

Many Base modules in one installation

In this tutorial, we will present the concept of links and show you how to combine multiple **Base** modules in one installation using links and groups. You will also learn how and why to use many **Base** modules in one building.

1. What are links?

In software version 1.7.5 the functionality of the **Base** modules has been developed. It enables to define direct connections between modules on a point-to-point basis. Such a single connection is called **link**. Thanks to links the **Base** modules can exchange information between each other and react to events. This opens the way for implementation of the functionalities which were not available before.

The **Base** modules can be connected using local network or Internet. **In case of Internet connections it is necessary to open the UDP 4554 port.**

The modules can also synchronize groups of variables between multiple modules. This function is not described in this tutorial, because it requires installation of all modules in one IP subnet which may not be possible in this scenario.

2. Where to use links?

The links were designed for installations in which there are at least two or more **Base** modules and based on them an integrated automation network will be created. The links can be particularly used between devices with any IP addresses.

2.1. DOMIQ network in an apartment building

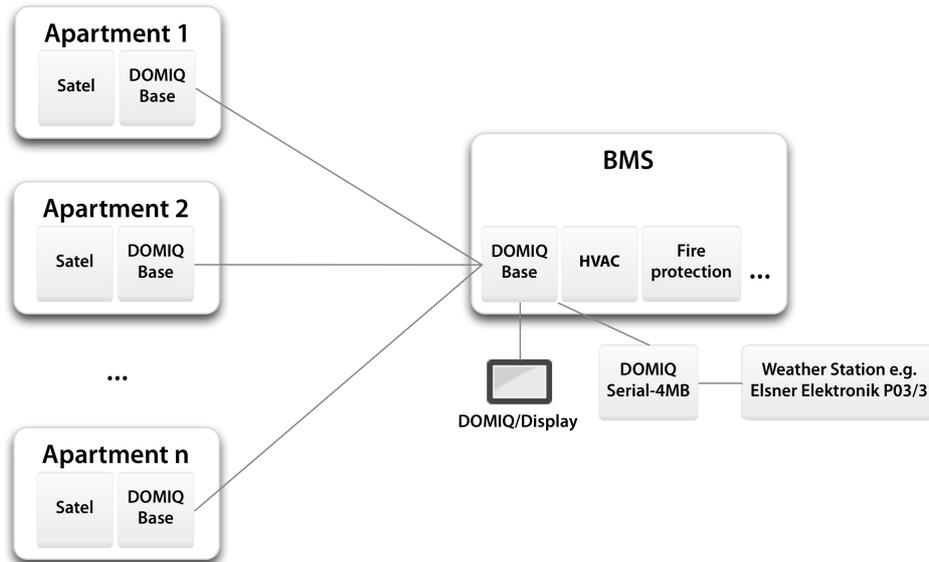
A perfect area of application of the **DOMIQ** network are buildings with many apartments. In such place the **DOMIQ** system can supervise, watch over the safety, warn and provide the residents with useful information.

Imagine that in each apartment there are: the **Base** module and the **Satel** alarm system with safety sensors (e.g. smoke detectors, flooding sensor, sensor of breaking the window ect.). In the building we installed an additional **Base** module working as a master module over the modules installed in apartments. We created links between the master module and all the **Base** modules installed in apartments. Reception of the building was also equipped with a **DOMIQ/Display** touchscreen, which is designed to serve as a supervisory panel. In the case of any emergency situation (e.g. fire, flooding, burglary ec.) in any apartment, the security receives an immediate notification to the touch panel concerning the exact location of alarm and its cause.

Moreover, using the master module and touch panel you can monitor the entire building. In case of adding an additional weather station, you can provide current weather information to all residents of the building.

Additional plus of this solution is the fact that you can use standard functions of the **Base** modules in each apartment in order to integrate it with the existing automation system.

The structure of the configured system is presented below:



If for some reasons it is impossible to install **Satel** alarm system, you can use security sensors connected to the **LCN** binary inputs or the ones which communicate using MOD-BUS protocol and connect them directly to the **DOMIQ/Serial-4MB** module.

The links enable to create intelligent control system of the entire building, including independence of all its parts (apartments, storeys etc.)

3. DOMIQ/Base configuration

In 1.7.5 version in the configuration interface of the **Base** module we added the **Links** tab. It enables to define point-to-point connections between the **Base** modules and connections which enable using the **Base** module as a segment coupler of the LCN network. In the tutorial "Base as segment coupler" we described how to use **Base** as a segment coupler.

The **Links** tab is presented below:

Address	Name	Type	Version	Identity
192.168.10.20	Base	DOMIQBase	1.8.4-3-140411	084a3a03e0b0a714f510b036cfa603
192.168.10.110	DOMIQ	DOMIQBase	1.8.4-2	0584f4f7c3838e313a430417c4e89407
192.168.10.121	DOMIQ	DOMIQBase	1.8.4-2	44780a0f920a9f6e4b70304ab92108
192.168.10.115	115	DOMIQBase	1.8.4-2	4c0561a3913a0b4930a21237157a8163
192.168.10.117	DOMIQ	DOMIQBase	1.8.4-2	20a7055154ef44a27a3818346d555
192.168.10.104	DOMIQ	DOMIQBase	1.8.4-2	024439a6a30a9d085c1723a4655c45
192.168.10.108	108	DOMIQBase	1.8.4-2	07f78a04908e4814c1f650a3c0c005
192.168.10.120	DOMIQ	DOMIQBase	1.8.4-2	0c2a4743b69778a06529f1874b50b281
192.168.10.107	DOMIQ	DOMIQBase	1.8.4-2	a7da3013770c1b666d0f2f4f4e5026

Structure of this tab is very similar to **Events** and **Timers**. There is **Structure** on left side of the window. The defined links are displayed in this section.

The right part of the window contains the **Details** section, in which the connection parameters are configured.

Details

Name

Description

Address

Port

Secret

Identity

Name: The name of the **Base** module, with which the connection is to be established. It can be any name without spaces. The name is not connected with the name of the module given in **Settings**.

Description: A short description of the connection.

Address: The IP address of the **Base** module, with which the connection is to be established.

Port: Fill in this field if the connection with the **Base** module was redirected to a custom port.

Password: If this field is filled in, the data sent between the modules will be encrypted. Otherwise the encryption is not active.

Identifier: Every **Base** module has its own identifier. It additionally secures the data transmission between the **Base** modules. It prevents from using the IP address assigned to the **Base** module by another network device.

In the **Details** section mentioned below there is a list of all the **Base** modules available in the network together with identifiers.

3.1. Example configuration of the links

In order to present an example configuration we will use the idea of a building with many apartments described in the previous chapter. We will only show a single link between an apartment and a master module (another links can be configured similarly).

Configuration of a single link is as follows:

Slave module

1. Log in into the configuration interface of the **Base** module installed in the apartment and select the **Links** tab.
2. Add a new connection and fill in its properties:
 - In the **Name** field enter the name of the master module, e.g. *reception*.
 - In the **Address** field enter the IP address of the master module.
 - If the connection should be encrypted, fill in the **Password** field.
 - You can fill in the fields **Identifier**, **Description** and **Port** (optional).

Master module

Repeat steps 1 and 2. Remember to change the IP address to the one of the slave module and change its name, e.g. *apartment1*. We recommend to use names which are connected with the numbers of apartments.

3. The configuration is ready. Make connections between the apartments and the master module.

3.2. Identifiers

As we mentioned above, the links enable to send commands or react to events from other **Base** modules. The special network identifiers were reserved for this purpose. Syntax of the identifiers with examples was shown in the tables below. The presented examples refer to the building with many apartments. You can create any commands or events analogically.

	Identifier	Value	Description
Command	NC . M . K M - Base module, to which the command is addressed. K - Command	any	Any network command
Event	NE . M . K M - Name of the Base module, in which an event appeared.	any	Any network event



- `NC.reception.C.VAR.flooding.apartment1=1`
Setting the variable `VAR.flooding.apartment1` to the value of 1 in the master module *reception*
- `NE.apartment1.E.IDS.alarm.1=1`
The master **Base** module receives such information regarding alarm in zone 1 in the apartment to which the module *apartment1* is assigned. The master module can take any reaction to this event, e.g. display a certain visualization screen on the operator panel, switch on the alarm siren etc.
- `NE.(%w+).E.IDS.input.10=1`
An event which informs about value change of the alarm input number 10 in any apartment. The input can be e.g. fire sensor. Any reaction to this event can be taken by the master module similarly to the example below.

3.3. Automatic communication of the Base modules

If the connections with all apartments are ready, you can configure downloading the information from the **Satel** system. The data will be downloaded and passed automatically to the master **Base** module in case of alarm situation.

We recommend to program the alarm system in such a way, that the sensors responsible for detection of a certain emergency event (fire, flooding etc.) set the value of any unused output of the alarm system. Changes of this output will serve to invoke events in the master **Base** module. It will make further configuration of events easier.

The example presented in the next part of this section shows how to define events in case of flooding, fire and burglary. The numbers of outputs and inputs have a demonstrative character. The example concerns a single apartment. The slave module installed in this apartment is called *apartment1*.

3.3.1. Configuration of the slave module

In case of the slave module define the reaction to changes of the outputs connected with security sensors. In this case it is sending network commands to the master module. You can also define the local reactions to alarm situations.

Flooding

1. Add a new event.
2. In the **Description** field enter a short description of an event, e.g. *Flooding*.
3. In the **Channel** field enter `E.IDS.output.<output_number>`, where `<output_number>` stands for the output whose value is set in case of value change of any flooding sensors, e.g. `E.IDS.output.15`
4. Leave the **Value** field empty. As a result the change from 0 to 1 and the other way round will be detected.
5. Add a command and then in the displayed window in the **Name** field enter: `NC.<master_module>.C.VAR.flooding.<slave_module>`
e.g.:
`NC.reception.C.VAR.flooding.apartment1`
In the **Value** field enter: `$D0`.

Details

Enabled

Description Flooding

Channel E.IDS.output.15

Data

Condition

Actions

CHANNEL name=[38] value=\$D0 binary=false

Add LCN... Add VAR... Add Channel...

Send value to channel

Name NC.reception.C.VAR.flooding.apartment1

Value \$D0

Binary

OK Cancel

Fire

1. Add another event.
2. In the **Description** field enter a short description of an event, e.g. *Fire*.
3. In the **Channel** field enter `E.IDS.output.<output_number>`, where `<output_number>` stands for the output whose value is set in case of value change of any fire sensors, e.g. `E.IDS.output.16`
4. Leave the **Value** field empty.
5. Add a command and then in the displayed window in the **Name** field enter: `NC.<master_module>.C.VAR.flooding.<slave_module>`
e.g.:
`NC.reception.C.VAR.fire.apartment1`
In the **Value** field enter: `$D0`.

Burglary

1. Add another event.
2. In the **Description** field enter a short description of an event, e.g. *Burglary*.
3. In the **Channel** field enter `E.IDS.alarm. (%d+)`. The event will be called in case of alarm in any zone.
4. Leave the **Value** field empty.
5. Add a command and then in the displayed window in the **Name** field enter: `NC.<master_module>.C.VAR.alarm.<slave_module>`
e.g.:
`NC.reception.C.VAR.alarm.apartment1`
In the **Value** field enter: `$D0`.

This procedure should be repeated for every single apartment. Remember to change the numbers and names of the apartments.

3.3.2. Configuration of the master module

For the master module configure the rules which will be performed if the alarm notification is received. Configure the visualization in which the information about alarms will be displayed.

Visualization configuration

The visualization can contain simple interface in which the state of the alarm in the apartments will be shown in form of control lamps, e.g.:

Apartment No.	Flooding	Fire	Burglary
1	●	●	●
2	●	●	●
3	●	●	●
4	●	●	●
5	●	●	●
6	●	●	●
7	●	●	●
8	●	●	●
9	●	●	●
10	●	●	●

In **Channel** field of the control lamps enter the names of the VAR variables which you also used during configuration of the slave modules. In this case e.g. `VAR.flooding.apartment1`, `VAR.fire.apartment1`, `VAR.alarm.apartment1` itd.

In order to display the current states of the VAR variables from the slave modules on the visualization, define an event which will download such values. The definition is as follows:

1. Add an event.
2. In the **Channel** field enter: `NE.(.+).C.VAR.(.+)` – as a result the master module state will not store any values of VAR variables from the slave modules. Leave the **Value** field empty.

3. Add a command. In the displayed window in the **Name** field enter: `C.VAR.$C2`. In the **Value** field enter: `$D0`.

Details	
Enabled	<input checked="" type="checkbox"/>
Description	
Channel	NE.(.+).C.VAR.(.+)
Data	
Condition	

Actions	
CHANNEL name=C.VAR.\$C2 value=\$D0 binary=false	Delete...

Events configuration

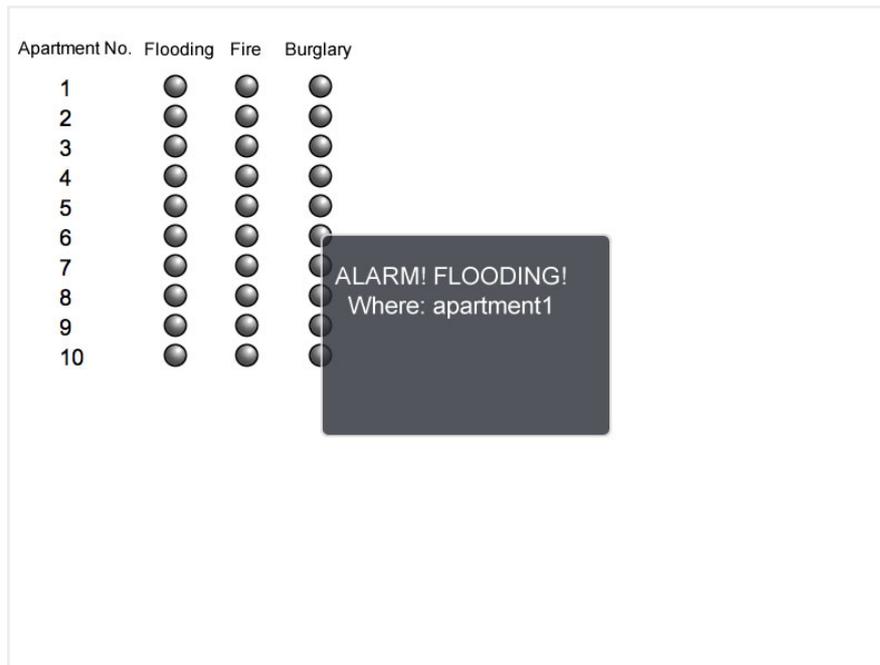
In this case, similarly as for the master module, define three events, one for each alarm type. In this case the reaction to the alarm will be notification displayed on the **Display** screen and change of the controller colour on the visualization.

Flooding

1. Add a new event.
2. In the **Description** field enter a short description of the event, e.g. *Flooding*.
3. In the **Channel** field enter `E.VAR.flooding.(.+)` – the event will be invoked in case of alarm in any apartment.
4. In the **Value** field enter `1`.
5. Add a command and then in the displayed window in the **Name** field enter e.g.: `C.DISPLAY.message=ALARM!FLOODING! Where: $C1`.

You can similarly define the event for fire and burglary. Remember to change the **Channel** field to `E.VAR.fire.(.+)` and `E.VAR.alarm.(.+)` as well as the message in the **Value** field.

The result is as follows:



4. Integration of the weather station

In the previous tutorial "MODBUS weather station" we presented the integration of the **Elsner Elektronik P03/3** weather station. In the next part of this chapter we show how to use this station in our **DOMIQ** structural network and provide all apartments with the weather information.

The weather station can be connected via **DOMIQ/Serial-4MB** module to the master **Base** module. Thanks to the links created before, the weather data from the master module will be passed to modules in the apartments. Therefore the installation costs will be optimized and its functionality will not be reduced.

4.1. Configuration of the master module

In order to configure the master module, import the configuration file of the weather station and export the data from the weather station to the slave modules.

Import of the weather station configuration

To the tutorial we attached the file `elsner_p03.xml` with the configuration of the weather station which is ready to import. Import the file in the **MODBUS** tab:

- Click on the interface and then **Import...**
- Select the downloaded file `elsner_p03.xml` from the hard drive. After correct import the device will appear on the list.
- Fill in the fields **Name** (without spaces) and **Address** (according to microswitches on the board of the weather station). In this case we chose the address `1` and name `meteo`.

Download and export the measure data

Data from the weather station will be stored and exported cyclically according to the set timer. In this case: every minute.

The procedure will be started from the event definition which will be responsible for storing weather data:

1. In the **Events** tab add a new event and fill in its properties:

- Enter a short description, e.g. *Downloading the weather information.*
- In the **Channel** field enter: `E.MODBUS.<interface>.<device>.(%w+)`, e.g. `E.MODBUS.int.meteo.(%w+)`
- Leave the **Value** field empty – any measure data will be captured.
- Add a command and in the displayed window in the **Name** field enter: `C.LOGIC`, and in the **Value** field: `tab["$C1"]="$D0"`

2. Go to the **Logic** tab and paste the following code:

```
-- Table of the weather data from the station
tab={}

-- Function sends the received measure data to all
-- slave modules and stores them in the VAR variables.
-- The function must be periodically invoked via Timers
function export()
    for i=1,10 to --the second parameter determines the number of
apartments
        for k,v in pairs(tab) do
            command("NC.apartment"..and.."C.VAR.weather"..k,v)
        end
    end
end

end
```

Adjust the second parameter of the `for` loop in the code (by default 10), which determines the number of apartments the data from the weather station will be sent to. If you used another links names, modify the command beginning from „NC.“.

3. Define the timer which will invoke the `export` function cyclically and send the measure data to slave modules. Add a new timer (**Timers** tab):

- In the field **Second** enter `0` – timer will be invoked every second.
- Add a command and in the displayed window in the **Name** field enter: `C.LOGIC` and in the **Value** field: `export()`.

4.2. Configuration of the slave modules

Slave modules do not require any additional configuration. In the **State** tab you can check if the data were exported correctly to the slave modules. The weather data will be visible as VAR-variables with the following syntax: `VAR.weather.<name>`, e.g. `VAR.weather.temperature`.

The data can be used in different ways: they can be displayed in the visualization or used in events etc.