulator.

Simulation can be started/stopped in two ways. The first way is fully automatic – based on timers. During configuration, you need to set time for the start of simulation, for example at sunset and time for the end of simulation, for example the time when residents usually go to bed. The second way allows to start/stop simulation manually using the **DOMIQ/Remote** or **DOMIQ/AirDisplay**.

When simulation is running, lights will be turned on in a random order and will remain turned on for a random period of time. The cycle will be repeated until time of the end of simulation is reached.

2. Running the Presence Simulator

Running the presence simulator consists of several steps that are described later in this chapter. The description only includes control of interior lighting. Control of shutters/ blinds, external lighting, watering of a garden should be realized using standard timers.

2.1. The Script

The main part of the presence simulator is the script that is the attached to this tutorial. The script need to be uploaded to the **Base** module in the **Resources** tab (**Resources** > **Scripts)**. Next, navigate to the **Logic** tab and import the script using the following command: `import'presencesimulator'`

The next step is customization of the script to suit the particular installation:

1. Add lights that will be used in the simulation. In order to do that, invoke the `addDevice` function for each device and pass the device's identifier as an argument. Below is the example of adding two lights.

```
addDevice('LCN.output.0.10.1')
addDevice('EXP.ouput.7.1')
```

2. Set the maximum time, expressed in minutes, for how long lights will be turned on. This value is also the maximum interval between successive switches of a given light. The default value is 30 minutes. If you want to use the default value, then this step can be omitted. In order to change the default value, invoke the `setInterval` function in the **Logic** tab. The example below presents, how to set the interval to 60 minutes:

```
setInterval(60)
```

3. At the end put the following line of code: `checkStart()`

2.2. Timers

In order to run the presence simulator automatically, two timers need to be defined: the first one to run the simulator and the second one to stop it. We recommend to set up the start time 30 minutes after sunset. The definition of timers is as follows:

IMPORTANT! Use only names of MEM variables shown below. Using other names will prevent proper operation of the simulator.

1. Add a timer – it will be triggered at the scheduled time of start of presence simulation:
   1. In the **Hour** field enter: *sunset*, if you want to start simulation at sunset or *MEM.simulationstart.hour* if the start time should be set by user.
   2. In the **Minute** field enter: *30* in order to trigger the timer 30 minutes after sunset. You can enter any other offset as well. Type *MEM.simulationstart.minute* if the start time should be set by user.
   3. In the **Actions** section click on the **Add command** button. In the new window, in the **Name** field enter: *C.MEM.simulationtrigger* and in the **Value** type: *on*
   4. Add another command, in the **Name** field enter: *C.LOGIC* and in the **Value** enter: *checkStart()*. The presented order of commands is mandatory.
2. Add another timer – it will be triggered at scheduled time to stop the presence simulation. **The end time of simulation may not be later than 23:59 or earlier than the start time of the simulation**.
   1. In the **Hour** field enter: *MEM.similationend.hour*.
   2. In the **Minute** field enter: *MEM.simulationend.minute*
   3. Click on the **Add command** button in the **Actions** section. In the **Name** field type: *C.MEM.simulationtrigger* and in the **Value** enter: *off*

2.3. User Interface

The user interface consists of 2 or 3 elements, depending on, if simulation time will be set automatically or by user.

1. Go to the **Remote** tab.
2. Add a **Switch** and double–click on it. In the **Channel** field enter: *MEM.presencesimulation*. State of the switch is check by the simulation script. The switch allows to block

execution of simulation script. If the switch is turned off at the scheduled start time of the simulation, the simulation will not start. However, if the switch was turned on later than the scheduled start time of the simulation, but before the end of the simulation, then the simulation will start immediately. Similarly, if the switch was turned off when simulation is running, it will be stopped immediately.

3. Add a **Time** element – it will be used to set the start time of the simulation. You can ommit this step, if you plan to run simulation at sunset. Otherwise, please follow the procedure below:
    1. In the **Hour Channel** field enter: *MEM.simulationstart.hour*
    2. In the **Minute Channel** field enter: *MEM.simulationstart.minute*
4. Analogically add another **Time** element, which will be used to set the end time of the simulation.
    1. In the **Hour Channel** field enter: *MEM.simulationend.hour*
    2. In the **Minute Channel** field enter: *MEM.simulationend.minute*

2.4. Events

As we mentioned in the previous subchapter, the switch in user interface allows to start the simulation manually after the start time of the simulation. In order to achieve this functionality, you need to add an event, which will be triggered by turning on the switch. The procedure is as follows:

1. Add an event.
2. In the **Channel** field enter: *E.MEM.presencesimulation*
3. In the **Data** field type: *on*
4. In the **Actions** section, click on the **Add command** button. In the **Name** field enter: *C.LOGIC* and in the **Value** field type: *checkStart()*

Additionally, you need to create an event that will be triggered when user changes the end time of simulation. This event is necessary for proper operation on the simulation script. Please follow the procedure below:

1. Add an event.
2. In the **Channel** field enter: *E.MEM.simulationend.(.+)*
3. Leave the **Data** field empty.
4. Click on the **Add command** button and in the **Name** field type: *C.LOGIC* and in the **Value** field enter: *calculateEndTime()*

2.5. The Order of Operations

Starting the presence simulator should proceed in the following order:

1. Upload the script (in the **Resources** tab) and configure it in the **Logic** tab.
2. Create user interface in the **Remote** tab.
3. Define timers and events.
4. Turn on the switch added in step 2.3.2.
5. Set the end time of simulation and optionally the start time of simulation using user interface.