

LCN light scenes

It can be very problematic to control the lighting in particular in big buildings. It is especially noticeable when you want to control many lights at the same time. In such cases the light scenes implemented in the LCN modules can be helpful as they enable to save certain settings and call them later on. However, the light scenes in the LCN modules have an important disadvantage – in order to reconfigure it you need access to the LCN-Pro software and some expert knowledge so it is simply impossible for the system's users to make changes.

Therefore we decided to make it easier and we created a script which enables the user to reconfigure scenes using **DOMIQ** user interfaces: **DOMIQ/Remote**, **AirDisplay**, visualization via Internet browser or LCN wall buttons.

1. Script configuration

1.1. General information and creating light scene groups

The mentioned script enables to create an unlimited number of the light scenes as well as choosing which lights are to be controlled with a certain light scene. Each scene has two actions: save and call. The script is attached to the **DOMIQ/Base** module software (**Resources** > **Scripts**). It is also available on the website of the tutorial as an attachment. If there is no script in your module, you can add it manually in **Resources** > **Scripts**.

In order to start the configuration, import the script in the **Logic** tab. For this purpose enter the following code line in the **Logic** tab: `import 'lcnscenes'`. The next step is defining the variable by which you will refer to the main function of the script (and indirectly to the light scene group). Place the following code line in the **Logic** tab:

`<variable_name>=scenes()`, where the expression `<variable_name>` should be replaced with any other name (without spaces). In this case we used the name `test` and we will use it till the end of this tutorial. So the variable declaration has the following form:
`test = scenes()`

You can give the variables any names but consider the declaration of the above variable as a group of light scenes. E.g. when you give the variable the name `ground floor`, you should be planning to create a group of scenes which will control the lights on the ground floor and to mention only the modules which control lights on the ground floor.

The above actions should be repeated for all groups of scenes which you are going to use in the installation. In this case we will limit it to one variable. When you give the variable the name `bedroom` you should create a group of light scenes which will be limited just to the bedroom – the scenes will include the modules and circuits which control the lights in the bedroom etc.

1.2. Defining light scenes

The next step is defining the light scenes within one group. For this purpose use the `create` function implemented inside the script. The general syntax of the `create` function is as follows:

`create(number, register, {devices_table})`, where:

- `number` is the number of the light scene in the LCN module. Possible values from 1 to 10.
- `register` – number of the register in the LCN module in which a given scene is to be saved. Possible values from 0 to 9.
- `devices_table` – here enter the identifiers of the LCN modules. Mark which outputs are to take part in a light scene. Dimmable outputs and relays are marked according to the syntax of the `LCN.scenes` identifier which was described in the documentation of the **Base** module. An example of the table can be seen below. **ATTENTION: the values must be passed in pairs – device/group identifier and then outputs.**

An example of the scene definition has the following form:

```
test:create(1,1,{'0.11','outputs:1100','0.12','relays:----11--',  
'0g10','outputs:0100'})
```

The above definition creates scene number 1 in the register number 1. The following elements will be assigned to the scene: dimmable outputs 1 and 2 in the module 11, relays 5 and 6 in the module 12 and the dimmable output 2 in the modules which belong to group 10.

The above definition should be repeated for all light scenes within a given group of scenes.

Below we present a definition block of the scenes from an existing installation:

```
test:create(1,1,{'0.121','outputs:1100','0.111','out-  
puts:1100','0.113','outputs:1000','0.121','relays:----11--'})  
test:create(2,1,{'0.121','outputs:1100','0.111','out-  
puts:1100','0.113','outputs:1000','0.121','relays:----11--'})  
test:create(3,1,{'0.112','outputs:1100','0.113','out-  
puts:0100','0.121','relays:--1-----'})  
test:create(4,1,{'0.112','outputs:1100','0.113','out-  
puts:0100','0.121','relays:--1-----'})
```

2. User interface

Having defined the light scenes you can configure user interface. We will show how to make elements controlling the script via menu of the **DOMIQ/Remote** application, visualization or **LCN** wall buttons.

2.1. DOMIQ/Remote

Before starting the appropriate configuration procedure of the interface controlling scenes, you have to create the structure of application (at least one page and section).

In the described case we will show elements like **Button**, because the elements of this kind differentiate long and short pressing. The action for a short pressing is loading of the scene while long pressing causes its saving.

In the described case we show how to edit the controller, which will load and save the scene number 1. The procedure is as follows:

1. Having added the element to the interface structure, double click on it in order to open the edit window.
2. In the **Label** field enter the name of the light scene.
3. Firstly define the action loading a given scene. In the **Hit** tab click on **Add command** and then in the new window enter `C.LOGIC`. In the **Value** field enter function loading the scene with the following syntax `<variable_name>:load(<scene_number>)`, in this case: `test:load(1)`.
4. In the edit mode of the **Button** element, go to **Hold**, in order to define the command which will save the light scene. Similarly to the previous point click on **Add command**. In the new window in the **Name** field enter: `C.LOGIC` and in the **Value** field enter calling the function which saves the scene according to the following syntax: `<<variable_name>:save(<scene_number>)`, in this case: `test:save(1)`.

The above actions need to be repeated for the other scenes (or groups of scenes) which were created in the script.

2.2. Visualization

In case of visualization the configuration is similar, but the buttons do not differentiate long and short pressing so you have to add two elements like **Button** to each scene – one to save the scene, the other to load it. The configuration of the button is as follows:

1. In the **Label** field enter the description of the button.
2. In the **Command** field enter the command which calls the function in the script. For loading enter: `LOGIC=<variable_name>:load(<scene_number>)`, in this case: `LOGIC=test:load(1)`. The action for saving is similar: `LOGIC=<variable_name>:save(<scene_number>)`, in this case: `LOGIC=test:save(1)`

2.3. LCN

In this section we present how to combine LCN wall buttons with the script supporting light scenes. It is the same as in case of the interface in the **Remote** application – short pressing calls the scene while the long one saves it. In order to combine the script with the wall buttons it is necessary to make the **Base** module "aware" that a given button

was pressed. In order to achieve this, assign in all the LCN modules the commands "Send keys" to the address of the **Base** module. In the **Base** module define the events which will react to these commands. Define a separate event for each light scene and the script will automatically differentiate if the pressing was long or short.

The definition of an event is as follows:

1. Add a new event in the **Events** tab.
2. In the **Channel** field enter: `E.LCN.key.S.M.TP=action`, where S – stands for the segment number, M – module number, T – keys board in the LCN module, P – key number, `action` – action name. If the scene is assigned to e.g. module number 10 and A1 key, enter `E.LCN.key.0.10.A1` in the **Channel** field: `E.LCN.key.0.10.A1`. Leave the **Data** field empty - the script detects the typ of pressing automatically.
3. In the **Actions** sections click on **Add command**. In the **Name** field enter: `C.LOGIC`, and in the **Value** field: `actionRemap("$D0", "variable_name", scene_number)`, The expression `variable_name` should be replaced with the name of another variable to which a certain scene is assigned. Instead of `scene_number` enter the number of the scene you want to load/save. Argument `"$D0"` guarantees that the function will receive the type of pressing (short or long). In this case it will have the following form: `actionRemap("$D0", "test", 1)`.