Heating under control

Heating-related expenses constitute a substantial part of the total building/flat upkeep. Through this self-help guide, we want to offer you a solution which will allow you to reduce your heating expenses and make heating control significantly more comfortable and suited to the needs of the household members.

The main goal of this self-help guide is, beside offering a more convenient and economical heating control is to show how with very little effort the rather complex functionality can be realised, using the DOMIQ/Base module.

The latter part of the self-help guide delineates the operating conception of the temperature control software and presents the implementation of the software in DOMIQ user interfaces. In addition the heating automation process with the use of temporal rules and events has been described.

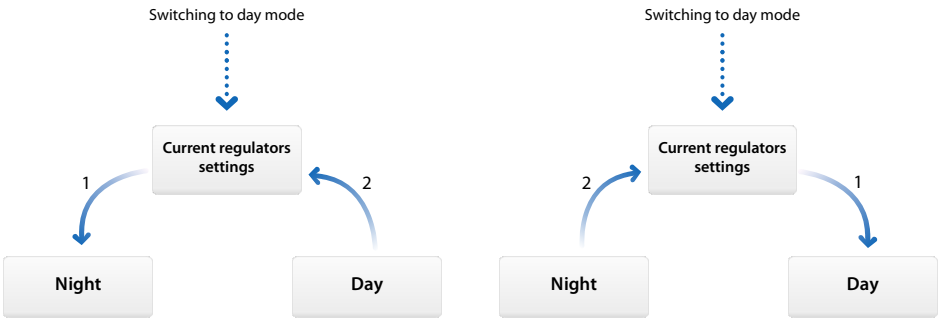
1. The Concept

The temperature control software enables switching between predefined heating operation modes, e.g., *day/night* or *warm/normal/economic*, etc. Each of the predefined modes can be freely changed during everyday use of the building automation system. Changes introduced into the settings for a given mode will be automatically saved by the DOMIQ system when switching the heating operation mode.

The programme also enables easy modification of the list of devices participating in the temperature control process. The primary goal of the temperature control software was to create an algorithm which would adjust to the changing needs of the household members and allow them to modify the house automation system settings from the user interface level.

2. Operating principle and source code

The operating principle of the programme is presented in the diagrams below. They illustrate switching from the day mode to the night mode and vice versa. The first diagram presents switching from the day mode to the night mode:



When the **Day** mode is enabled, the current regulator settings are saved as default settings for the **Night** mode (1). Next, the default regulator values assigned to the **Day** mode are sent to the regulators (2).

In case of switching from the day mode to the night mode, the changes are conducted in an analogous manner.

The source code of the temperature control software is available as an appendix to this guide, titled `switcher.lua`. The file can be downloaded from the webpage with the self-help guide description, at **www.domiq.pl**.

Save the file in the **DOMIQ/Base** module > **Resources** > **Scripts**. Next, activate it in the **Logic** tab with the command `import 'switcher'`.

Also in the **Logic** tab, add code fragments configuring the programme for operation with particular LCN regulators. An example of the configuration code is presented below:

```
-- user-defined code
temperature=switcher("temp")
-- add a regulator No. 1 in the LCN module with the address No. 26.
temperature:add 'LCN.regulator.0.26.1'
-- other regulators are added analogously
temperature:add 'LCN.regulator.0.26.2'
temperature:add 'LCN.regulator.0.36.1'
temperature:add 'LCN.regulator.0.36.2'
```

A few words of explanation of the user-definable code:

The variable `temperature` contains a reference to the object created by activating the `switcher(name)` function. The variable can be given any name (without spaces). The argument of the `switcher` function. (in this case, `"temp"`) must be a unique sequence of characters.

In the event we want to use the script to control multiple sets of devices or several rooms, it is important to provide unique names to the variables referring to objects created by calling the `switcher` function, and also to call the `switcher` function with a unique argument. The examples presented in the latter part of this guide show how to control a single set of devices.

The code lines `variable_name:add '<device id>'`, in this case, e.g. `temperature:add 'LCN.regulator.0.26.1'` are used in declaring LCN regulators used in temperature control.

For the curious

If this much information about the operation of the `switcher.lua` script is enough for you, skip this section. Otherwise, keep reading the following part of this section.

The first function of the script – `par:add(dev)` is used for adding new regulators. It reads identifier names, e.g. `LCN.regulator.0.26.1` and saves them in table `d`.

The main functionality of the script is executed inside the function `par:switch(what)`, and more precisely, in the `for` loop which it contains. At the moment of switching of heating modes, two primary actions are executed in the loop:

- Current regulator settings are saved to non-volatile variables MEM. These settings are at the same time saved as default values for the switched heating mode. This is represented by the line:
  `set("MEM.sw."..name.."."..current.."."..d.channel(),d.value).`
- Reading from the MEM variables of regulator settings assigned to the heating mode being activated `local val=get("MEM.sw."..name.."."..what.."."..d.channel())`, then selecting these values as the current regulator settings: `d:set(val).`

The line `set("MEM.sw."..name,what)` saves the current heating operating mode into a MEM variable.

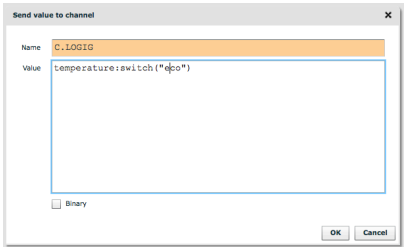
3. Implementation in user interfaces

In this section, we will present the implementation of the temperature control programme in the user interfaces of the **DOMIQ** system. The examples presented further into the section refer to controlling four LCN regulators in three operating modes: *Eco, Day, Night*. The number of modes and their names, as well as the number of devices controlled, can be freely modified by the person configuring **DOMIQ/Base**.

3.1. DOMIQ/Remote

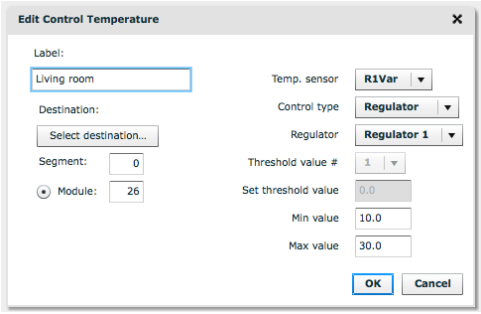
Using the `switcher.lua` script in combination with the application **DOMIQ/Remote** allows to create a very convenient building heating control interface. In order to create such an interface, perform the following actions:

1. Select the **Remote** tab.
2. Add a new **page** or **section** or only a **section**, if you are using an existing **page**. Assign a label to the page/section, e.g. *Temperature*.
3. Add a **Button group** and select the number of columns, depending on how you wish to group the buttons.
4. Add a **Button** and double-click it, in order to modify its properties:
   - Enter a brief description of the button in the field **Label**. In this case: *Eco*.
   - Click the **Add channel...** button, in the window which will appear, in the field Name, enter: `C.LOGIC`, and next to the field, the value: `temperature:switch("eco").`



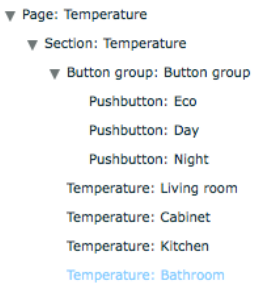
5. Repeat Point 3. for the remaining two buttons, changing the labels and content of the **Values** field to, respectively: `temperature:switch("day")`, `temperature:switch("night")`.Pressing of the added buttons will activate their assigned heating operation modes.

6. Add **Temperature** elements in the number matching the declaration in the source code and define their parameters.



7. At the end, add the **Text** element, which will display the current heating operation mode. Double-click it and fill in the properties:
   - In the **Label** field enter the description, e.g *Current mode*.
   - In the **Channel** field, enter the name of the MEM variable in which information about the current operating mode is stored. The general syntax is as follows: `MEM.sw.<nazwa_argumentu_funkcji_switcher>`. In this case `MEM.sw.temp`.

The final structure in the **Remote** tab should look as follows:



The result in **DOMIQ/Remote**:

Configuration of heating operation modes in DOMIQ/Remote

With the user interface built, the heating modes can now be configured:

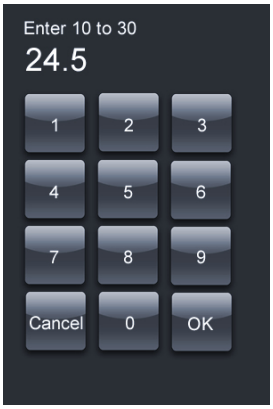
1. Select the mode to configure, e.g., **Eco**.
2. Set the temperature values in the regulators with the **Temperature** controls.
3. Click the next button, in order to begin the configuration of the settings for the next mode. After pressing the settings of the previous mode, they will be automatically saved to the **Base** module memory.
4. Perform analogous actions for the two other modes.

Your configuration is fully functional! Now you can switch at will the heating modes in **DOMIQ/Remote**.

Remember! If you modify regulator settings during operation in a given mode, e.g., **Eco** and then select another mode, the changed settings will be saved to the **DOMIQ/Base** module memory as default settings of the **Eco** module.

3.2. DOMIQ/Display

In this subsection, we will show you how to create a temperature control screen in the **DOMIQ/Display** panel. Here, we will also use the `switcher.lua` script discussed in Section 2, as well as another function of the **Temperature** element, which enables the introduction of numerical values with a virtual keypad.



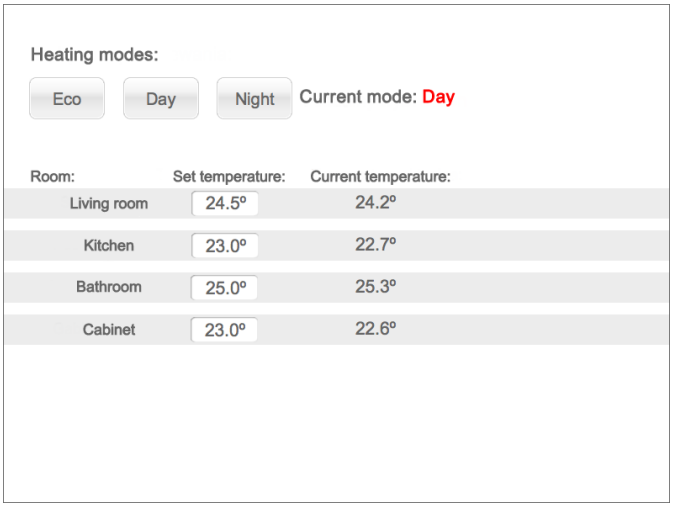
The new function is available in the **DOMIQ/Base** module software, since version 1.7.4 and at present it is available only in **DOMIQ/Display** panels with the software version 1.7.0.

In order to define the visualisation screen to control the temperature, follow the instructions below:

1. Select the **Display** tab.
2. Click the **Add screen** button.
3. Configure the screen settings: enter a unique identifier, you can optionally add a visualisation background.
4. Add a **Button**, then click it and fill in its properties:
   - Enter a description in the **Label**, field, in this case *Eco*.
   - In the **Command** field, enter: `LOGIC=temperatura:switch(”eco”).`
5. Repeat Point 4. for the other two buttons, changing the labels and the values of the **Command** field to: `LOGIC=temperatura:switch(”day”), LOGIC=temperatura:switch(”night”).` Pressing the two added buttons will activate the corresponding heating operation mode.

6. Add two **Text** elements. In the first **Text** field, enter the description, e.g.: *Current mode:*. In the second one, in the **Channel** field, enter: `MEM.sw.<argument_name_of_the_switcher_function>`, in this case `MEM.sw.temp`.
7. Add the element **Temperature** and fill in its properties:
   - In the **Channel** field enter the LCN regulator address, in this case `LCN.regulator.0.26.1.`
   - In the field **Format** enter: *0.0*.
   - Optionally, you can change the font colour and size. In this case, the colour: *#666666*, size *21*.
   - In the fields **min** and **max** enter the lower and upper temperature regulation range limits, e.g. *10* and *30*.
8. Repeat Step 7. for the other regulators.
9. If you wish to display the current temperature next to the temperature value, add another **Temperature** element and fill it in as in Point 7. Remember to change the value in the **Channel** field to the address of the corresponding LCN temperature sensor. In this case, `LCN.value.0.26.r1`.

An exemplary temperature control screen:



The field descriptions and the graphic composition can be freely modified. All of the graphics elements and the configuration file of the visualisation screen are attached to the guide as an appendix. Copy the content of the attached file `display.xml` to the file `display.xml` in your **DOMIQ/Base** module.

Configuration of heating operation modes in the DOMIQ/Display panel

This configuration is carried out in a similar fashion as in the case of **DOMIQ/Remote**.

1. Select the mode to configure, e.g., **Eco**.
2. Set the temperature values in the regulators – click the **Temperature** elements and enter the values with the numerical keypad.
3. Click the next button, in order to begin the configuration of the settings for the next mode. After pressing the settings of the previous mode, they will be automatically saved to the **Base** module.
4. Perform analogous actions for the two other modes.

Your configuration is fully functional! Now you can switch at will the heating modes with the **DOMIQ/Display** panel.

Remember! If you modify regulator settings during operation in a given mode, e.g., **Eco** and then select another mode, the changed settings will be saved to the **DOMIQ/Base** module memory as default settings of the **Eco** mode.

4. Automation

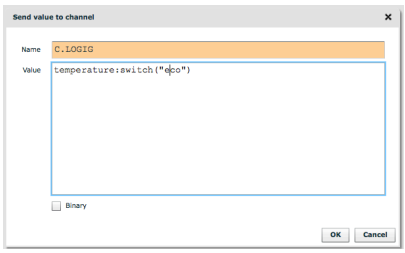
Temperature control will be even more convenient and more precise, if it is integrated with events and timers. This will make temperatures in the rooms adjusted to the preferences and habits of the residents. The rest of this section presents an example of integrating a temperature control programme with a Satel alarm system, LCN devices with using events and timers.

4.1. Integration with an alarm control panel

Implementation of events `E.IDS.armed.<zone_number>` allows you to conveniently connect temperature control with a SATEL alarm system. The operating principle is very simple: after detecting an event, the **DOMIQ** system sets a specified heating operation mode, in this case the **Eco** mode.

Defining such a reaction of the automation system is done in the following manner:

1. Select the **Events** tab and add a new event.
2. In the **Details** section:
   - In the **Channel** field, enter: `E.IDS.armed.(%d+)` – checks the arming of any alarm zone.
   - In the **Data** field, enter: *1*.
3. In the **Actions** section, click **Add channel...** .
4. In the window which will pop up, in the **Name** field, enter: `C.LOGIC`, and in the **Value** field: `<variable_name>:switch(”<mode_name>”)`, in this case: `temperatura:switch("eco")`.



If you want for the reaction to occur after arming a particular alarm zone, e.g., zone no. 1, enter: : `E.IDS.armed.1`. in the **Channel** field.

Finally, the event definition screen should look like this:

4.2. Integration with the LCN system

If your system does not feature a SATEL alarm system, a similar effect can be achieved using the LCN keys and the event `E.LCN.key`, which informs about pressing of buttons in the LCN system. Analogously to the first example, after detecting this event, the **DOMIQ** system will set the heating mode assigned to that button.

For the needs of this example, let us assume that the first three buttons in Table **B** with the command **Long** are used for changing the heating modes. To do this, select the **Send keys**, command for the individual keys in the LCN-Pro software, and set the **DOMIQ/Base** module as the target (default ID: 254). With the options of the LCN module configured in this manner, you can begin to define events:

1. Add a new event.
2. In the **Channel** field, enter: `E.LCN.key`
3. In the **Value** field, enter: `B1 make`.
4. Next click **Add channel...** .
5. In the window which will be displayed in the **Name** field, enter `C.LOGIC`, and in the **Value**: `<variable_name>:switch(”<mode_name>”)`, in this case: `temperature:switch("eco")`.
6. For the remaining heating operation modes, repeat steps **1** through **5**, remembering to change the key number and the action executed by pressing it, in this case: `temperature:switch("day")` and `temperature:switch("night")`.

4.3. Using timers

Using temporal rules allows to automatically change heating modes depending on the time of day, day of the week, etc. Further in this subsection are instructions on how to create new **DOMIQ/Remote** elements which enable the changing of activation times of individual heating modes, and the procedure of traditional (in the configurator) defining of temporal rules.

4.3.1. Timer control in DOMIQ/Remote.

In order to realise this functionality, use the **Timers** and the **Remote** tab:

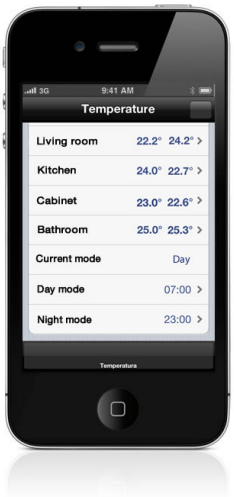
1. Add a timer group and name it, e.g., *Temperature*.
2. Add a new timer and fill in its properties:
   - In the **Label** field, enter its description, depending on the timer's intended function.
   - Fill in the attributes in the **Details** section:
     - In the **Hour** field, enter the name of a MEM variable containing the name of the variable referring to the `switcher` function and the name o the mode. In this case `MEM.hour.temperature.day`.

- In the **Minute** field, enter: `MEM.minute.temperature.day`.
- In the **Days of the week** field, enter: `MEM.temperature.days`.

This declares the non-volatile MEM variables in which will be stored values set in the **DOMIQ/Remote**.

3. In the **Command** section define the command which will be executed by activating the timer:
   - Click **Add channel...**
   - In the window which will pop up, in the **Name** field, enter `C.LOGIC`, and in the **Value** field: `<variable_name>:switch("<mode_name>")`, in this case: `temperature:switch("day")`.
4. Click **Save**, in order to confirm the changes.
5. Go to **Remote** tab.
6. Add a **Time** element.
7. Double-click it and fill in its properties:
   - In the **Label** field, enter the control description.
   - In the field **Channel** with hours, enter: `MEM.hour.temperature.day`.
   - In the field **Channel** z minutą, enter: `MEM.minute.temperature.day`
8. Repeat points 2 through 7 for the night heating mode.

The result in **DOMIQ/Remote**:



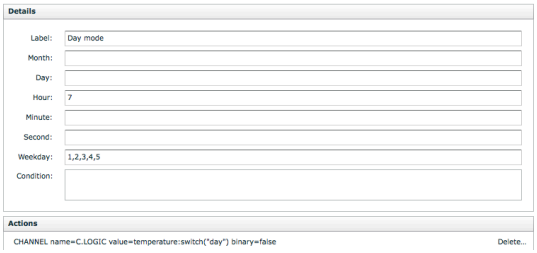
4.3.2. Defining timers in the configurator

Defining timers directly in the configurator has one disadvantage - the user is unable to update settings. However, despite this drawback, this solution has one big advantage – greater flexibility of defined rules. Namely, you can very quickly define separate rules for individual weekdays, others for the weekend, etc. Such a solution is also available in **DOMIQ/Remote** though it would entail defining multiple separate user interface elements (elements such as **Time** and **Selection**), which would severely reduce the usability and clarity of the entire temperature control interface.

As an example, we present defining independent timers controlling heating for weekdays and the weekend.

We will start with defining temporal rules for weekdays.

1. Add a timer group and call it, e.g. *Heating*.
2. Add a new timer. In the **Label** field, enter its description, e.g. *Day mode*.
3. In the **Hour** field,enter the hours of this timer's activation, e.g., *7*. Fill in the **Minute** field field in an analogous manner.
4. In the **Weekday** field, enter *1,2,3,4,5* – weekdays.
5. In the **Actions** section, click **Add channel...** . In the window which will pop up, in the **Name** enter `C.LOGIC`, and in the **Value** field: `<variable_name>:switch(”<mode_name>”)`, in this case: `temperature:switch("day")`.



6. Add another timer and perform analogous actions for the night mode.

This way, we have defined timers for weekdays. Defining the weekend is done analogously. The only things which change are the timer activation time, the content of the **Weekday** field – enter *6,7*

The above method can be used to analogously define separate rules for the individual days of the week.