euLINK MODBUS - Quick Guide for Integrators

Table of contents

Skills required:

- Basic knowledge of computers and networks
- Familiarity with the FIBARO Home Center environment
- Knowledge of the basics of the operation of specific devices integrated into the euLINK gateway (including HVAC, PV, EV, etc.).
- Installation practice in the field of electronic equipment will be useful
- Ability to read diagrams

1. Where to start?

If you are an experienced MODBUS installer and have already installed the euLINK gateway, you can skip the initial steps and go directly to section 10 on page no. 40. However, if this is your first encounter with the euLINK gateway and with MODBUS technology, we encourage you to read all sections of this Step-by-Step Guide.

2. Basic building infrastructure concepts: MODBUS, BMS, HVAC, PV, EV, AC, etc.

All available literature on the infrastructures in question - including this guide - often uses abbreviations derived from the English language, the most important of which are worth explaining at the beginning of the Guide:

- MODBUS - Open communication protocol using serial links (MODBUS RTU) or TCP/IP network (MODBUS TCP)
- BMS - Building Management System
- HVAC - Heating, Ventilation, Air Conditioning or other indoor climate management
- AC - Air Conditioning, included in HVAC
- IDU - Indoor Unit - the indoor unit of an air-conditioning system (wall-mounted, ceiling-mounted, ducted, etc.), jargonised as "Split".
- ODU - Outdoor Unit - air-conditioning outdoor unit
- PV - Photovoltaics, sometimes including Energy Storage
- ES - Energy Storage, mainly electricity, sometimes also heat
- EV - Electrical Vehicles and their chargers
- HP - Heat Pumps, i.e. equipment for heating buildings and hot water, basically classified as HVAC
- RECU - Recuperators, or heat recovery air handling units, basically classified as HVAC
- DHW - Domestic Hot Water
- RES - Renewable Energy Sources, i.e. energy extraction from wind, solar radiation, waves and tides, etc.

3. Why MODBUS?

We are sometimes confronted with the question why does euLINK use the relatively old MODBUS communication protocol, when today most household appliances are equipped with WiFi and a smartphone app? Why go back to solutions based on wired communication?

It is true that many household appliances and infrastructure devices have a LAN and/or WiFi port - but mostly only to use their own app and **NOT** to integrate with a third-party BMS. As a result, householders have to install multiple smartphone/tablet apps for themselves and learn how to use each one. The worst part, however, is that all these devices do NOT communicate with each other and do NOT exchange any information with each other. Any smart home or

building can only be truly smart if it has a 'single brain' that gathers all the information from all the sensors and devices. Based on rational and well-considered algorithms, this 'brain' should make rational decisions and send them to the relevant 'actor' modules and infrastructure devices. Thanks to the simplicity of its scenes, the FIBARO Home Center controller performs excellently in the role of such a brain. However, if, instead of a single brain, a house or a building is handled by many unconnected systems, it will not really be a Smart Home, but rather a difficult case of split-self.

A universal and simple communication protocol is needed to manage all the building infrastructure equipment provided by different manufacturers. These requirements are well met by the MODBUS protocol, although it is indeed not very young (Modicon developed it in 1979). However, it is well-known and very well-documented, and owes its enormous popularity, among other things, to the lack of any licences and implementation fees. It is the second (after KNX) most commonly implemented protocol in building infrastructure devices for integration into the BMS. However, care must be taken when selecting infrastructure devices, as there is a certain category of the simplest and cheapest devices on the market that do not have any integration possibilities in any universal protocol. So, if a developer is keen on integrating infrastructure devices in the Smart Home, the communication capabilities of these devices need to be checked before purchase, otherwise it will be too late.

The MODBUS TCP protocol is becoming a trend in the integration of multi-vendor infrastructure equipment into BMS platforms. The SunSpec[1] Alliance, for example, now recommends the implementation of the MODBUS TCP protocol in all new photovoltaic inverters and energy management equipment. So we are seeing a renaissance of the MODBUS protocol - rather than its decline!

If it is not possible to use the MODBUS TCP protocol (via LAN) to communicate with the infrastructure device - then indeed a MODBUS RTU serial transmission cable should be used. However, in the vast majority of practical situations, it is sufficient to use a single-pair cable, running from the euLINK gateway to the nearest device, from it to the next and so on to the last device (a data-bus with a 'bus' topography). In a normal single-family home, this will probably be no more than 10-30m of single-pair cable. Adding these 2 wires will not set the modern, wireless home back to the 'outdated', older, wired world. Besides, there are already hundreds of metres of power, network (LAN/Internet), alarm, aerial and other cables in the walls of this modern building anyway. If for some reason running a physical cable is not possible at all - this function can also be realised wirelessly, although such a solution will be slightly more expensive. A concrete example of such a solution is described below in section 9.c) on page no. 39.

The FIBARO Home Center itself does not have too many built-in hardware communication ports and non-standard integration protocols, so this role can be fulfilled by the euLINK gateway.

4. Parameter templates for MODBUS devices

The '**template**' concept is key to understanding the communication of the euLINK gateway with other devices. The operation of each integrated device is controlled by an embedded microcomputer, which stores various configuration and operating data in its memory. Some of these values can be published by

[1] SunSpec is the standards organisation for the power distribution industry, bringing together some 180 of the largest PV equipment manufacturers, energy managers, etc.

the manufacturer for integration into external building management systems (BMS). However, the arrangement of this data in memory is by no means standardised, so each manufacturer and each device has its own order and assignment of these data to physical values. For integration into the BMS to be possible, the manufacturer must make the documentation of the arrangement of this data available to external developers. In the case of the MODBUS protocol, this documentation is in the form of a table, called the '**MODBUS Register Map**', informing the programmer in which register a particular value is stored.

Depending on the function of the device, this table can range from several to even hundreds of registers. Fortunately, not all of them are necessary for integration into the BMS. Configuration parameters are almost never needed because HVAC/PV/EV installers have specialised software tools for this. And not even all operating parameter readings are needed, as for heat pumps or air conditioners, a dozen or so readings are usually sufficient (e.g. unit status, current temperature, operating mode, fault code, etc.). In addition, users will need to issue a few simple commands (e.g. turn on/off, change temperature set point, change operating mode, etc.).

Thus, the programmer, creating a controller for this device in the BMS, must know the basic functions of the device in order to be able to select the necessary data from the MODBUS Register Map. The BMS programmer must then code the communication in the chosen protocol in the controller under construction, aiming to read a given parameter or send a command. Sounds simple, but the problem is that there are hundreds of device manufacturers, each producing dozens of models of HVAC/PV/EV equipment that change every few weeks, plus new models are constantly entering the market. So there are thousands of appliance variants, their list changes dynamically and hundreds of programmers would have to be employed to constantly update a universal BMS. And this 'race' cannot be won anyway.

Therefore, the euLINK gateway philosophy takes a different approach. We have identified a kind of 'core' which is common to many protocols, and our developers have coded the operation of this 'core' into the euLINK software. In contrast, we collect the differences between individual devices in the form of a set of parameters and specific commands , which we call a '**device template**'. The preparation of such a template requires documentation from the device manufacturer (e.g. Register Map) and a meticulous engineering approach, but no programming languages or other programming skills are needed. As a result, not only can we create new templates, but advanced smart home installers can do so too. There are thousands of such people in Europe, far more than there are programmers who could ever be employed on such a project.

To facilitate the work of Template Designers, euLINK has a unique built-in graphical editor that makes it easy to enter commands, readings and their arithmetic and logical processing into the template. Any advanced installer can build the template himself to his own specifications. He or she can also publish it in our online database euCLOUD, where the template will be checked by a moderator, translated into 15 languages and made available to all other euLINK users. There are already several hundred templates in the database, and several more are always waiting in the queue for moderation.

A description of template creation with examples is included in a separate manual. There are also already some of the first template creators who are publishing their templates in our database. We would like to recognise their help by rewarding the creators of the most popular templates with gifts in the form of Smart Home devices produced by us and our Partners (primarily by FIBARO).

There are also other tools available on the market that enable communication with HVAC/PV/EV equipment via MODBUS and other protocols. However, while the price of such a tool is usually comparable to euLINK, it is limited to one HVAC unit or one family of units from one manufacturer (e.g. only air conditioners or only heat pumps). Thus, if there are several different HVAC/PV/EV devices in a regular house, several such tools would need to be purchased and programmed for integration with the FIBARO system. The euLINK gateway is the first such product that can communicate with different HVAC/PV/EV devices from different manufacturers, and you only need to buy it **once** and use several different templates from our database. If the template you need is not yet in the euCLOUD database and you are not yet experienced in creating your own templates, please leave us a message on the "wish list" or send us an email at support@eutonomy.com and we will try to develop such a template quickly.

5. Preparations for installation

Basic preparations are not worth underestimating, as neglecting them can result in an awkwardly large waste of time in the later stages of installation.

a) Gathering the necessary documentation for the equipment to be integrated

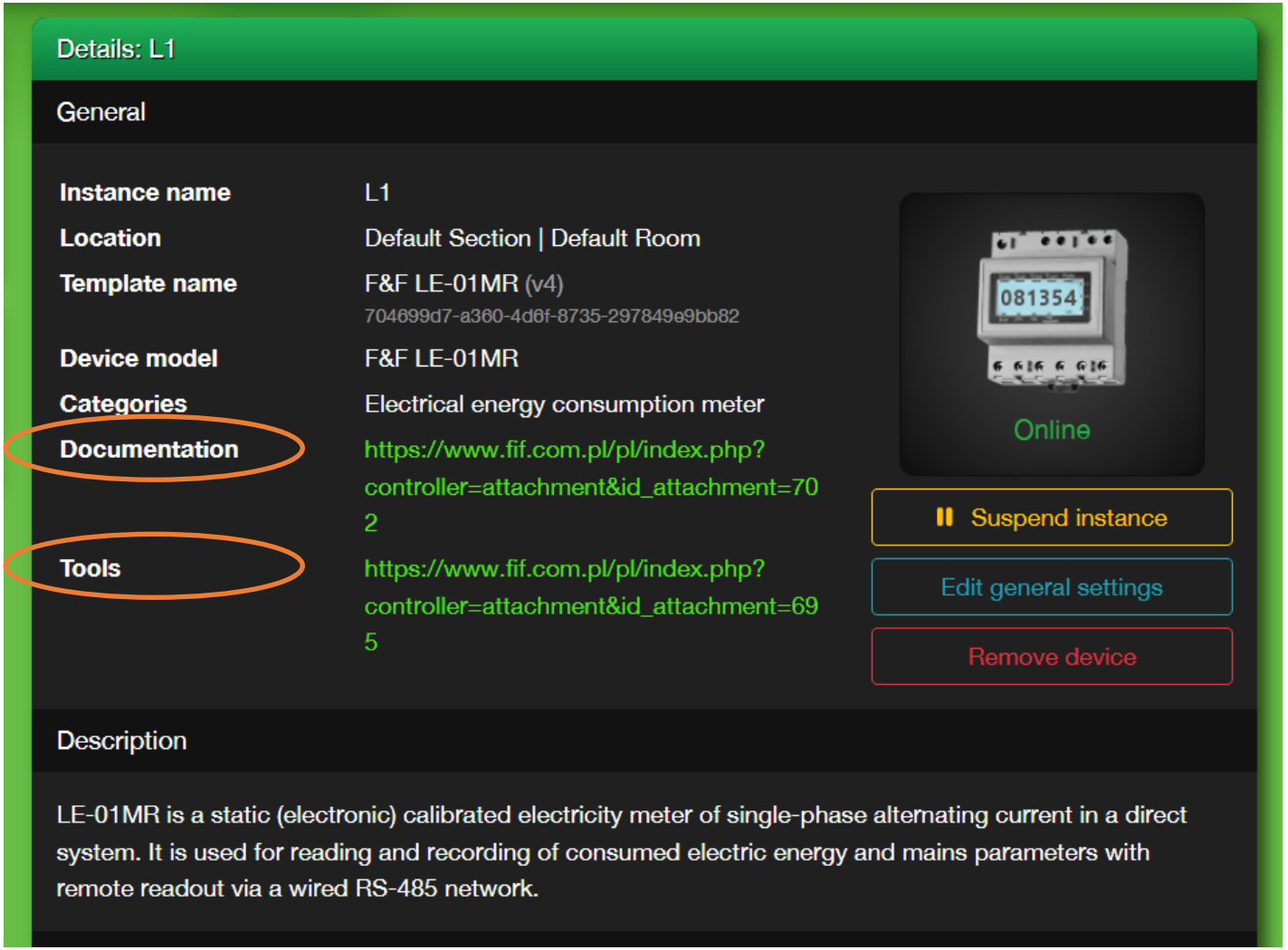
Successful integration of an HVAC/PV/EV unit may require small, mundane changes to its configuration. You may need to set its Slave_ID address (for MODBUS RTU) or TCP/IP address and TCP port (for MODBUS TCP). Sometimes, you may even need to indicate permission to communicate with an external BMS in the MODBUS protocol, as this is disabled by default - presumably for security reasons. These steps are almost always left to the team installing the HVAC/PV/EV unit in question, but there are exceptions. Sometimes a long time has passed since the unit was installed and the need for integration was not yet considered at the design stage. Occasionally, even a certified HVAC/PV/EV equipment installer will feel insecure about an unfamiliar SmartHome and ask to agree on a list of configuration changes, or even to assist in making those changes. Therefore, the most up-to-date documentation of the HVAC/PV/EV equipment from its manufacturer will be useful for many reasons.

To make it easier to gather documentation, we include a link to the published material in our templates for each MODBUS device. After all, the creator of the template for the euLINK gateway has to use this documentation when building the template and device emulator, so he must have found this documentation beforehand. It makes no sense for the integrator to waste his valuable time finding the same material. If there are additional tools (e.g. configuration programs or schematics), the template header for the euLINK gateway will have a link to the mentioned tools in addition to the documentation link. Information about the template and its documentation can be found by navigating to *Menu => Devices => All templates => Find template =>*  *Show details*:


F&F LE-01MR
Template details
Template name
F&F LE-01MR (v0)
969fb7e1-f9ad-4e85-abfb-8018d37f35b8
29.03.2024 17:59
+ Create instance
Clone this template
Manufacturer
F&F
Models
LE-01MR
Categories
Electrical energy consumption meter
Hardware requirements

Description
LE-01MR is a static (electronic) calibrated electricity meter of single-phase alternating current in a direct system. It is used for reading and recording of consumed electric energy and mains parameters with remote readout via a wired RS-485 network.
Comment
Connectors
Modbus RTU
Documentation
https://www.fif.com.pl/pl/index.php?controller=attachment&id_attachment=702
Tools
https://www.fif.com.pl/pl/index.php?controller=attachment&id_attachment=695
Manual in PDF format
Application for device configuration

The same links to the documentation can also be read in the configuration of a device whose instance has already been created based on the template and is operational. To do this, navigate to *Menu => Home =>* (Find Device on the list) *=>*  *=> General:*



b) Installation plan

Reading the documentation of the devices to be integrated enables the designer to become familiar enough with their specifics to be able to draw up a wiring diagram and a list of the devices to be integrated with the MODBUS RTU protocol, for example in the form of the following table:

Room		Name	Slave_ID	Terminator	Device			
Section	Room				Manufacturer	Category	Type	Interface
Ground floor	Boiler room	euLINK	-	120 Ω	Eutonomy	Gateway	euLINK Lite	Built-in
Ground floor	Salon	AC Salon	2	-	Daikin	Air conditioner	FXAQ-A	RTD-RA
Floor 1	Bedroom	AC Bedroom	3	-	Daikin	Air conditioner	FXDQ-M	RTD-RA
Floor 1	Roof	Weather	4	120 Ω	Elsner Elektr.	Weather station	P03/3-Modbus	Built-in

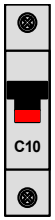
Of course, in the table above, the most important are the rooms, the devices and the Slave_ID addresses assigned to them.

An analogous table for devices with MODBUS TCP protocol could look as follows:

Room		Name	Unit_ID	TCP/IP parameters		Device		
Section	Room			IP address	Port	Manufacturer	Category	Type
Ground floor	Boiler room	euLINK	-	192.168.1.26	-	Eutonomy	Gateway	euLINK Lite
Ground floor	Boiler room	Heat pump	1	192.168.1.28	502	NIBE	Heat pump	Fighter 1120 + RCU-10
Ground floor	Garage	Charger	1	192.168.1.33	5502	Go-e	EV charger	Gemini
Floor 1	Roof	PV	1	192.168.1.41	502	RCT Power	Photovoltaics	Power Storage DC 6.0

The tables appear to be very simple, but each heading is important - especially when looking for a possible anomaly. Without such a table, subsequent configuration, commissioning, diagnostics and servicing will be virtually impossible, so it is best to create such a table at the outset and keep it up to date.

6. Installation and commissioning of the euLINK gateway



a) Connection of power supply and peripherals

The euLINK gateway has a built-in mains power supply, suitable for AC voltage from 100V to 240V AC. The power consumption of the gateway does not exceed 14W, and the gateway power supply circuit should be protected with a 10A overcurrent circuit breaker with C characteristic.

The euLINK gateway is designed for mounting on a DIN TH35 rail in the electrical switchgear[2], as this is where power supply and control cables from the building's infrastructure devices generally converge. Communication of the euLINK gateway with the FIBARO Home Center takes place via the local area network (LAN), so at least 1 LAN cable should be led to the electrical switchboard. Although the euLINK gateway can also use WiFi, it is a "last resort" in older installations where no LAN cable has been run to the distribution board.

Before mounting the euLINK gateway on the DIN rail, it is worth examining all the gateway ports carefully, as some of them are somewhat unusual. The power supply, LAN or USB socket is not questionable, but the other sockets require some attention:

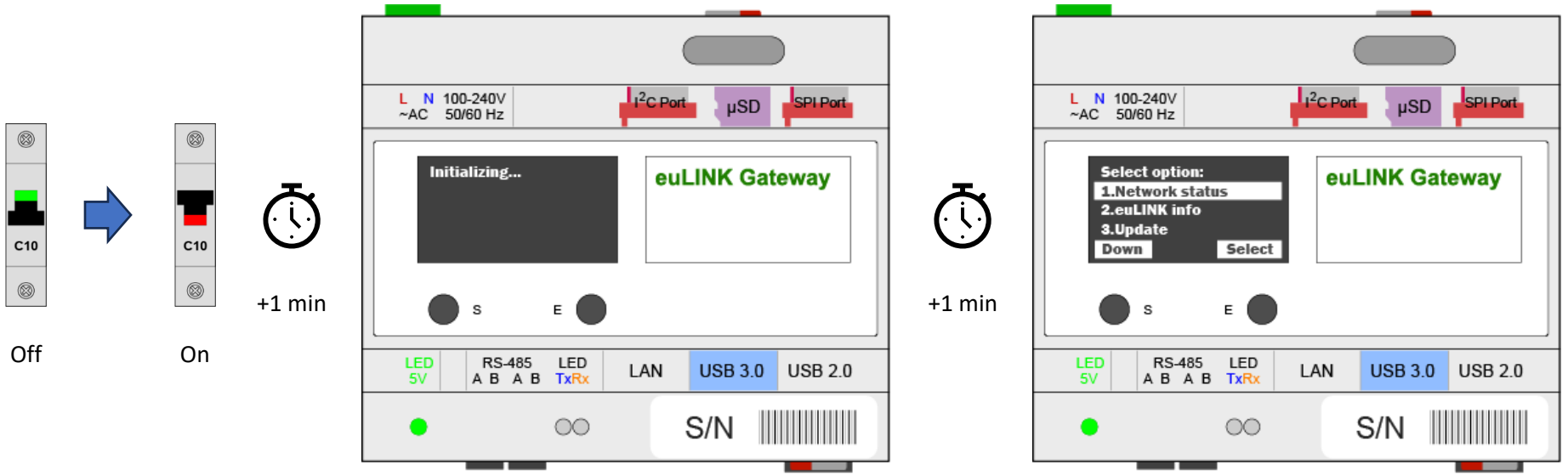
- Micromatch sockets with the description **I²C Port** and **SPI Port** are used to connect peripheral devices, e.g. the *euLINK DALI port*. Please pay close attention to the stickers on the euLINK gateway housing indicating the correct direction to connect the plugs to these two ports to avoid damage to the gateway and/or peripheral devices.
- RS-485 serial port sockets - although there are two sockets, there is only one built-in RS-485 port. Its sockets are duplicated because this makes it easier to connect the euLINK gateway to two separate segments belonging to one and the same RS-485 bus. This also facilitates independent diagnostics of both segments. If the RS-485 bus consists of only one segment, the second slot will remain unused. Under no circumstances should a 120Ω resistor be connected to either of these sockets - see section 8.c) on page 28. If more than 1 RS-485 serial port is required, use the USB⇔RS-485 converter, described in detail further on page 41.

b) Launch of the euLINK gateway

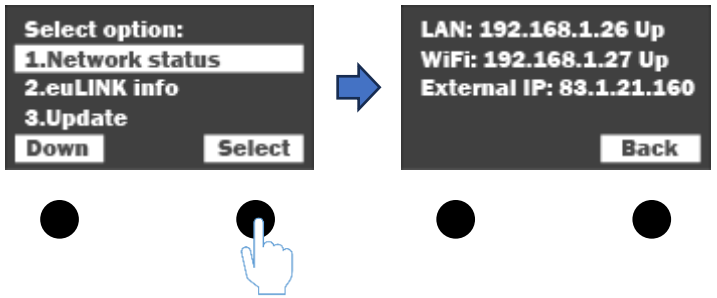
When power is applied, the green **5V** LED should light up immediately and the **Tx** and **Rx** LEDs may light up briefly. However, after 10 seconds at the most, both **Tx** and **Rx LEDs** should go out, and if this has not happened, it could indicate either a startup problem with the euLINK gateway operating system or a fault with the main (top) microSD card.

[2] The euLINK gate can also be mounted directly on a wall or screwed to another surface using the 3 rear latches, which have holes for screws with a maximum diameter of 4mm. These latches should be extended slightly to protrude beyond the housing.

Approximately one minute after power is applied, the OLED display of the euLINK gateway comes to life and displays the message '*Initialising*', and after another minute the normal menu should be displayed:



The IP address of the euLINK gateway can be read from the OLED display, which is particularly useful if the LAN address has been assigned dynamically by the DHCP service. To navigate the OLED menu, the two buttons below the display are sufficient, whose functions are admittedly dynamically variable, but their current role is always displayed in the lowest row of the OLED, just above the buttons. Both the main OLED menu and each submenu counts only a few items, so you only need to press the *Down* button a few times to traverse the entire menu and return to the beginning.



c) Logging in to the euLINK gateway for the first time and using the configuration wizard

Any web browser can be used to configure, test and diagnose the euLINK gateway by pointing to the read IP address of the euLINK gateway.

When logging in for the first time, use the access data below:

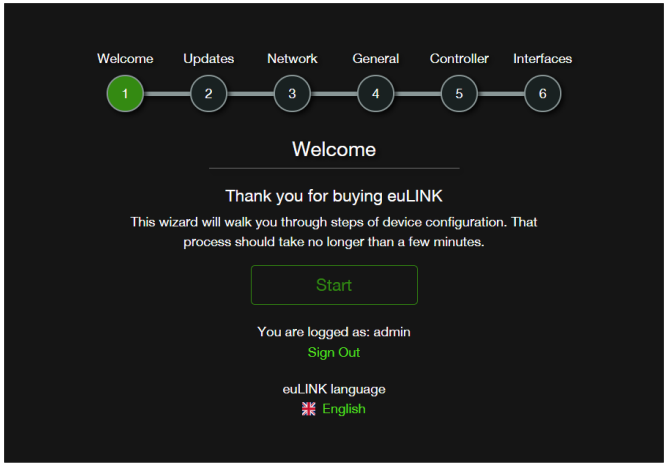
- Username: ***admin***
- Password ***admin***

However, you must remember to set a password that is difficult to break before commissioning the system. The user name, password and e-mail address can be changed by navigating to *Menu => Settings => Users =>* *=> Edit User*.

The euLINK gateway should recognise the language to which the installer's browser is configured and, if this is one of the languages known to the gateway, euLINK will attempt to conduct all further communication with the installer in this language. If the language is not recognised, the euLINK gateway will adopt English by default. The list of built-in languages includes:

English	French	Estonian	Dutch	Portuguese
Polish	Spanish	Russian	Norwegian	Czech
German	Italian	Ukrainian	Swedish	Slovak

The language can also be changed manually - either before logging in (in the login dialog) or after logging in on the main euLINK page in the top right corner by clicking on the flag icon.



When logging into the euLINK gateway for the first time, a wizard will be launched to prepare the basic configuration of the gateway, including:

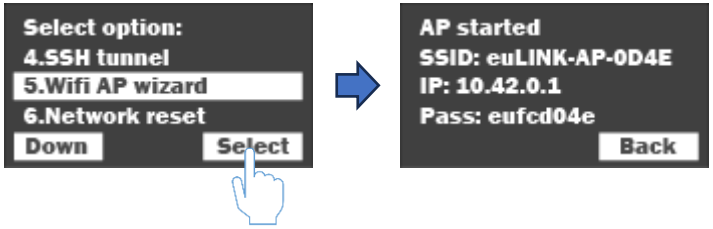
1. Welcome
2. Checking the availability of a newer software version
3. LAN and WiFi network configuration
4. General parameters (gateway name, name and e-mail of the installer)
5. IP address, login and password for the FIBARO Home Center controller
6. Configuration of built-in RS-485 port, protocol selection

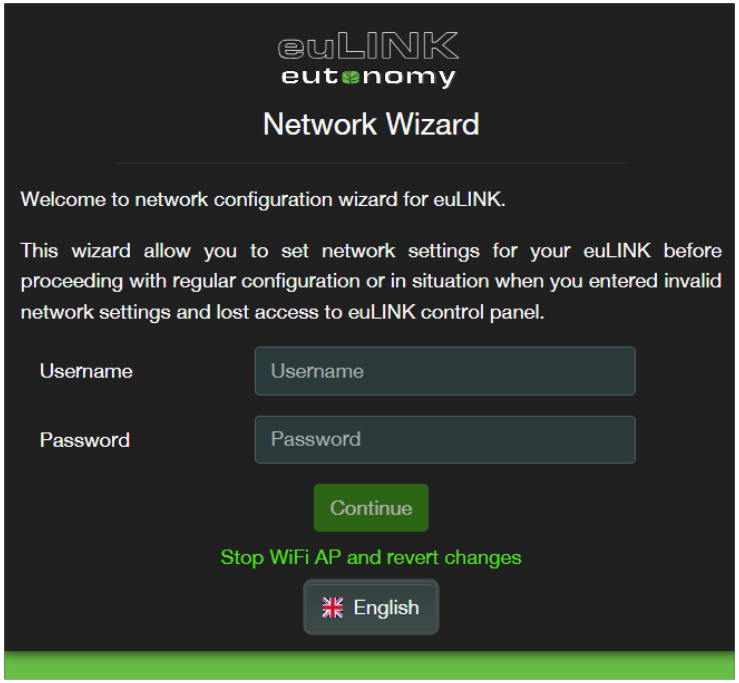
The work of the wizard can always be interrupted, because after restarting the wizard will pick up from where it was interrupted. You can skip some steps of the wizard and complete the configuration at a later date.

d) WiFi configuration in the absence of a wired LAN connection

If there is no wired LAN available at the euLINK gateway site, the first configuration may be somewhat difficult. The euLINK can then run its own wireless network, allowing the installer to establish a connection and perform the network configuration. This method can be used even if the wired LAN is available but dynamic address assignment (DHCP) is not in force and euLINK does not yet have a valid IP address stored.

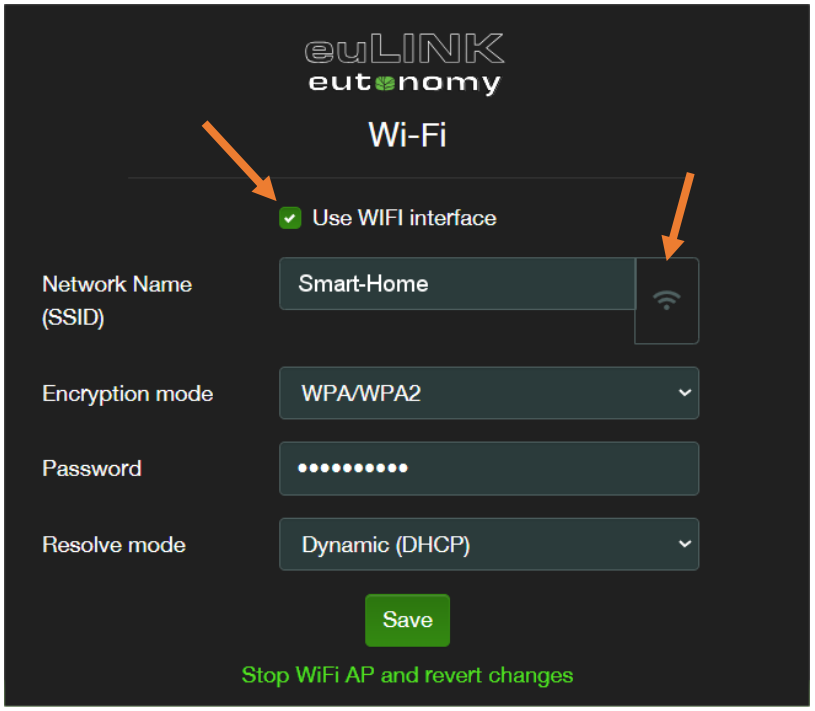
The procedure consists of selecting the item "*5.Wifi AP wizard*" in the OLED display menu and confirming this command. As a result, the OLED display should show a confirmation of the start of the AP (*Access Point WiFi*) and all the information necessary to log into this mini-network (SSID name, IP address and password). The password is always composed of the letters *eu* and the serial number of the euLINK gateway being installed (in small letters). The installer can then use the web browser on his/her computer, tablet or smartphone and point to its IP address, as read from the OLED display: http://10.42.0.1/.



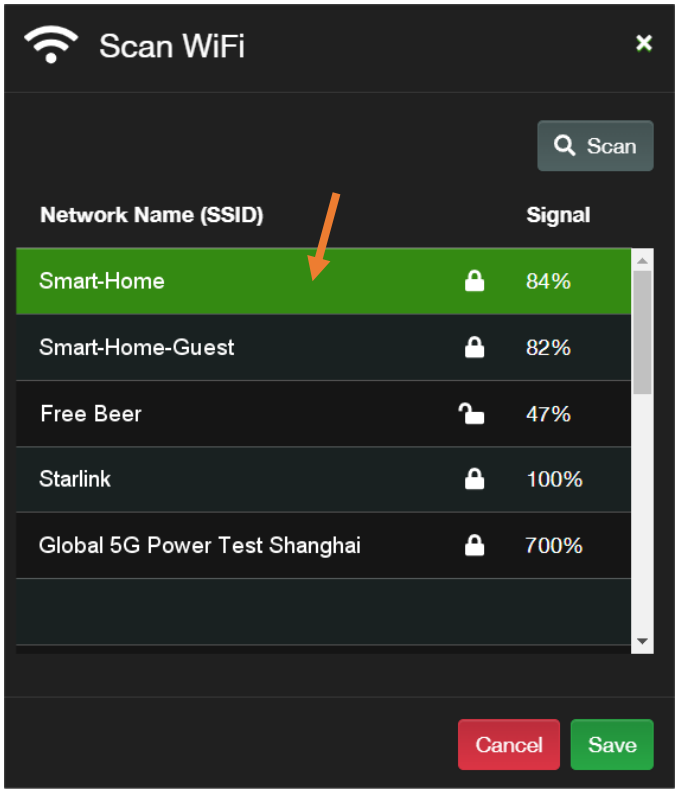


As with the regular wizard, you need to enter your current euLINK gateway account and password (**admin/admin** at the factory) to log in and configure the target network. You will first be asked for the IP address (static or taken from DHCP) for the wired interface of the euLINK gateway. If the aim is not to correct a static address, the DHCP setting can be left in place.

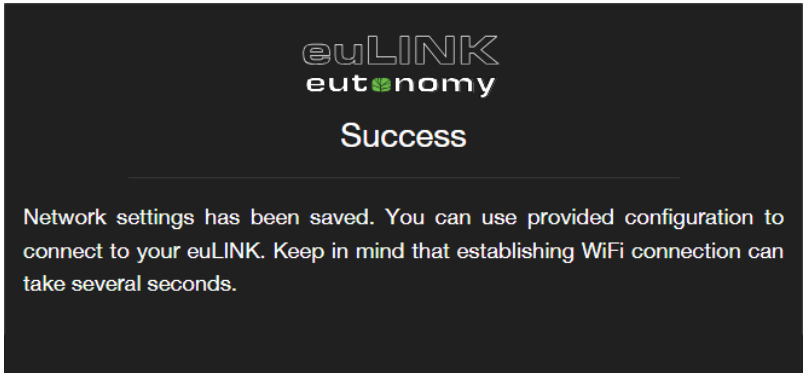
You will then be prompted to use and configure your wireless network. After pressing  (to the right of the SSID network name field), the euLINK gateway will scan the environment and display a list of found wireless networks. Indicating the target network and saving this decision will close the window with the list of networks and return to the configuration. Enter the password for the selected WiFi network and decide on the IP addressing (static or DHCP).



After saving the wireless network settings, the euLINK gateway's internal WiFi access point should be stopped from the OLED display menu using the *STOP* button. You can then log the installer's computer into the target WiFi network, read the euLINK gateway's WiFi network address from the OLED menu and connect to this address using a browser.



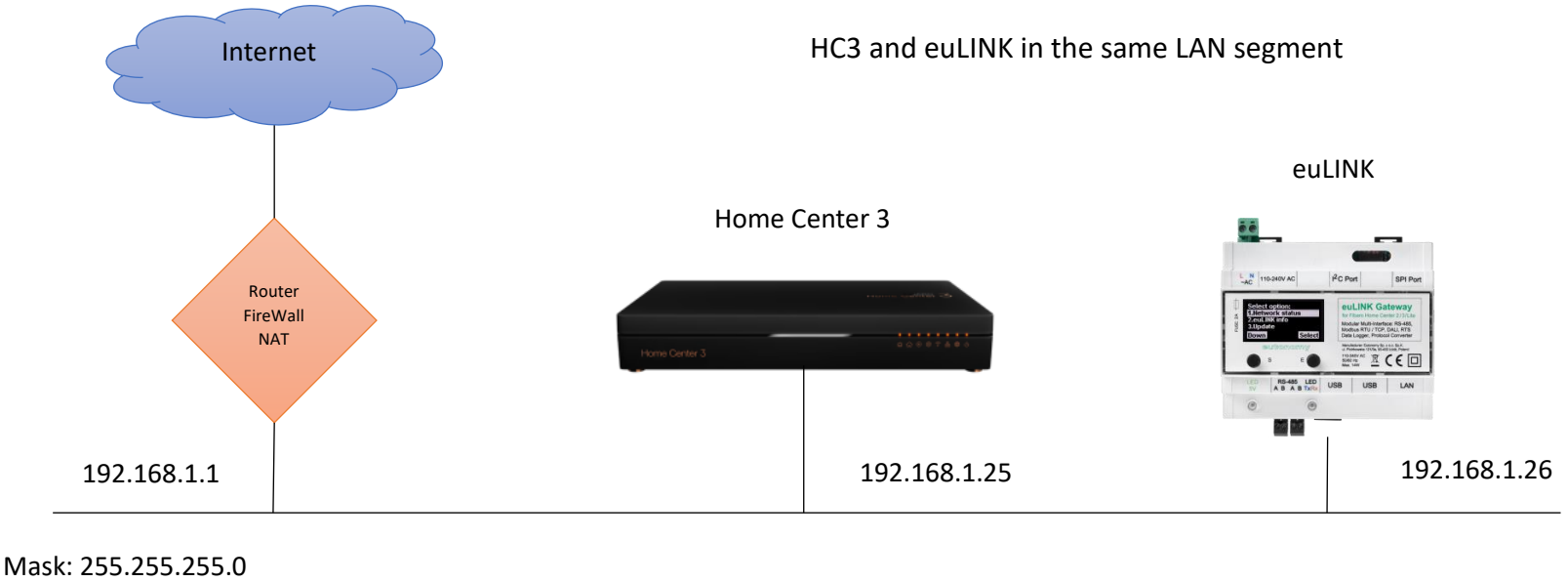
The subsequent procedure is no different to using a wired LAN.



7. Physical connection of the euLINK gateway to the FIBARO Home Center controller

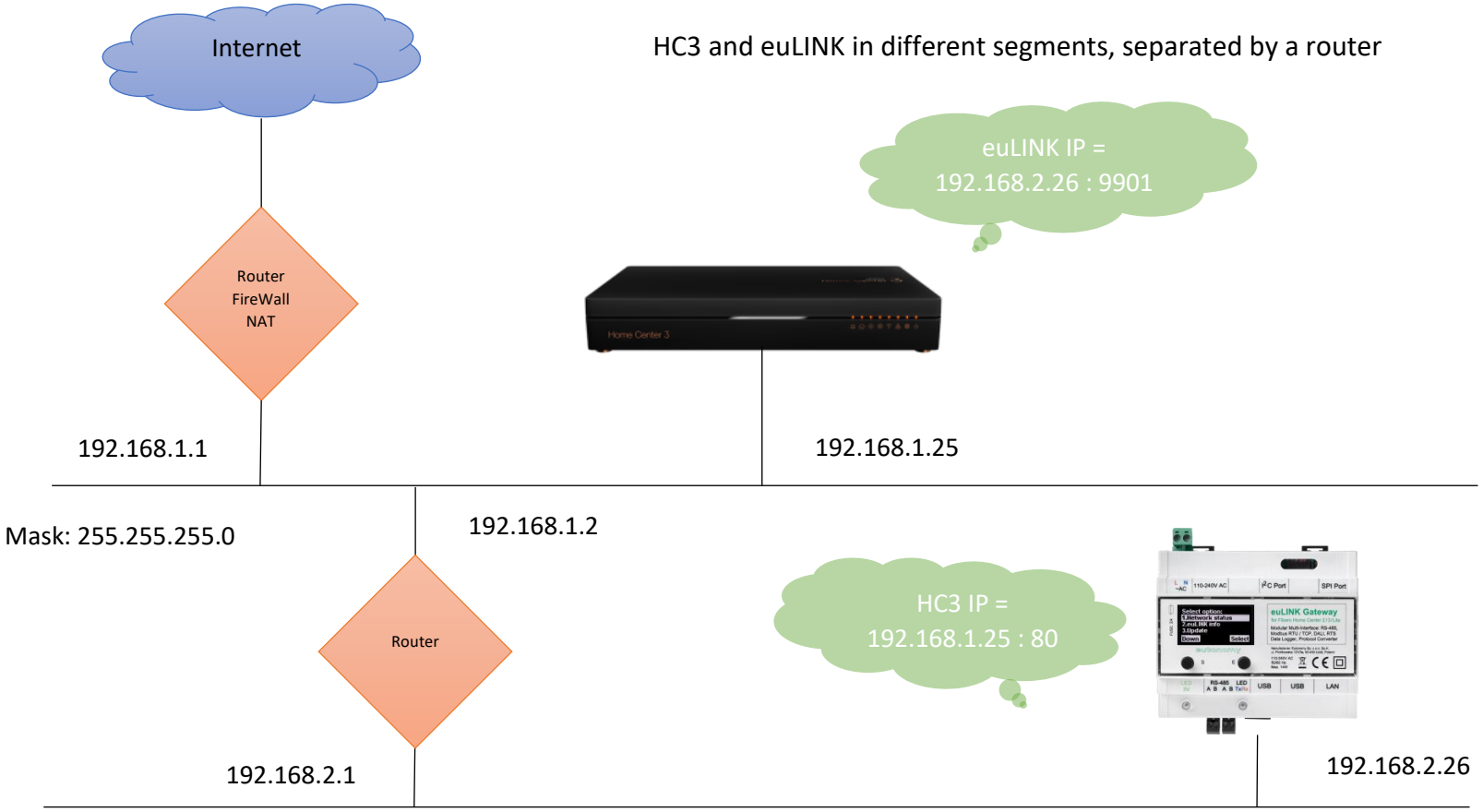
To enter the parameters of the FIBARO Home Center controller into the euLINK configuration, navigate to: *Menu => Settings => Controllers*.

It is not required that the two devices are connected to the same local LAN segment for correct operation. But, of course, this is the recommended option, as this facilitates a number of activities. For example, when entering the access data for the Home Center controller into the euLINK gateway configuration, you can enter the IP address manually, but if both devices are in the same segment, you can have the euLINK gateway search the network environment and euLINK will display any HC2, HC Lite, HC3, HC3 Lite or Yubii controller found in the segment.

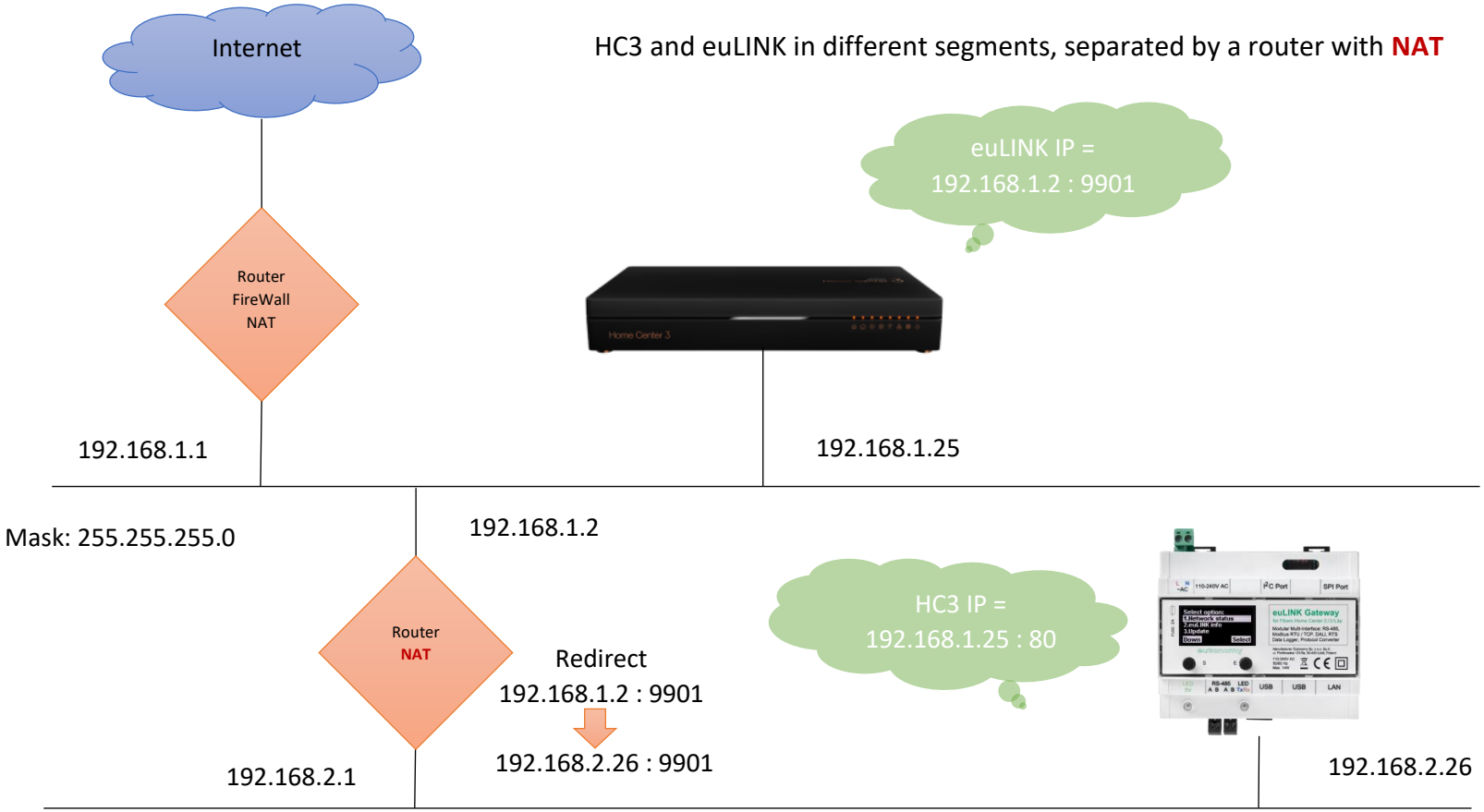


Furthermore, if at some point the euLINK gateway finds that the Home Center controller is not responding at the saved IP address, it will attempt to find it and, if it detects a Home Center controller operating at a different IP address with the same serial number as the saved one, it will attempt to log in using the saved username and password. If this attempt is successful, the euLINK gateway recognises that the IP address of the Home Centre controller has changed and will henceforth use this newly found address.

It is also possible to work in a multi-segment network, connected by one or more routers. It is important that both IP addresses are reachable for each other and that the TCP ports needed for communication are not blocked, i.e. port 80 in the direction euLINK=>HC3 and port 9901 in the direction HC3=>euLINK. For the older controller family (HC2, HC Lite), port 9900 is used instead of port 9901.

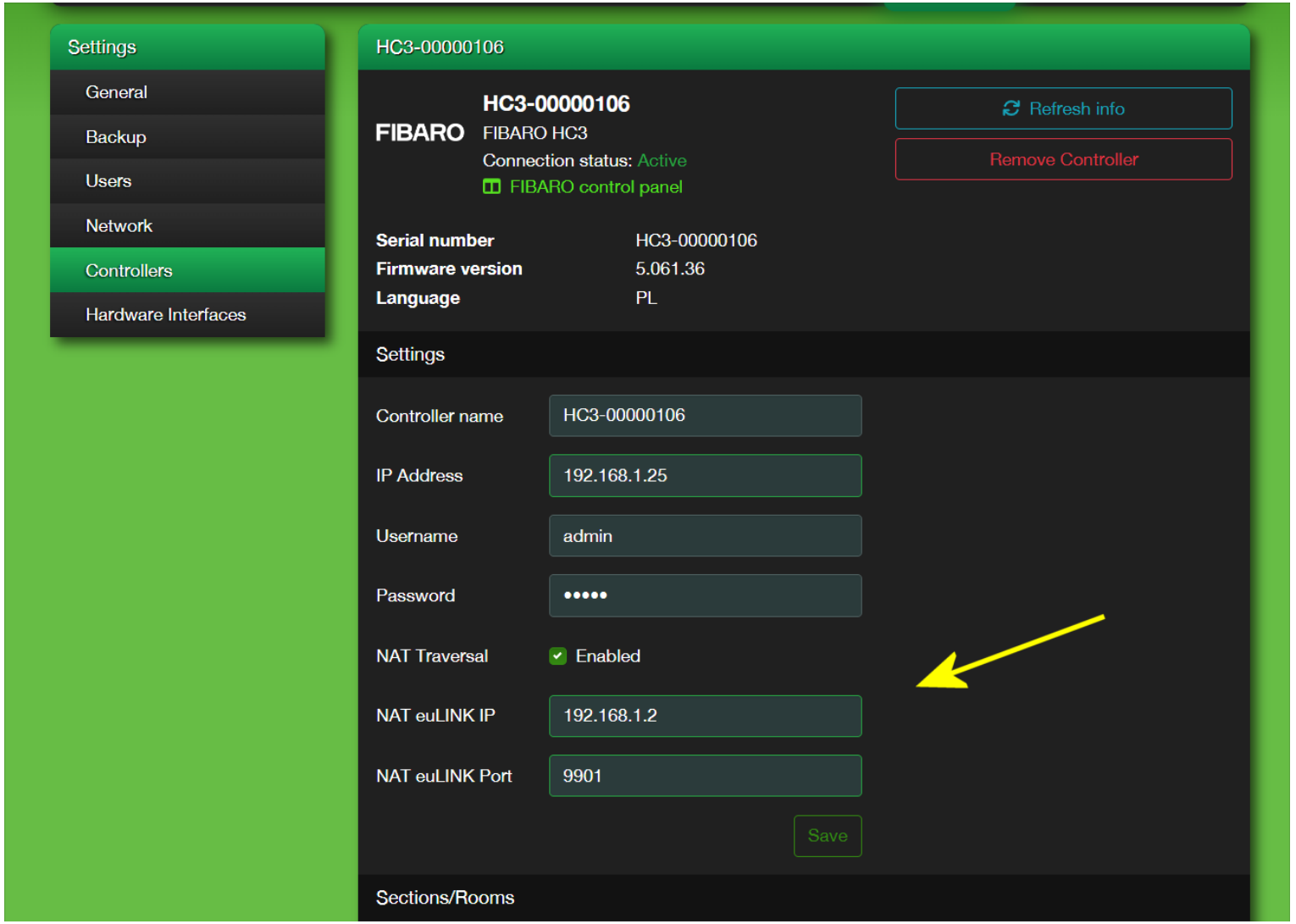


It is even possible to work in a network whose segments are separated by a NAT-translated router. The addressing should then look as follows:



In the settings of the euLINK gateway in the controller section, the option "NAT Traversal" must then be selected and the IP address of the NAT-translated router, as seen from the Home Centre controller side, must be indicated. The euLINK gateway configured in this way will import any QuickApp objects with the NAT router address, instead of the euLINK gateway address. Of course, on the NAT router, you will then need to configure the redirection of traffic entering

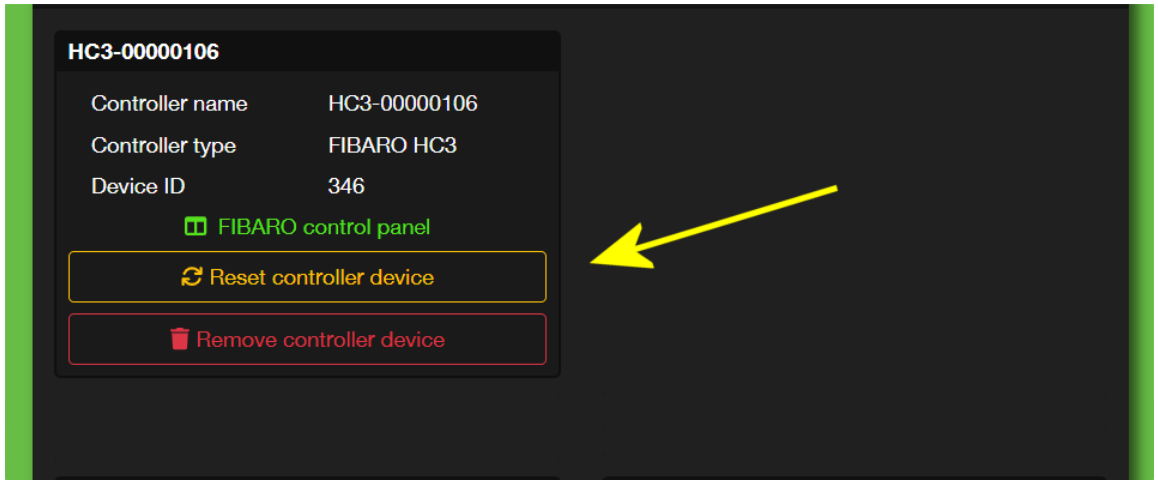
TCP port 9901 to the euLINK gateway address and the same TCP port 9901. The controller settings page on the euLINK gateway for the above example looks as follows:



a) Fixed IP addressing of the euLINK gateway and the FIBARO Home Center controller

Both devices - the FIBARO Home Center controller and the euLINK gateway - need to know each other's IP addresses, so these addresses must remain constant throughout their lifetime. Traffic between the devices is kept to a minimum, so they only transmit information (euLINK to HC3) and commands (HC3 to euLINK) to each other when they really need to. And for this they need to know each other's IP addresses. If one or both of these devices have addresses assigned dynamically by the DHCP service, then even with a minor network reconfiguration these addresses can change and communication in one or both directions becomes impossible. This can manifest itself, for example, in such a way that information about the integrated devices will be sent via the euLINK gateway to HC3, but no longer commands from HC3 will reach the euLINK gateway and further to the integrated devices. This almost certainly indicates that the IP address of the euLINK gateway has recently changed and that the HC3 controller has not been informed of this. It is therefore necessary to give both devices - HC3 and euLINK gateway - fixed IP addresses and try not to change them afterwards. This can be achieved by entering fixed IP addresses in the configuration of both devices, but this will make any future LAN reconfiguration more difficult. It is therefore much better to indicate in the configuration of the DHCP address allocation service that the IP addresses assigned to the euLINK gateway and HC3 are to be permanently assigned to their unchangeable MAC addresses.

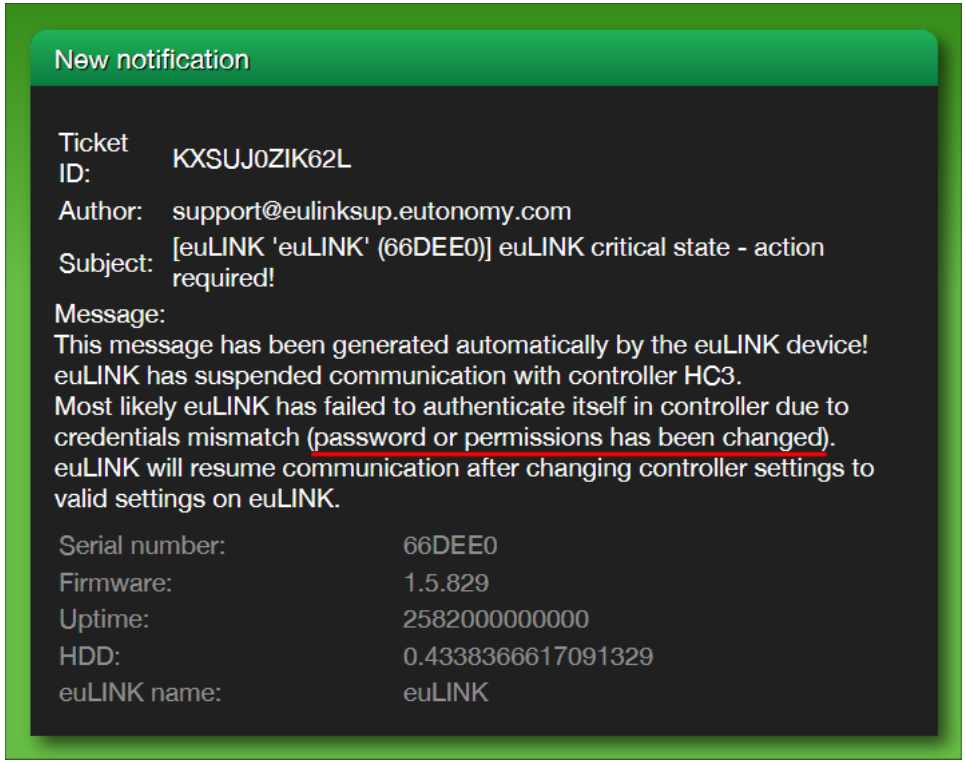
It is a good idea to plan the IP address of both devices in advance and not to change it afterwards without a serious reason, as this can have time-consuming consequences. This is because every QuickApp object imported by the euLINK gateway has the gateway's IP address stored, and if this is later changed, you will need to remember to update it for each QuickApp object. However, should the need arise, the fastest way to make corrections is from the device configuration screen on the euLINK gateway. This is done using the 'Reset controller device' function, which sets all the parameters of the imported QuickApp object to the current values, including saving the current IP address of the euLINK gateway.



The euLINK gateway has some defence mechanisms built in, so that if an HC3 controller stops responding to it at a known address, euLINK will search the local network and if it finds a controller with the same serial number at a different IP address, it recognises this new address as valid. And if the euLINK gateway detects that its own IP address has changed, it will try to update all entries in the QuickApp objects it has imported in HC3. But this will always be a non-standard action that is better avoided.

b) Login of the euLINK gateway to the FIBARO Home Center controller

In order to automatically import QuickApp objects, global variables, icons and for subsequent status updates of integrated devices, the euLINK gateway must be able to log in to the Home Center controller with local administrator rights. Usually, the username and password for logging into Home Center are entered at the beginning of the installation. If the administrator name or password needs to be changed later, remember to update this name and/or password in the euLINK gateway settings as well. Otherwise, the euLINK gateway will lose the ability to update the status readings of the integrated devices. To make matters worse, after detecting several unsuccessful attempts to log in from the same IP address, the Home Center controller will consider this as an intrusion attempt and will block any possibility of logging in from this suspicious IP address for the next 30 minutes. To avoid this, the euLINK gateway stops further attempts to log in to Home Center after 2 unsuccessful attempts to avoid being blacklisted by the IP address. Instead, it displays a message about problems with logging in and sends an email to the administrator with similar content to that shown on the right:



You should then enter a new name or password as soon as possible and save the change. This should restore normal communication with Home Center. It is worth checking that the login error message has disappeared and that the device readings are updated in Home Center. If not, please wait 30 minutes or restart the Home Center controller.

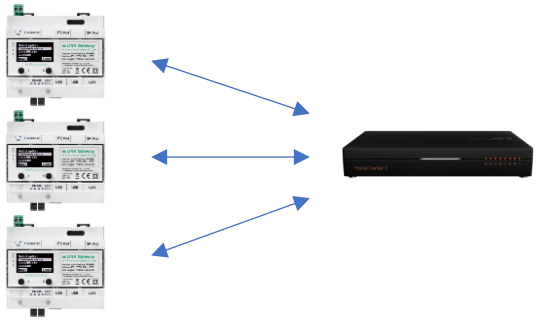
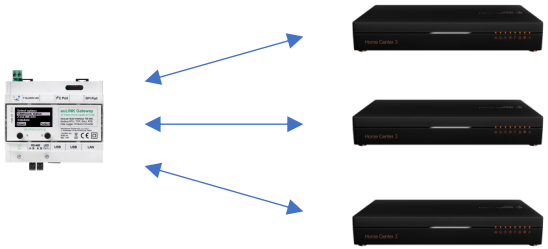
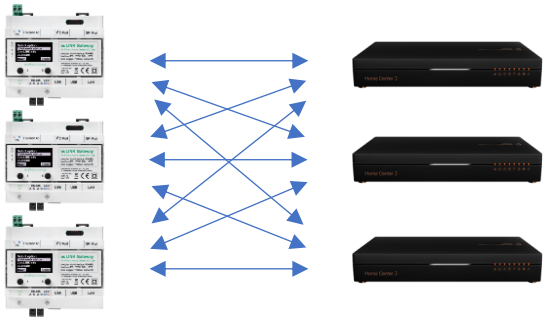
c) Preparation of rooms on the FIBARO Home Center side and downloading them to the euLINK gateway

Once the euLINK gateway is properly connected to the Home Center, download the list of rooms defined in the Home Center configuration by navigating to *Menu => Settings => Controllers =>*  *=> Sections/rooms => Automatic synchronisation*. The list of rooms will be used to assign the integrated MODBUS devices to the corresponding locations. Sometimes it is useful to add a separate room (e.g. "Boiler Room" or "Technical Room") on the Home Centre side in advance to which specialised HVAC/PV/EV equipment can be assigned.

d) Connection of more than one euLINK gateway to the Home Centre and vice versa (N:1, 1:N, M:N)

Designers of complex systems often ask whether it is possible to connect more than one euLINK gateway to a Home Center controller - and if so, what are the quantity restrictions? The answer is simple: the euLINK gateway architecture imposes no restrictions, so euLINK gateways and Home Center controllers can be combined in all possible constellations. Of course, in practice, there will be limitations at some point, but these will be performance limits caused by the load on the euLINK gateways, Home Center controllers or the LAN connecting them. Connections between several euLINK gateways and one or more Home Center controllers are not unusual, and are in fact frequently encountered.

The following connection variants are possible between euLINK gateways and Home Center controllers (HC2, HCL, HC3, HC3L, Yubii):

N:1	1:N	M:N
		

euLINK gateways work with FIBARO Home Center controllers not only in single-family homes, but also in larger buildings and often even in commercial pavilions, factory halls or warehouses. One record-breaking installation includes 1 HC3 controller, 3 euLINK 2G gateways and 10 euLINK DALI ports, installed in a factory hall, where they control the operation of around **600** DALI luminaires and several MODBUS light intensity sensors. Of course, in practice there is no need to control each luminaire individually, so they are virtually grouped together and the HC3 controller manages a constellation of dozens of DALI luminaire groups. Long-term observation of the resource commitment (CPU and memory load) shows that both the HC3 controller and all 3 euLINK gateways have a fair amount of spare capacity and mainly get "bored". The euLINK gateways, in combination with the FIBARO Home Center controllers, are one of the cheapest tools on the market for controlling medium and large DALI lighting systems.

8. Physical connections for MODBUS RTU

Almost certainly, some sort of cable will need to be run to the euLINK gateway site from the HVAC/PV/EV equipment for the integration to take place. Wireless solutions are generally dedicated to the air-conditioning manufacturer's own applications rather than for integration with third-party products, so a cable is more likely to prove indispensable. But at the building construction stage, the addition of one or more cables generally does not pose a problem.

On the other hand, the answer to the question of what kind of cable it should be and where it should run from depends strictly on the type of devices being integrated and their communication interfaces. This will be most easily explained using the example of air conditioners. By interface, we mean the communication module with which the air conditioner is (or can be) equipped and which can be used for integration. Sometimes such a module is built-in, and sometimes it has to be selected and purchased, because the manufacturer has developed several variants of modules for different communication protocols. It should not be a module for communication in KNX, BACnet or EnOcean protocols, because they have a completely different philosophy. For our purposes, a universal protocol will be useful, preferably MODBUS RTU or MODBUS TCP.

Actually, mainly 2 scenarios are encountered with air conditioners:

- The air-conditioning system is of the **MultiSplit** type and its outdoor unit is (or can be) equipped with one central communication module, which can be used for integration. In this case, one communication module is sufficient to control all indoor units. The necessary cable runs from the euLINK gateway to the outdoor air-conditioning unit (ODU).
- The outdoor unit does not have any communication capabilities, but the indoor units (IDUs) do. Often, the same ports to which the thermostat or wall controller in a room is connected are used for this communication. In this case, a communication module has to be purchased for each indoor IDU and a cable has to be run from the euLINK gateway (i.e. from the switchboard) to the nearest IDU, from it to the next one and so on until the last one. Such communication modules are usually cheaper than a central module, but unfortunately - you have to buy such a module for each IDU, which can be expensive with a larger number of IDUs.

If the air-conditioning manufacturer has developed a suitable communication module, this is often only available from the manufacturer's internal sales network and you will have to wait for it to arrive, sometimes even for several weeks. If this module has to be installed inside the ODU, this should be done by an authorised air-conditioning installer so that the investor does not lose the warranty on the ODU.

a) Addressing of MODBUS RTU devices on the RS-485 serial bus

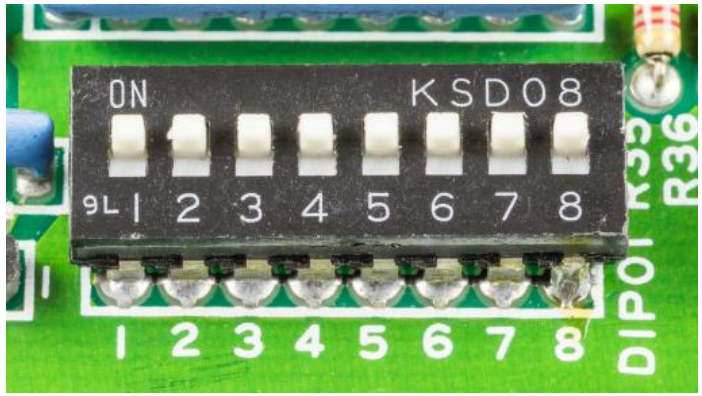
The MODBUS RTU protocol uses a two-wire serial bus compatible with the RS-485 standard for communication.

All communication on the RS-485 bus takes place at the initiative of a single master, the MODBUS master. The euLINK gateway acts in this role. The MODBUS master sends requests and commands to a single slave device using its Slave_ID address, which is unique within the RS-485 bus. Although the MODBUS RTU

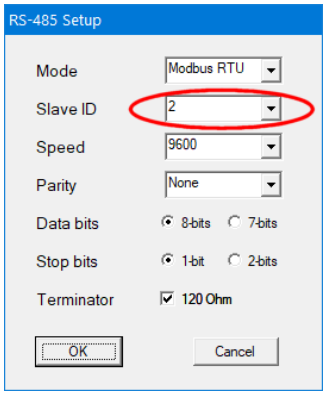
protocol defines the possibility of sending a single query to all slaves (*broadcast* to the Slave_ID=0 address), this approach is not used in home systems, but only in some industrial systems. The called slave is obliged to respond - either in the form of an acknowledgement or to report an error code. Failure to respond within the specified waiting time (generally the *timeout* is around 1s) is also treated as an error by the master. Only after the response has been received or the waiting time has elapsed can the MODBUS master proceed to query the next MODBUS slave.

According to the MODBUS RTU standard, slave addresses can take values in the range from 1 to 247. However, there are devices that do not comply with the standard and their addresses can go as high as 255. The euLINK gateway is able to handle even these non-standard addresses. Some devices have a narrower range of permissible Slave_ID addresses (e.g. from 1 to 15 or up to 63), but this still falls within the MODBUS RTU standard.

It is the responsibility of the installer to give each MODBUS Slave on the RS-485 bus a unique Slave_ID address to the extent, described in the manual by the manufacturer of the slave. The manual also always explains how this address change can be performed. Most often a DIP Switch is used for this, less often it is set in the software configuration of the slave:



Setting the Slave_ID address on the DIP switch

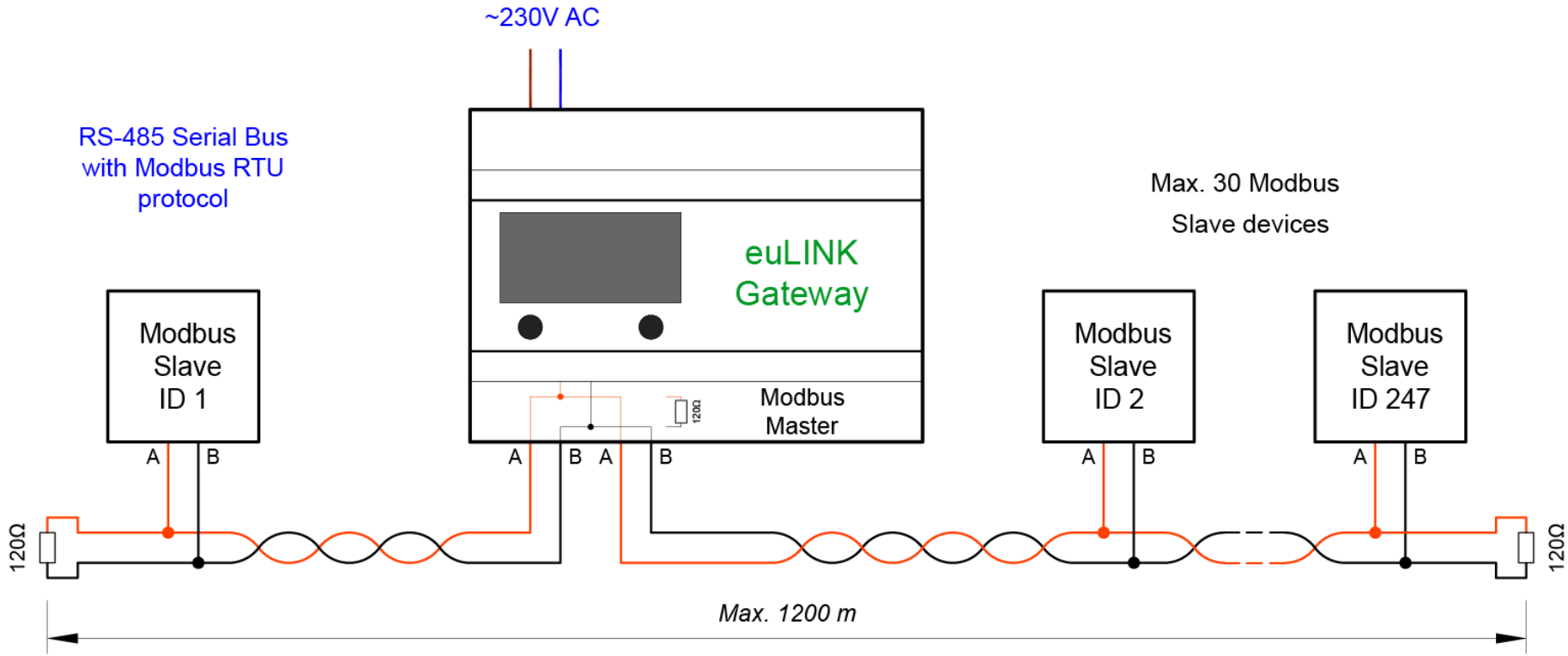


Software address configuration

It is a very costly and time-consuming mistake to connect all devices without changing the factory configuration, as there will almost certainly be a conflict of default Slave_ID addresses and it will be very difficult to identify individual devices - even after adjusting their settings. Most manufacturers default to a factory Slave_ID=1 address, so it is best avoided when planning and configuring devices. It is a very good habit to start the addressing with **Slave_ID=2** and subsequent values, because if a new device is connected in the future, the chance of conflict will be minimised. And because some devices can be changed to a Slave_ID address remotely, it will be possible to switch a new device to a different address, freeing up the controversial Slave_ID=1 for subsequent, future devices.

b) RS-485 bus cabling

The cabling of the RS-485 serial bus is best carried out with a 2-wire "twisted pair" cable with a conductor cross-section of 2x0.4mm^2 (AWG 22) with an impedance of 120Ω, such as the popular Belden 3105A type. Theoretically, the RS-485 bus can be as long as 1,200m, but in practice many device manufacturers limit this length to a few hundred metres - especially for transmission speeds higher than the popular 9600bit/s.

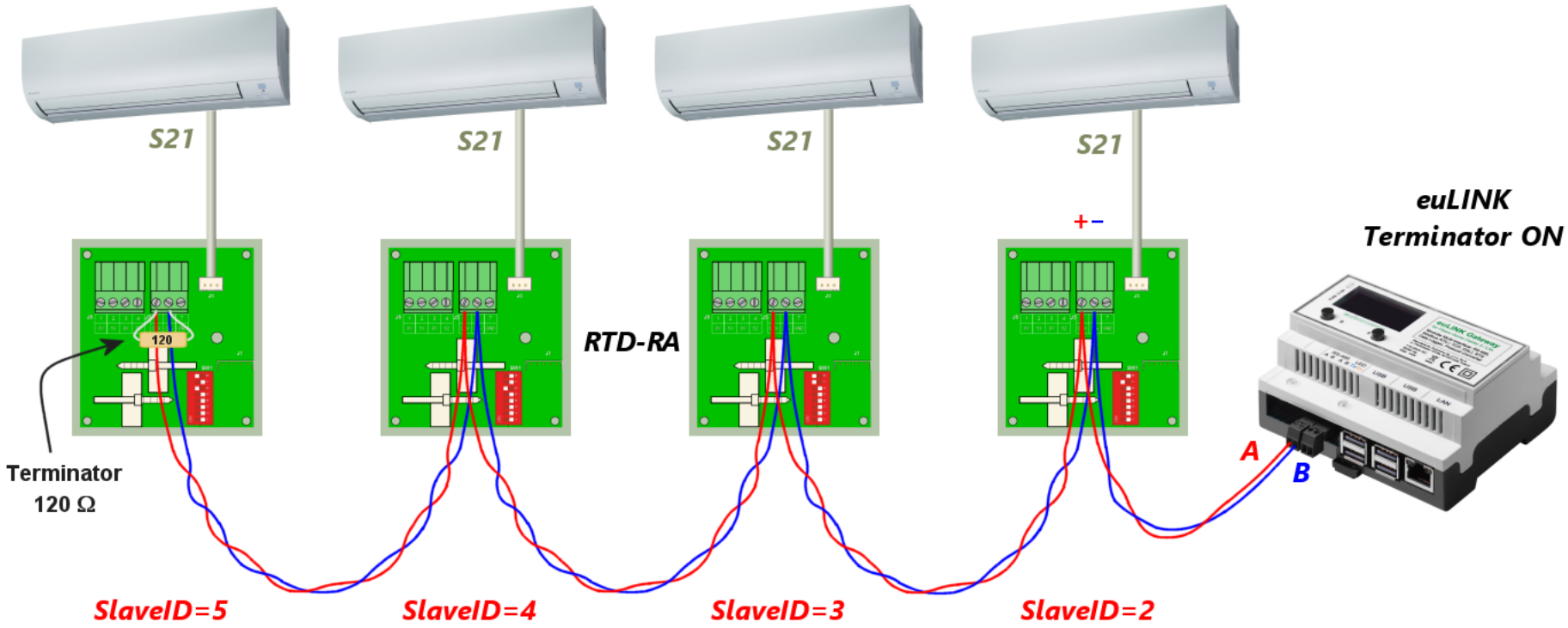


For shorter distances (roughly up to 100m), the use of popular UTP, STP or FTP category 5e or better twisted pair cables gives good results. In environments with high levels of interference (e.g. in industrial settings), it is worth using a shielded twisted pair cable, but then it is important to ensure that the shield is connected to earth at one end only. Some disadvantages of UTP cables are their smaller cross-section (AWG 24) and, in the case of wire , the greater brittleness (breakage) of the conductors, especially in the commonly used screw terminals. This can cause a lack of contact around any type of cable connection (sockets, plugs, terminals and other connectors) and it is from these places that the search for a possible break in the cabling should begin.

One thing that must absolutely be borne in mind is to observe the polarity of the connections, as on the RS-485 serial bus there is a strict assignment of the individual wires to the negative and positive poles of the RS-485 transmitter and receiver. The accepted standards for the designation of these poles will be helpful, i.e. the positive pole is labelled "A" or "+", or alternatively "A+". Correspondingly, the negative pole is labelled "B" or "-", possibly "B-". Therefore, connect the "A" terminal of the RS-485 port on the euLINK gateway to the "A" or "+" terminal of the slave device and the "B" terminal of the euLINK gateway to the corresponding "B" or "-" terminal of the device. Caution should be exercised, however, as there are exceptions to this rule as well. For example, RTD-RA communication interfaces made by *RealTime Control Systems* for Daikin air conditioners have slightly opposite descriptions, i.e. "DB+" and "DA-". In this case, check the manufacturer's instructions carefully and connect terminal "A" of the RS-485 port on the euLINK gateway to terminal "DB+" on the RTD-DA interface, and terminal "B" of the euLINK gateway to terminal i "DA-" of the RTD-RA interface, as shown in the diagram below.

If in doubt, the unique feature of the euLINK gateway, which is the ability to reverse the polarity of the built-in RS-485 port by software, can be used. To do this, navigate to *Menu => Settings => Hardware interfaces => Default RS-485 =>* 🔧 *=> Edit the data bus => Swap AB*. If the connected device is configured correctly (e.g. speed, parity), but you only suspect that it has been reverse-connected, you can swap the polarity of the port on a trial basis and see if this restores communication with the device. However, all other correctly connected devices will then cease to function and, although they will not break down, the correct setting of the euLINK gateway port must be restored and the connection of the suspected device corrected by physically swapping the wires on its terminals.

Please note in the diagram below how the cabling is routed, i.e. from one terminal to the other, without any intermediate branches or "bridges". The RS-485 serial **bus** must have a **bus** topology, i.e. no branches, "star", "tree" or "ring" type connections of any kind. Individual sections must run directly from the terminals of one device to another, and if sections are to connect, then only at the terminals of the device and not somewhere near it.



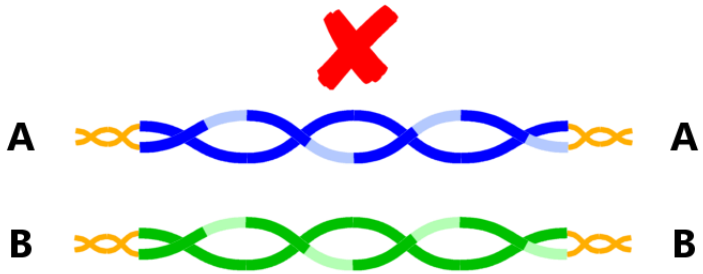
A 120Ω resistor should be connected at the beginning and end of the RS-485 bus as a line terminator, as shown in the diagram above. Since sometimes the way the terminator is connected is questionable, this topic is described in great detail in the following subsection c) on page no. 28.

It is worth ensuring that the pair of transmission cables is twisted together over the longest possible length, as illustrated in the diagram above. Due to the differential method of signal transmission, the use of twisted pair cable means that interference occurring along the cable path cancels each other out on reception. This makes RS-485 transmission very resistant to external interference.

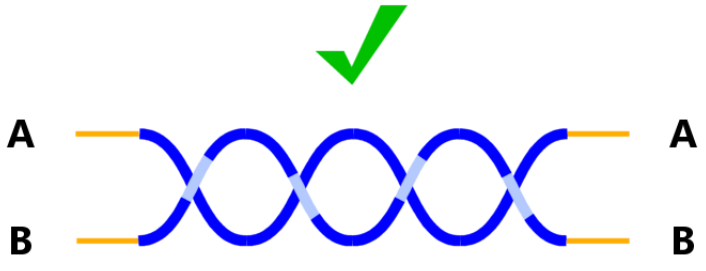
A common mistake made by cabling installation teams is that they want to reduce cable losses by connecting several conductors together if there are unused pairs in the cable. For example, a popular UTP cable has 4 pairs, while only one pair is needed for RS-485 transmission. Installers are therefore keen to combine conductors within pairs, in the hope of reducing the voltage drop. However, this is a serious mistake, as it results in the loss of all the advantages of a twisted

pair cable. Susceptibility to interference will increase greatly, and a negligible reduction in voltage drop will not bring the expected improvement in transmission quality.

The following illustration shows the incorrect and correct connection of a twisted pair cable for RS-485 serial transmission-:



Incorrect use of twisted pairs



Correct use of twisted pairs

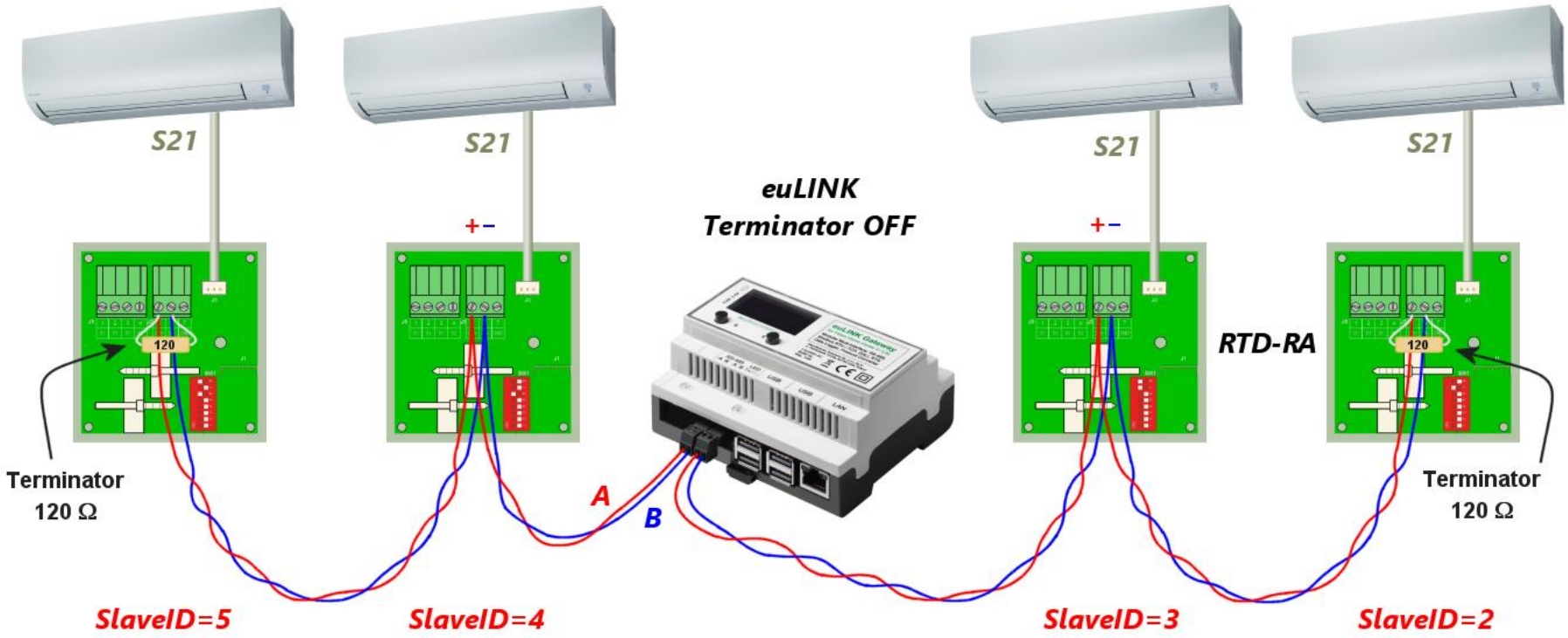
If wiring to the device is not possible, there are methods of wireless connection, described in more detail on page no. 39

c) RS-485 bus terminators

The technique of using terminators is sometimes questionable, and there are even quite a few myths surrounding the subject. For example, some publications state that terminators are not needed at all, which is not true. The purpose of terminators is to match the cable termination to the impedance of the bus, thus avoiding signal reflections and interference, which greatly improves the signal shape and facilitates the work of the RS-485 receiver. The more difficult the transmission conditions, the more helpful is the beneficial effect of terminators. Indeed, if the RS-485 bus is only a few metres long, the signal is strong enough that communication will work even without terminators. But even at several metres or at higher transmission speeds, terminators at the end of the line help a lot. The ideal is 2 terminators - no more and no less, one terminator at each end of the line. A common mistake is to unintentionally connect a third terminator, which will almost certainly prevent correct transmission. Similarly, placing one of the terminators somewhere in the middle of the bus instead of at the end can do more harm than good.

The terminator role is performed by a resistor with a value of 120Ω and a power of 0.5W. The euLINK gateway is supplied with two such resistors. They are fitted by screwing their ends together with the cable wires into the screw terminals of the first and last device on the bus - as shown in the diagram below.

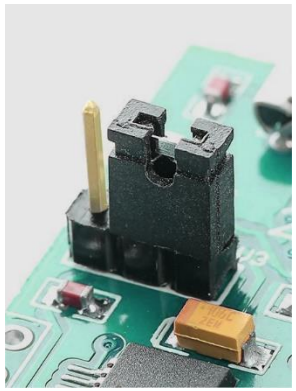
However, these resistors are not always needed, as some devices already have a suitable resistor soldered on the PCB, only the installer needs to decide whether the terminator should be included or not in a particular device. If it is neither the first nor the last device on the bus, then of course the terminator must not be enabled. For example, in the diagram below, the euLINK gateway is in the middle of the bus and not at either end (as often happens in practice) and then the internal 120Ω terminator in the euLINK gateway must not be switched on. Please also note that both bus segments (left and right) are connected to separate sockets of the RS-485 port of the euLINK gateway. This is *de facto* one port, and its two sockets are deliberately duplicated to facilitate a more secure cable connection and to allow independent measurements of each MODBUS segment:



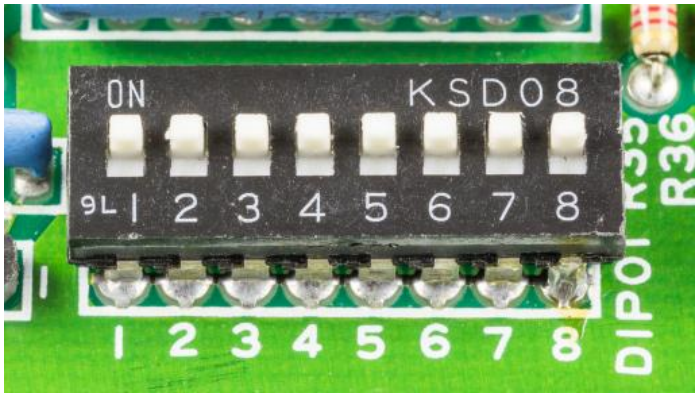
The method of enabling the built-in resistor depends on the design of the device and is always described in the device manual. Enabling the built-in terminator can be done by inserting (or relocating) a jumper, by adjusting a DIP switch or by software in the device configuration. For example, the euLINK gateway has

a software-enabled terminator in the configuration of the built-in RS-485 port-. To do -this, use the browser by navigating to *Menu => Settings => Hardware interfaces => Default RS-485 =>* 🔧 *=> Edit the data bus => Terminator*. This is described in detail in chapter 10 on page no. 40.

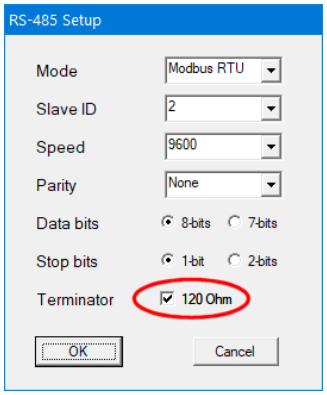
The following illustrations show the most common methods of incorporating the terminator:



Inserting a jumper



DIP switch setting



Software configuration

There are devices on the market which have a 120Ω terminator permanently soldered on, without any possibility of switching it off. If we do not know about it and we connect such a device somewhere in the middle of the bus, it will prevent any transmission, because it will introduce a third terminator on the bus, in addition in the wrong place. One common example of devices with a permanently built-in terminator are various types of USB⇔RS-485 converters, which are described in more detail in chapter 10.c) on page 42.

Therefore, in case of doubt, it is best to confirm this empirically by simply measuring the resistance with a universal meter on a range up to 200Ω or in automatic range selection mode[3] . Of course, the device should be disconnected from the power supply during the measurement.

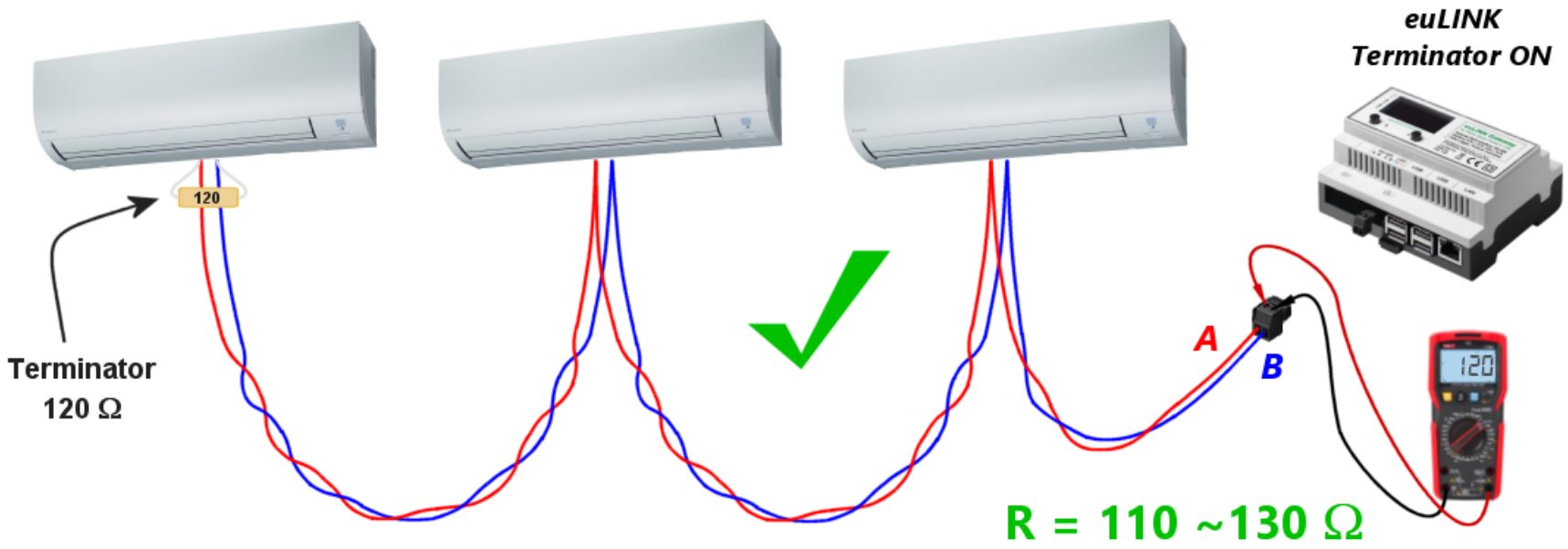
If the result of the measurement is between 100Ω and about 130Ω, the resistor should be considered switched on and look for a way to switch it off in the instructions. However, if the manufacturer has not provided for the possibility of disabling the terminator, then such a device can only be located at the beginning or end of the bus and absolutely no external 120Ω resistor should be attached next to it.

[3] It is worth starting by measuring the attached 120Ω resistor to ascertain how the meter presents the result multiplier (Ω/kΩ/MΩ) in automatic range selection mode.

d) RS-485 bus tests and measurements, fault location

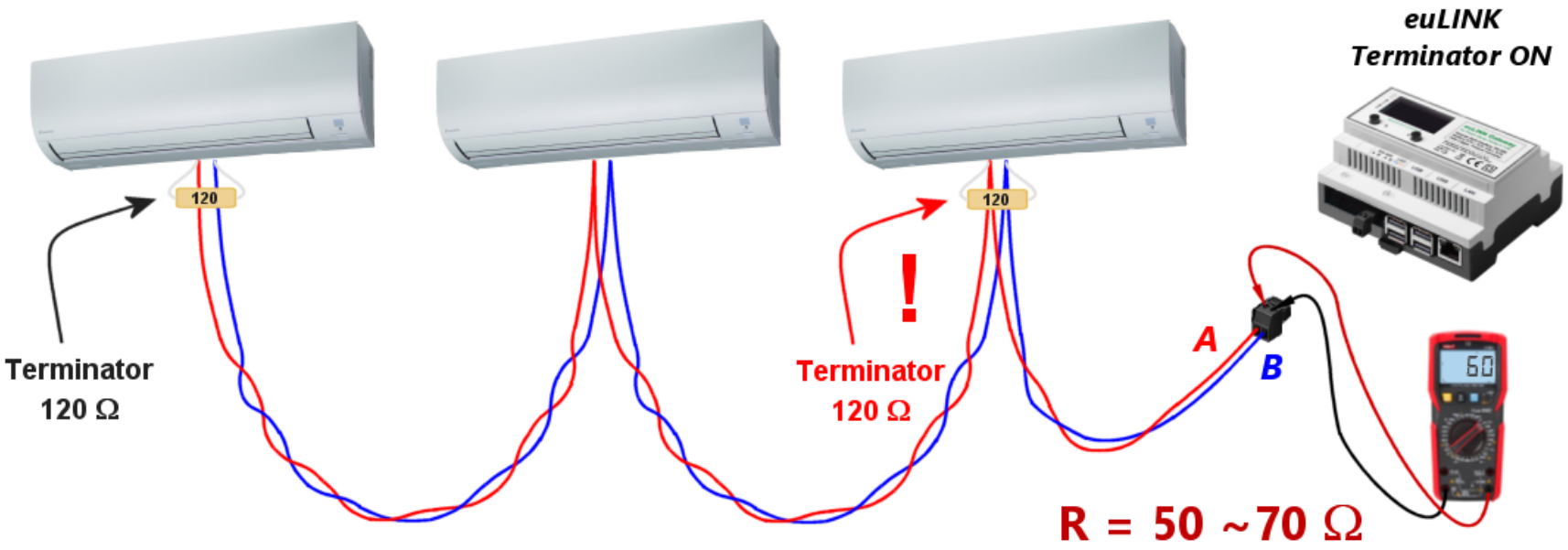
The need to ensure the correct placement of terminators on the bus is therefore not particularly onerous, and the presence of terminators can be extremely useful when looking for the cause of communication problems. This is because it is sufficient to disconnect the MODBUS master (i.e. remove the connector from the RS-485 port of the euLINK gateway) and measure the resistance between terminals A and B on the connector or on the cable with an ohmmeter. After all, such a bus segment disconnected from the euLINK gateway should have a 120Ω terminator at the end, so a measurement at the beginning of this segment should give a result between 100Ω and 130Ω, depending on the length of the segment and the type of cable. However, if the result is within the range of 50Ω to 70Ω, it means that two 120Ω resistors are connected in parallel, so such a measurement result reveals the existence of a redundant terminator within the measured segment. And yet it should not be there if we remembered to disconnect the bus segment from the euLINK gate. On the other hand, a measurement result exceeding several tens of kiloohms indicates the absence of the required terminator at the other end of the bus. A measurement of infinite resistance indicates a break on the bus, and a result close to zero indicates a short circuit. Any such wiring fault can be found by dividing the bus into individual sections and repeating the above resistance measurements for each section.

The following illustrations show the correct installation and possible anomalies and associated measurement results (with the euLINK gateway disconnected).



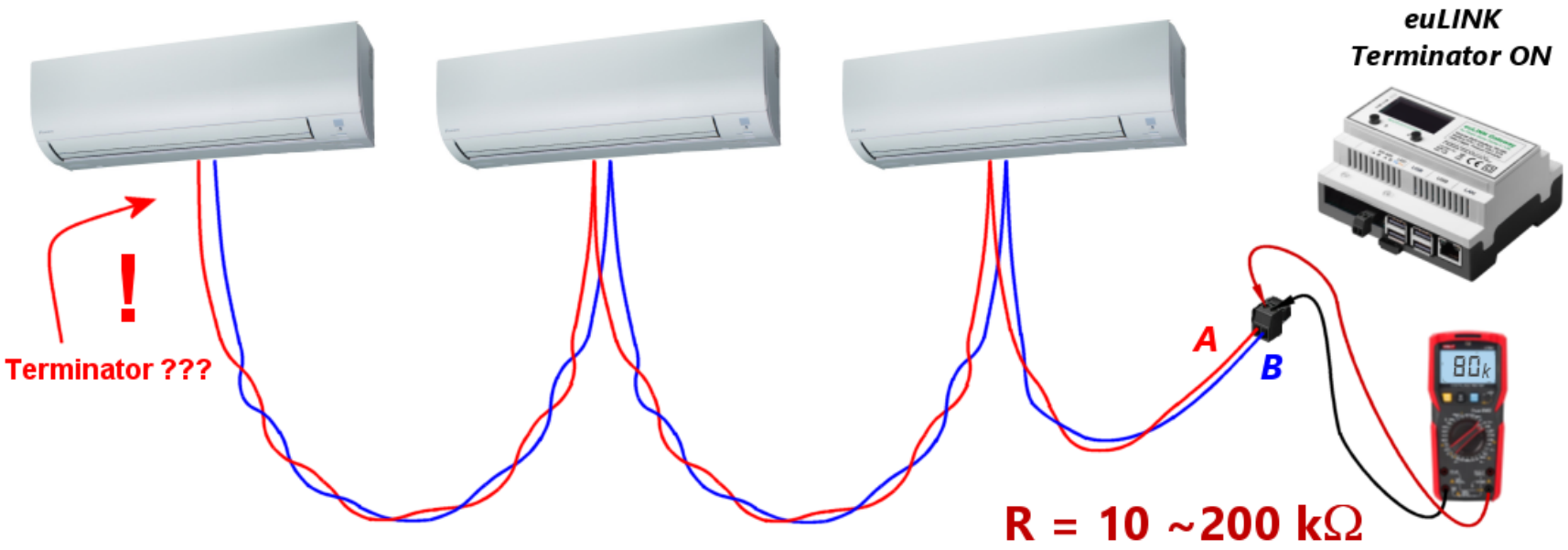
Correct installation, terminator fitted at the end, resistance reading: approx. 110Ω - 130Ω

The measurement result is unlikely to ever indicate an exact value of 120Ω, as the non-zero resistance of the wires is added to the result, depending proportionally on their length. On the other hand, each MODBUS device connected to the bus reduces the resistance reading a little, as their input impedance is also finite.



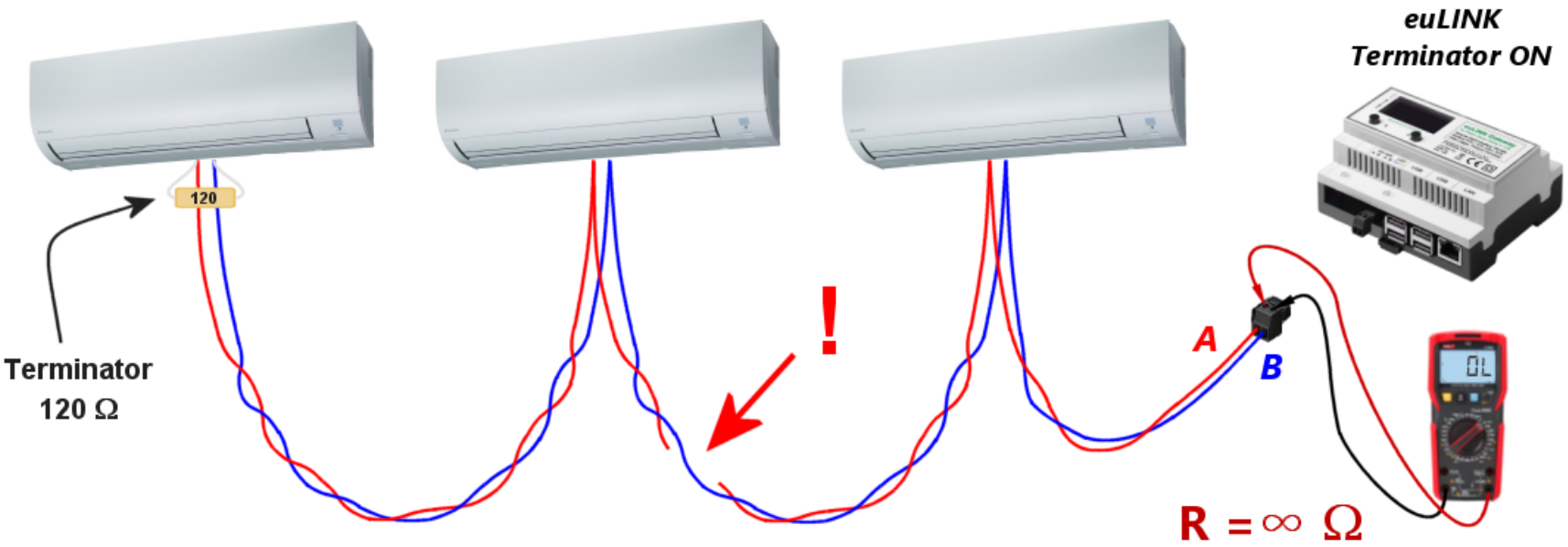
Undesired, excess terminator, resistance reading: approx. 50Ω - 70Ω

Please note that if there are more than 1 redundant terminator, the measurement will show an even lower resistance because all these resistors will be connected in parallel. It is essential to remove all redundant terminators and leave only the one mounted at the end of the line.



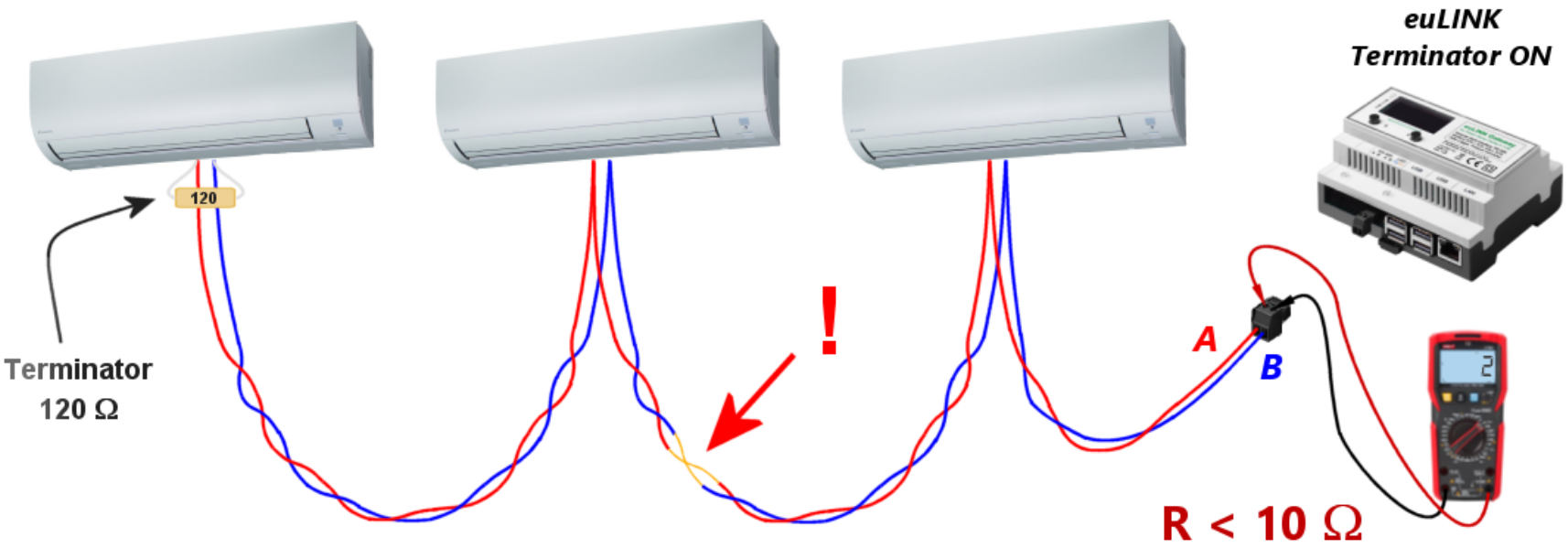
No terminator at the end of the line, resistance reading: tens of **kiloohms** or sometimes hundreds of kiloohms

The result can be in the wide range of tens of kiloohms or hundreds of kiloohms, as it depends only on the input impedance of the connected devices. Of course, the terminator at the end of the line should be switched on and the measurement repeated with the ohmmeter.



Break in wiring, resistance reading: infinity or hundreds of kiloohms

If one or more MODBUS devices are connected between the break in the wiring and the measurement location, the ohmmeter will not read an infinite resistance value, only a few hundred or a few tens of kiloohms. You can use this indication to make a rough estimate of where the wiring is damaged. To narrow down the search area, the same measurement can be repeated at other points in the wiring - preferably where more MODBUS devices are connected to the bus.



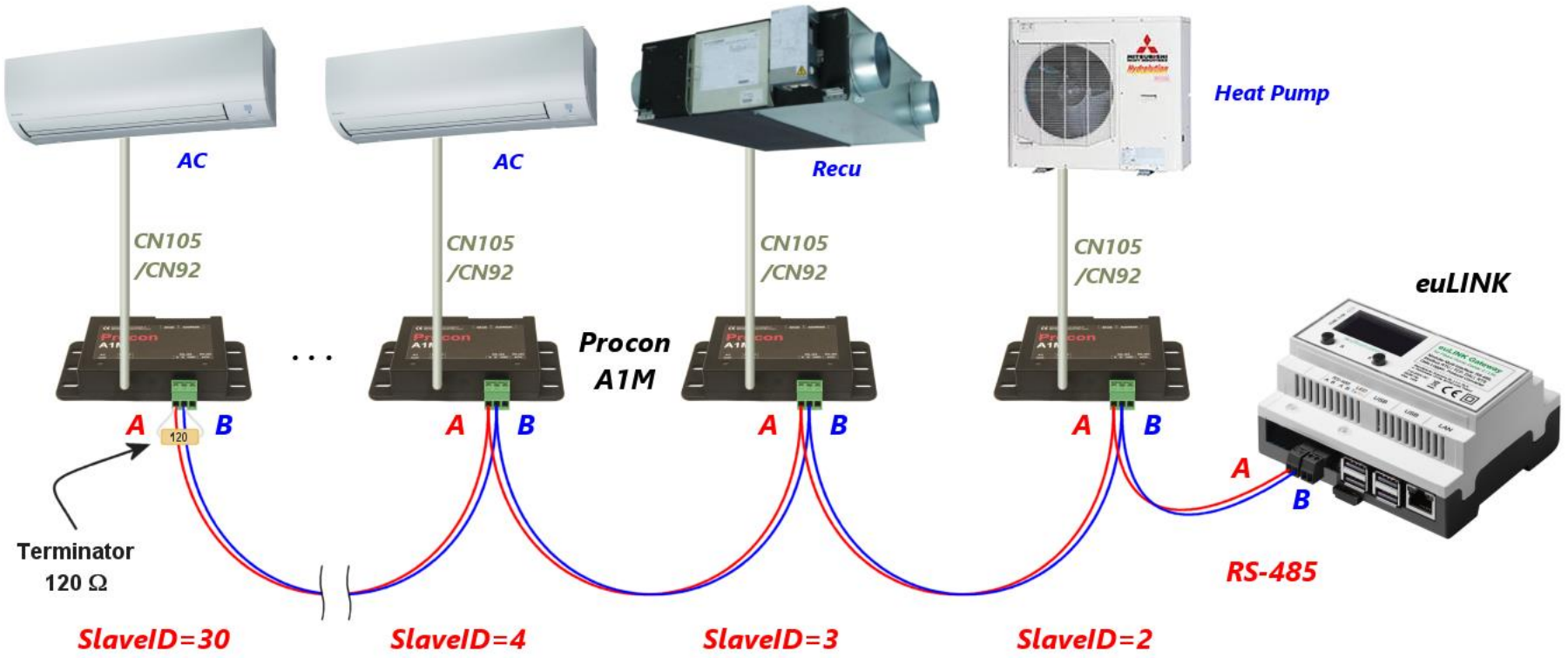
Short circuit in the wiring, resistance reading: several ohms

Each of the above scenarios gives very far apart measurement readings, making it relatively easy to determine the source of the problem. Just remember to take all such measurements on the plug after disconnecting it from the euLINK gateway.

If no faults are found in the cabling and the communication is still unstable or not working at all, you can still try disabling the terminator on the euLINK gateway, which in some cases can increase the signal level and improve the stability of the connection.

For the sake of schematic clarity, all of the above examples discussed communication with one type of integrated device, i.e. air conditioners. However, the described principles of construction and testing of wiring for MODBUS RTU transmission are independent of the type of equipment being integrated. The same phenomena and testing techniques apply to the integration of all HVAC equipment (heat pumps, recuperators, air conditioners), photovoltaics, energy storage, electric car chargers, energy consumption meters, weather stations or swimming pool and sauna equipment. All these enumerated devices can be connected to the same MODBUS RTU bus, as long as there are no more than 30 of them, they have the same communication parameters and differ only in

their MODBUS Slave_ID addresses. The devices can come from one manufacturer or from many different manufacturers, as long as they comply with the MODBUS RTU standard:



If it is not possible to select an identical set of parameters for all devices, the bus can be split into two and a USB⇔RS-485 converter can be used.

e) Slave_ID address collision with another device on the MODBUS RTU bus

A new device connected to an existing MODBUS RTU bus may cause problems if it is set to a Slave_ID address identical to another device already connected to the same bus. This will not cause any permanent damage, but communication with either device will not be possible until the address conflict is resolved. It is therefore worth remembering the good habit of configuring connected devices to a Slave_ID address other than 1, as 1 is the most common factory assigned Slave_ID address.

It may also happen that the connected device is configured not as a MODBUS slave, but as a MODBUS master, and it will then try to send its commands and queries to the devices on the bus. This will certainly prevent proper communication on the bus, as according to the MODBUS RTU standard only one device can operate on the bus as MODBUS Master. The MODBUS RTU protocol does not provide for any synchronisation of commands issued by MODBUS Master devices, so if two such devices are connected, the commands they send will overlap and interfere with each other, as will the responses of confused MODBUS Slave devices. Therefore, the configuration of each newly connected device should be carefully checked.

9. Physical connections for MODBUS TCP

When communicating with devices in the MODBUS TCP protocol, all rules typical for LAN traffic apply. In addition to numerous HVAC/PV/EV devices with MODBUS TCP protocol, converters from other protocols to MODBUS (e.g. MODBUS⇔M-bus and similar) are occasionally encountered, which makes it possible to extend the list of integrated devices with products using protocols other than MODBUS.

A separate category is RS-485⇔WiFi converters, which are used in situations where it is not possible to run RS-485 bus cabling to a device that only has a serial port and no WiFi or LAN port.

a) Communication via wired or wireless LAN

In any case, it is worth ensuring that the IP addresses of collaborating devices are static or that the addresses assigned to them by DHCP are always fixed (e.g. based on the unchanging MAC address of the device). In the euLINK gateway configuration, the IP address of each device is stored when it is first connected and subsequent changes to the addresses assigned by DHCP will result in a loss of communication between the euLINK gateway and the MODBUS TCP device. This phenomenon is analogous to the communication between the euLINK gateway and the Fibaro Home Center controller, which was described in more detail in chapter 7.a) on page 20. The procedure for IP addressing and for looking for anomalies in the LAN is also similar. However, there is rarely a need for traffic analysers because the euLINK gateway and MODBUS TCP devices exchange relatively small portions of information between each other and are thus not very sensitive to congestion and interference on the LAN.

The integrated device can use wireless communication, as the MODBUS TCP protocol does not impose any restrictions here. And most devices (including HC3, Yubii, euLINK, heat pumps, PV inverters, etc.) already have wireless communication capabilities. This is quite an improvement, but only provided that there is adequate WiFi coverage of the device's installation site. While low traffic volumes give good immunity to network troubles, poor WiFi signal coverage makes communication susceptible to all kinds of interference and periodic communication outages. Fortunately, all diagnostic tools and techniques for improving WiFi signal strength are identical to those for normal LANs and the MODBUS TCP protocol does not introduce any impediments to their use.

b) Multi-channel protocol converters, e.g. MODBUS⇔M-bus

Various converters and other multi-channel devices operating in the MODBUS TCP protocol can be found on the market. Examples are MODBUS⇔M-bus converters (for integration of meters equipped with M-bus interface) or MODBUS⇔DMX512 for stage lighting control. These are converters in which each device to be integrated is represented as a separate channel, and the number of supported channels can reach tens or even hundreds. Unfortunately, some of these devices have incorrectly implemented MODBUS TCP session differentiation support, which can make it difficult to identify the correct target devices. If you see such problems, please let us know at support@eutonomy.com and, if necessary, we will quickly add a specially developed mechanism to the euLINK gateway software to bypass such a problem in the specific device.

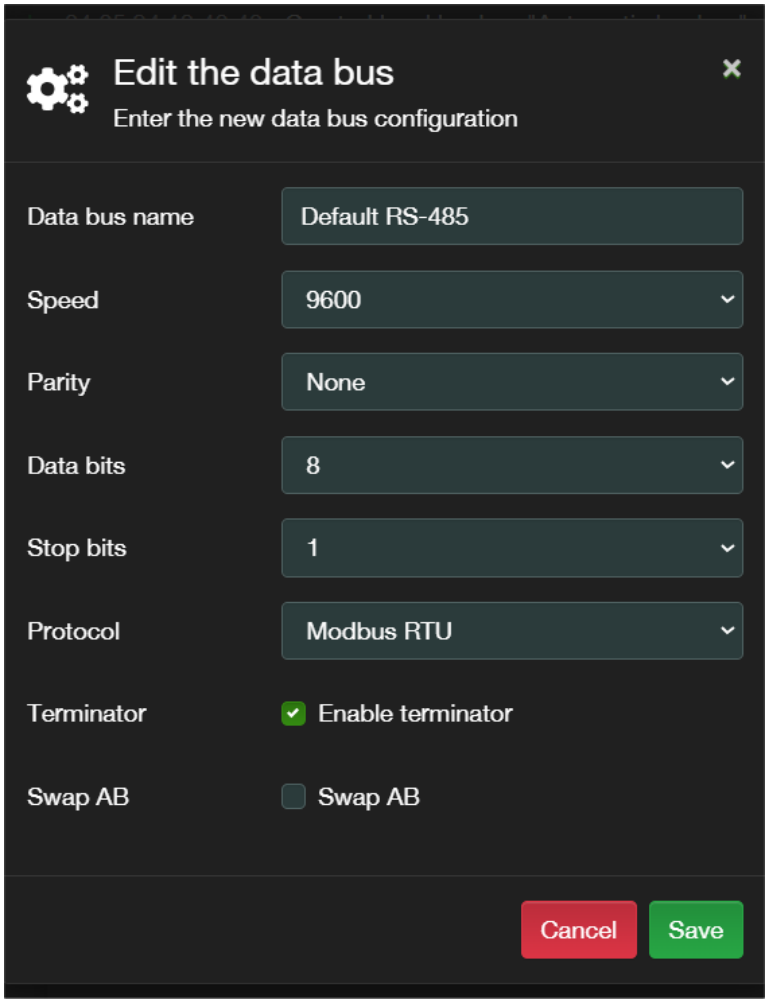
c) Serial to wireless transmission converters RS-485⇔WiFi

Sometimes it is technically very difficult or uneconomic to bring the RS-485 bus cabling to the installation location of the integrated device. This is particularly the case if the system is to be expanded with new devices in the future and it is not possible to lay the cabling, for example, under the plaster so that the cables do not spoil the aesthetics of the room. In such cases, an inexpensive RS-485⇔WiFi converter can be used, which converts the MODBUS RTU transmission into MODBUS TCP and sends it to the LAN wirelessly via WiFi. An example of this is the popular Elfin-EW11A converter, available from many online retailers in Europe and worldwide at a price of around €10 to €45. The converter supports WiFi 802.11 b/g/n communication, consumes only 5W of power and can use a supply voltage in a wide range from 5V to 18V DC. This often makes it possible to draw power from an integrated device, such as an air conditioner. The small dimensions (61 x 26 x 17.8 mm) allow the device to be hidden under the air conditioner housing.

If the euLINK gateway has a template for the integrated device, but only in MODBUS RTU protocol, it is very easy to convert it into a template for MODBUS TCP. To do this, simply use the "Clone template to new device" option, add "TCP" to the template name and select the MODBUS TCP communication method instead of the previous RTU. It is worth checking in the documentation of the RS-485⇔WiFi converter the TCP port number used for communication. The default is port 502, but there are exceptions to this rule. And, traditionally, you should ensure that the converter always receives the same IP address as entered in the euLINK gateway configuration.

10. Configuration of serial ports

a) RS-485 port built into the euLINK gateway



All devices on a single RS-485 bus must have the same communication parameters, i.e. baud rate, parity check, number of data and stop bits, and can (and must!) only differ in their Slave_ID address. The euLINK gateway acts as a master on the RS-485 bus, which polls the slaves using their unique Slave_ID addresses. Slave devices with different communication parameter settings must not be connected to the RS-485 bus, as they could misinterpret commands directed to differently configured devices, and their response could then be unpredictable!

Therefore, identical parameters must be set for each device connected to the common RS-485 bus. In some devices, this is done either hardware-based (e.g. using a DIP switch) or software-based - similar to setting their Slave_ID address, which is described in more detail in chapter 8.a) on page 23. For the euLINK gateway, the parameters are set by navigating to *Menu => Settings => Hardware interfaces -=> RS-485 => Default RS-485 =>* 🔧 *=> Edit the data bus*.

Some MODBUS RTU devices have a wide range of communication parameters, e.g. they have a long list of a dozen possible baud rates (from 1200 to 115200 bit/s) and a large selection of parity control methods[4] (even, odd, none). Other devices have a limited set of parameters, e.g. no choice at all of the number of data bits (they have a 'fixed' 8 bits) or the number of stop bits (they have a coded value of 1). In some cases, the list of baud rates is limited to 2-3 items, sometimes even to only one rate (usually the popular 9600 bit/s), and the list of parity variants is similarly short (usually only "NONE", i.e. no parity control).

The installer should therefore review the documentation of all integrated devices and determine which set of communication parameters is common to all devices.

[4] The parity check allows early detection of transmission errors (already on receipt of each 8-bit data byte).

A unique feature of the euLINK gateway's built-in RS-485 port is the ability to swap the polarity of the A and B terminals. The use of this feature is discussed in detail in chapter 8.b).

b) Increasing the number of RS-485 ports with USB⇔RS-485 converter

If the parameters of the devices to be connected do not have a common part, then 2 groups with common parameters should be selected and 2 RS-485 buses created - ensuring that the devices in each group are connected to the corresponding bus. Sometimes the method of dividing devices into groups also helps when a device is connected but physically difficult to access and its configuration cannot be easily changed. An example would be a weather station previously installed on the roof of a building. Up to 4 USB⇔RS-485 converters can be connected to the euLINK gateway, resulting in a total of 5 RS-485 buses.

Various USB⇔RS-485 converters are therefore readily available for use with the euLINK gateway to increase the number of serial ports to support several MODBUS RTU buses. These converters are inexpensive (costing around 5€) and should not be difficult to find in local online shops. There are also several other reasons why such converters can be useful, e.g. if the number of devices on one bus exceeds 30, it would be better to create a new bus using a USB⇔RS-485 converter and transfer redundant devices to it. The same can be done if the permissible length of one bus segment is exceeded, although it is difficult to imagine a distance of 1200m being exceeded in a domestic environment. However, the most common reason for splitting devices into 2 or more buses is the inability to reconcile their communication parameters. Remember that all devices on a single RS-485 bus must have the same communication parameters, and can only differ in the Slave_ID address. If two or more integrated devices have disjoint sets of communication parameters, then 2 or more buses with different parameters can be configured and the devices grouped accordingly.

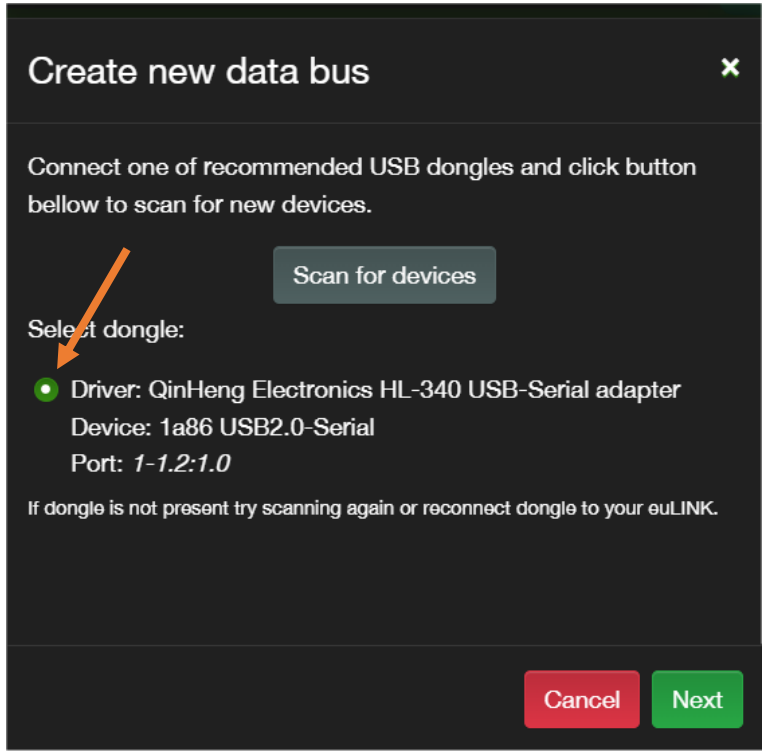
However, the principle of the presence of terminators at the beginning and end of the line also applies to any such additional bus. The converters described often have a terminator permanently built in, without the possibility of switching it off. The answer to the question of whether the terminator is built-in can be provided by a resistance measurement, described on page 31 and the previous page.

It is also worth bearing in mind the additional limitation that it is not possible to swap the polarity of terminals A and B by software, as this is only possible on the built-in RS-485 port of the euLINK gateway. So please carefully check the description of the AB connector, located on the bottom side of the board under the connector.

If you change the USB port to which the converter is connected, the euLINK gateway may lose the converter assignment - especially if there are more than 1, because they are identical. The euLINK gateway should then be restarted, the USB ports scanned for the converter, and once found, MODBUS RTU devices can be rewritten to it. However, it is best to avoid unnecessarily reassigning converters to individual USB ports.

c) Adding new RS-485 ports to the configuration

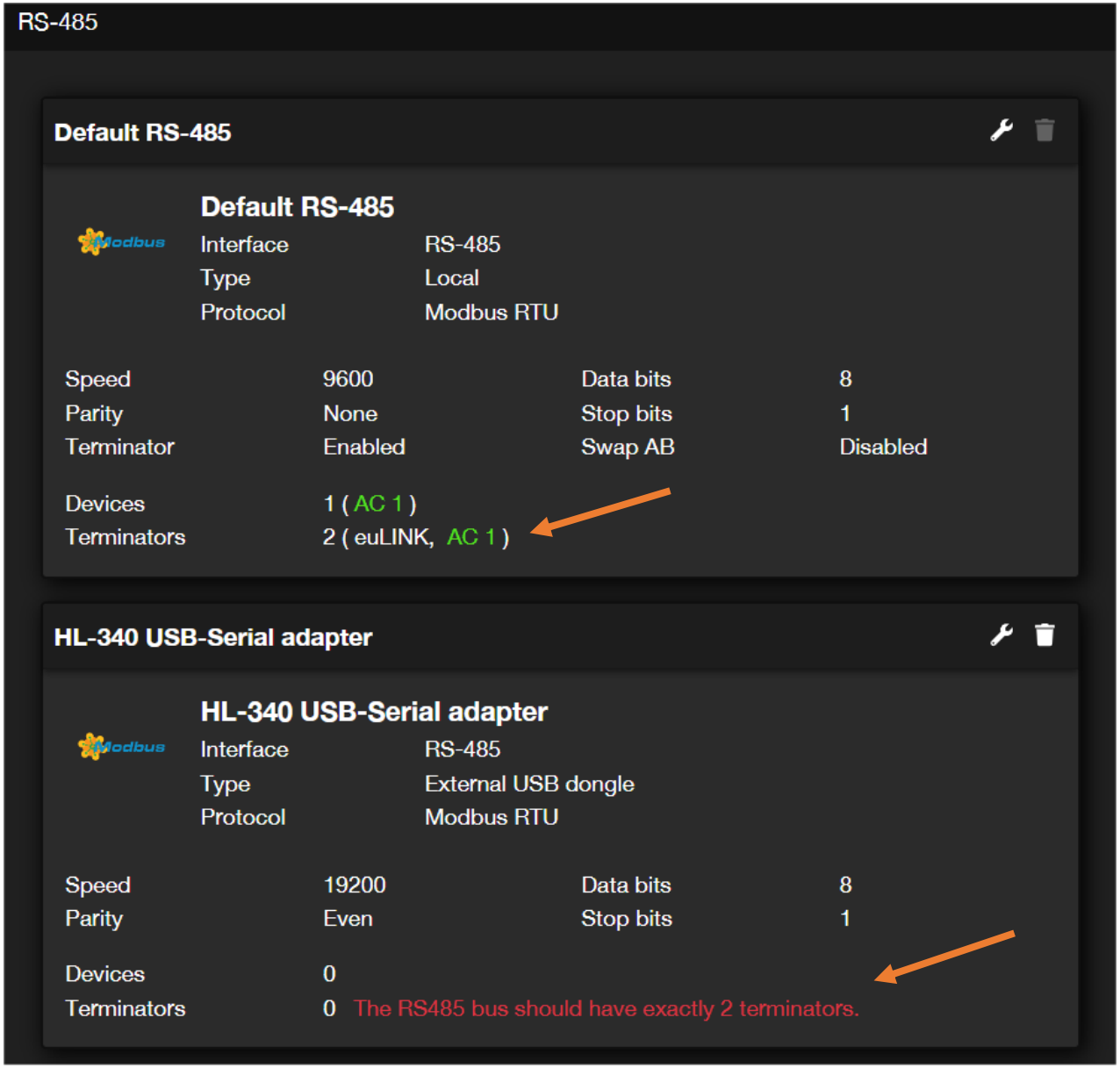
After physically connecting the USB⇔RS-485 converter to the USB port, navigate to *Menu => Settings => Hardware interfaces => RS-485 -=> +Add new data bus*. If the converter is Linux-compatible, it should be automatically recognised and shown in the list so that you can select it with a click and proceed to create the bus and configure it. If the converter is already inserted in a USB port but not detected, rescan the USB ports or possibly remove and reinsert the converter into the USB port.



The configuration of the communication parameters of the USB adapter⇔ RS-485 is generally no different from the configuration of the port embedded in the euLINK gateway.

The exception is the treatment of terminator enable information. In the configuration of the serial port, embedded in the euLINK gateway, marking the terminator causes the internal 120Ω resistor to actually be connected to the RS-485 bus circuit. Thus, this is not only information, but also an actual action. In all other cases - i.e. for other interfaces of the euLINK gateway and for integrated MODBUS RTU devices - this is only information, because euLINK is not able to remotely enable or disable the terminator in the integrated device. And this is not only the case for devices for which the terminator is switched on by hardware - using a DIP switch or jumper, or by directly attaching a 120Ω resistor to the screw terminals. This is because even for devices with a software-enabled terminator, sending such a command could be ineffective due to the still inadequate bus termination. The euLINK gateway is also unable to measure at which points on the bus terminators are installed. The installer must therefore personally check each device and disable the terminator on all intermediate devices and only enable it on both end devices. Depending on what the installer has set or found, the installer should then include the relevant terminator information in the configuration of each interface and each device connected to the bus. This will allow the euLINK gateway to count how many terminators there are on each bus and display an alarming message if the number of terminators is different from 2. However, if such an alarming message is displayed, please do not regard this as a barrier that prevents communication from starting on the bus. It may, after all, be the case that the relevant terminators are in fact enabled, only that information about them has not been entered in the device parameters. However, such a message is always a little worrying and needs to be checked reliably in practice.

By navigating to *Menu => Settings => Hardware Interfaces => RS-485* the installer should receive an up-to-date list of RS-485 ports, together with information on the number of terminators fitted, as shown in the following screen copy:


RS-485
Default RS-485
Default RS-485
Interface RS-485
Type Local
Protocol Modbus RTU
Speed 9600
Parity None
Terminator Enabled
Data bits 8
Stop bits 1
Swap AB Disabled
Devices 1 (AC 1)
Terminators 2 (euLINK, AC 1)
HL-340 USB-Serial adapter
HL-340 USB-Serial adapter
Interface RS-485
Type External USB dongle
Protocol Modbus RTU
Speed 19200
Parity Even
Data bits 8
Stop bits 1
Devices 0
Terminators 0 The RS485 bus should have exactly 2 terminators.

11. Setting up MODBUS device instances

The euLINK template is something like a 'cookery recipe' from which as many devices can be created in the configuration of the euLINK gateway as are needed for each device to represent an actual device in the building. Such newly created devices are called *instances*. Multiple device instances can be created from a single template, e.g. several air conditioners of the same type.

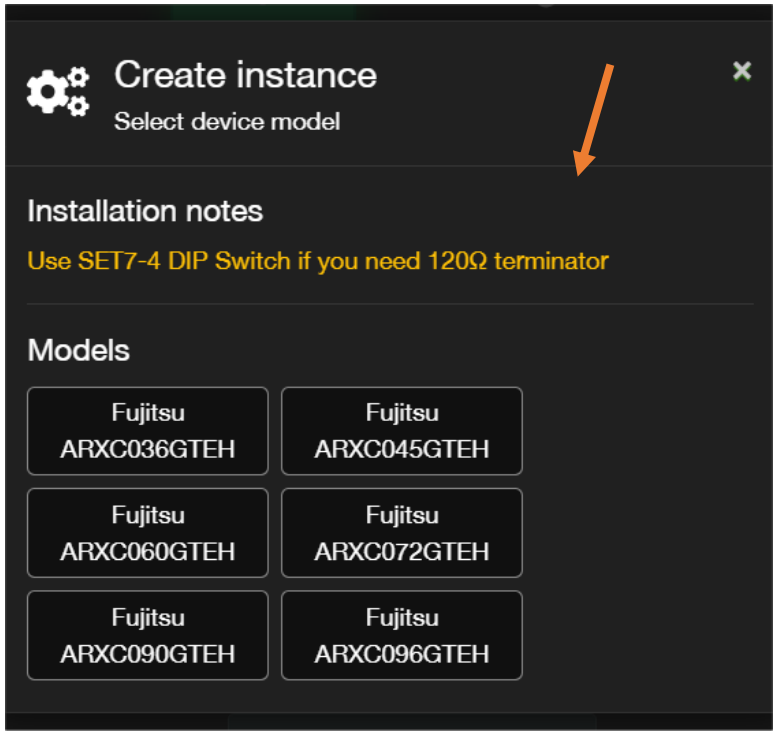
Before creating device instances, it is a good idea to ensure that a list of sections and rooms has already been entered into the euLINK gateway configuration (either by downloading them from the Home Centre or by creating them manually). This is because each new device instance will need to be assigned to the target room right away.

a) Device template selection

New devices are added by navigating to *Menu => Devices => All templates*. From the list of templates presented, select the correct device, or you can narrow down the list by entering the first few characters of the device name or model in the appropriate search field. The quickest search method is usually to expand the list of manufacturers or categories. Next to each template in the list there is a command and icon  *Show details*, which can be used to find out more about the template and make sure it applies to the device you are looking for.

b) Creating a device instance

After clicking on the *+Create instance* option next to the selected device, a window will appear (shown here on the right) in which the corresponding device model can be selected if the indicated template covers several different models of a particular device.



In this window, you may also come across notes, prepared for the installer by the template creator. These are specific, practical comments that can save the installer a lot of time, so  it is better not to ignore them. They may concern, for example, the specific configuration of device parameters, the inclusion of terminators in the device, or troublesome inconsistencies between the translation of the device manufacturer's documentation and reality.

The communication parameters of the device are then configured in the same way as for the RS-485 port parameters of the euLINK gateway.

At this stage, the differences between the two types of MODBUS device organisation become apparent, i.e. between individual devices and hierarchical (group) devices, a division that was further described in chapter 8 on page 23. Actually, the essence of this division is common to both MODBUS RTU and MODBUS TCP protocol devices.

i. Individual device

These are devices that are connected directly to the MODBUS, even if there are more devices of the same type in the building. These can be, for example, heat pumps, recuperators, meters, sensors, even some air conditioners. The RS-485 bus communication parameters are set on each such individual device and a separate instance is created for each device. The communication parameters of each instance in the euLINK gateway must strictly correspond to the parameters set on the individual device corresponding to that instance. Then, all that is left to do is to assign a name and indicate the room and the device is completed and ready for operation.

ii. Hierarchical device

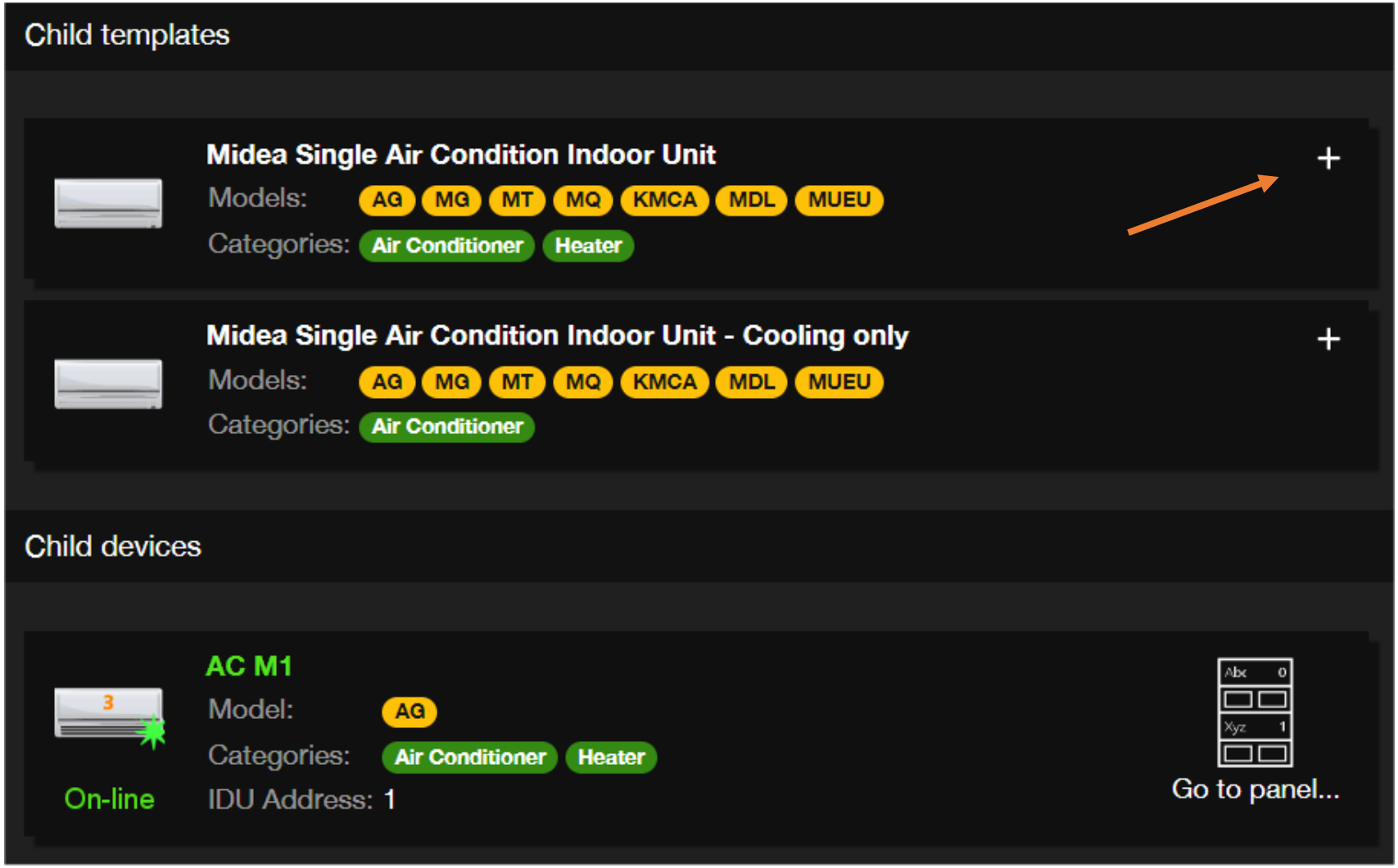
These are group units that are not directly connected to the MODBUS, but to a specialised "collective" device (master, central, main, etc.) and only this device is connected to the MODBUS. A good example of this is large **MultiSplit** air-conditioning systems, where only the „outdoor unit” (ODU) is connected to the MODBUS, and all subordinate air-conditioners (IDU) are only connected to it and managed by it. The list of templates in the euLINK gateway shows hierarchical templates very clearly, because they contain one or more subordinate templates in addition to the master. The command to create an instance first creates only the master device and it is the master device that all communication settings for MODBUS transmission apply to.

Once the master device instance has been created, configured and tested, you can proceed to create instances for the individual slave devices. Their instances are created in the euLINK gateway from within the configuration panel of the master device. Scroll down the configuration panel of the master up to the Slave *Templates* group, where you will see a list of defined slave templates. Then press the '+' symbol to create an instance of the slave based on the selected template. The configuration of slaves is extremely simple, as they have only one parameter - a unique *IDU Address*, which allows the master (ODU) to distinguish between its subordinate IDUs.

If a slave has already been created previously, it will be visible below in the *Slave Devices* group. Next to each instance of a slave device, you will see its IDU Address and current status.

Instances of slaves are also given names and assigned to rooms - like all other instances of integrated devices.

The following screen extract illustrates the appearance of the part of the ODU configuration panel where instances of slaves are created and viewed:



Clicking on the 'panel' icon enters the configuration panel for the slave unit, which is the same as for the individual unit.

c) Configuration of communication parameters

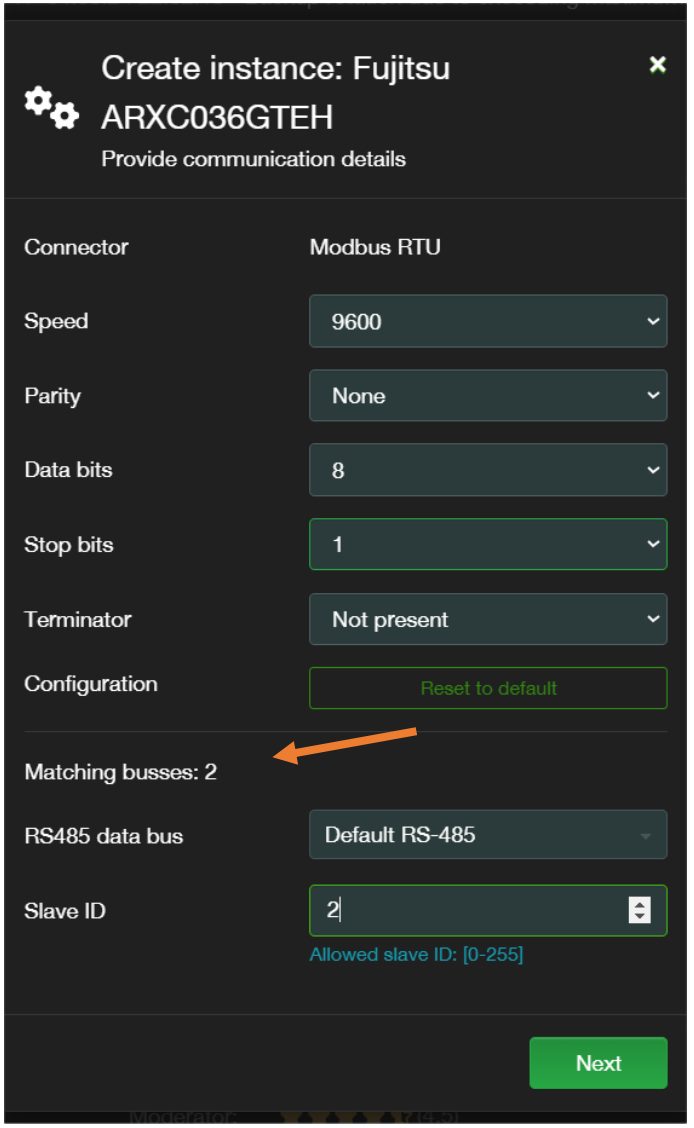
i. MODBUS RTU

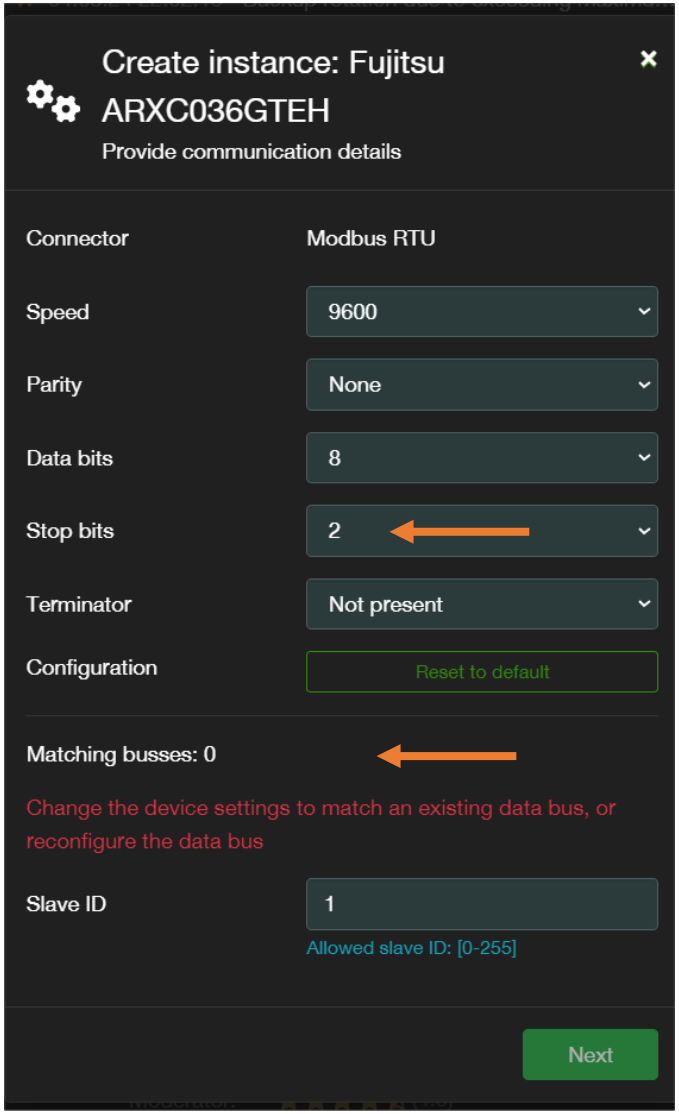
The communication parameters (speed, parity, number of bits) of a device using the MODBUS RTU protocol are configured in the same way as for the parameters of the RS-485 port of the euLINK gateway.

The configuration is fairly easy, but provided there are no parameter discrepancies or address conflicts. This is because once the communication parameters have been set, euLINK indicates whether they match the MODBUS RTU bus connected to the built-in RS-485 port or to any of the additional ports brought in by the USB⇔RS-485 converters. If the configuration of the device does not match any bus, then you will need to go back and check and correct the settings - first of the device being created and then of the bus. If there was a need to change the parameters of the bus, this would also apply to all devices previously connected to it. You would have to remove the association of all devices with the bus, physically change the parameters in the devices, change the configuration of the bus and re-associate all devices with it. Therefore, it is worth meticulously checking and planning this configuration right from the start.

Similarly, the euLINK gateway will protest when the installer attempts to give a device instance a MODBUS Slave_ID address that has already been previously assigned to another device. In such a situation, euLINK will reveal the name and type of the previously addressed device to help avoid conflict.

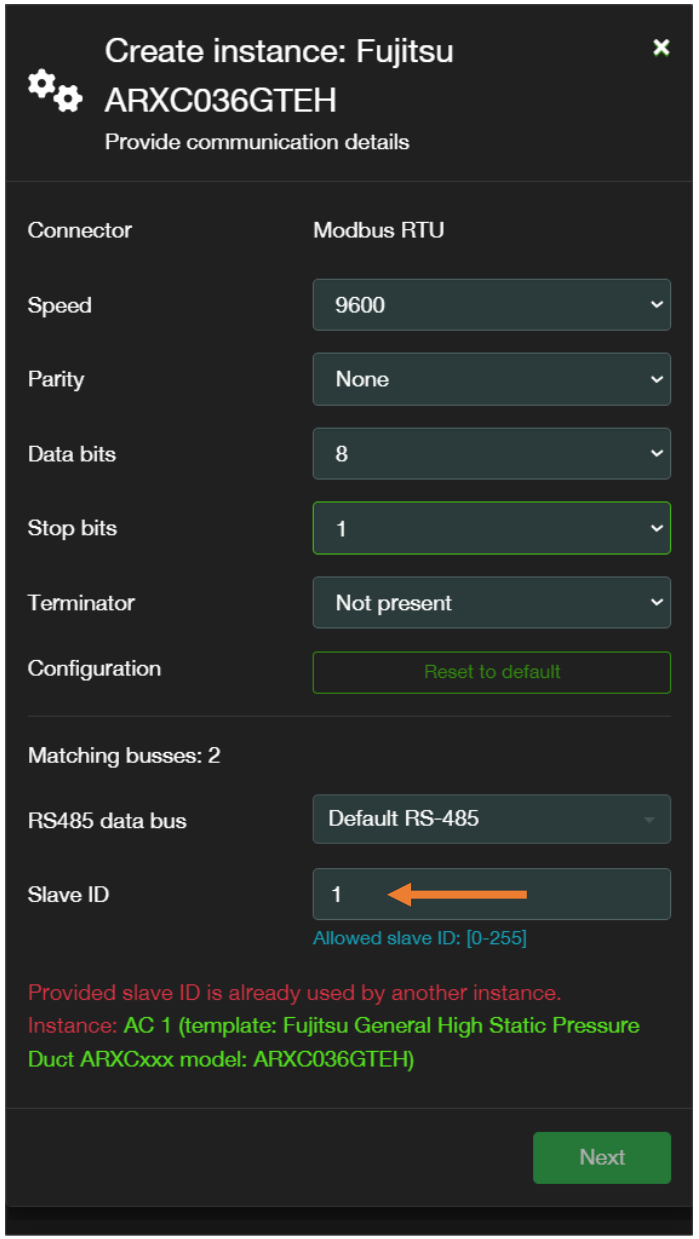
In this way, the euLINK gateway tries to prevent the installer from making the most common configuration mistakes. However, the easiest way to avoid similar mistakes is to prepare and follow a plan early, as described in chapter 5.b) on page 10. Keeping an eye on design assumptions is necessary because neither the MODBUS protocol nor the RS-485 bus standard provide any possibility for automatic setting of communication parameters and Slave_ID addresses. This is a pity, because we would certainly equip the euLINK gateway with the possibility of automated configuration of devices ☺





Parameter mismatch

Slave_ID address conflict



ii. MODBUS TCP

The configuration of an instance of a device equipped with MODBUS TCP protocol support does not hide any surprises. You set the IP address of the device, the TCP port number on which the device listens for connections, and the Slave_ID address (or actually Unit ID, according to MODBUS TCP protocol nomenclature).

Of course, care must be taken to ensure that the IP address assigned to the device never changes, e.g. in the manner described in section 7.a) on page 20.

The default TCP port number for the MODBUS TCP protocol is **502**, but this should always be checked with the device manufacturer's documentation.

In theory, a Unit ID may be needed, because sometimes there are several slave devices running on the same IP address that need to be distinguished somehow. But in recent years a different approach has prevailed, i.e. different TCP port values are used for different devices.

iii. Other protocols

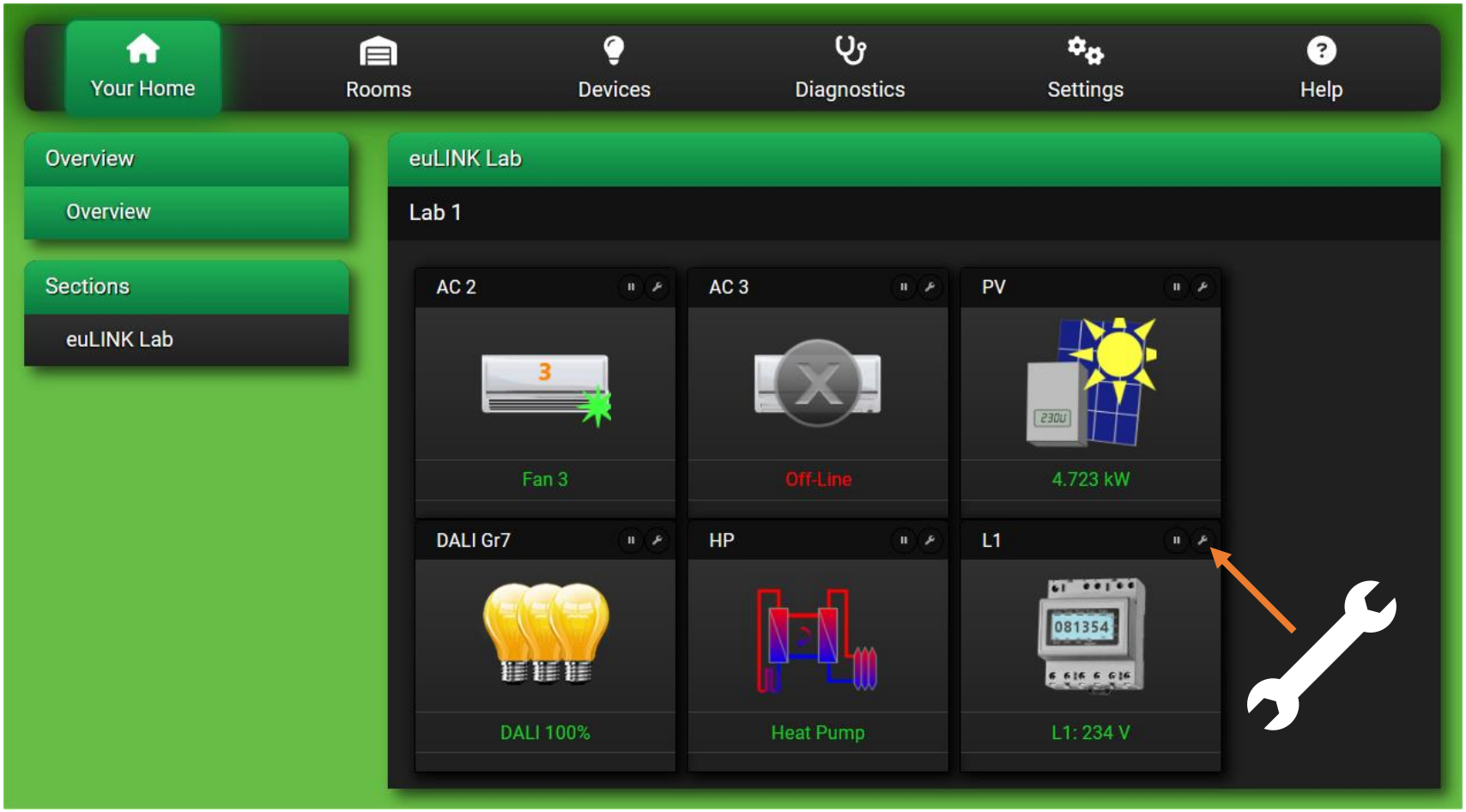
It is possible to implement other communication protocols and there are already several such successful examples in the euLINK gateway software (e.g. RS-485 Raw protocol). However, these forms of communication are much less standardised than MODBUS, KNX or BACnet. Therefore, before designing a system using other protocols, please contact the euLINK gateway support team to confirm the feasibility and timing of such a task.

d) Naming and assigning the device to a room

Naming a new device and assigning it to a room is a satisfactory formality. Of course, both the name and the target room of the device can always be changed later in the device configuration.

12. Testing of integrated MODBUS devices from the euLINK gateway

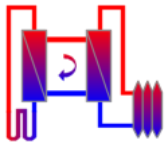
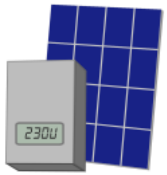
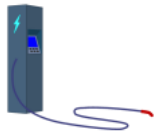
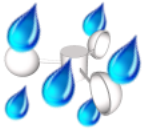
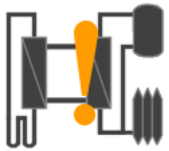
Once a device has been created, the euLINK gateway shows its configuration panel. The same panel can always be opened by navigating to *Menu => Devices* and also in *Menu => Home* by clicking on the key symbol on the 'tile' of the device:



In the panel you can check and change the configuration of the device, you can also view its readings and test the device's reactions to commands. It is a good idea to carry out as comprehensive tests of the device as possible from the euLINK gateway before it is imported into the FIBARO Home Center.

a) Integrated device status tests

The euLINK template should recognise at least 1 „normal" device state and at least 2 „abnormal" states (*no connection* and *failure*). Each state has a separate icon to facilitate a quick assessment of the device's condition. The table contains some example devices with icons for their different states:

	Air conditioner	Heat pump	Recuperator	Photovoltaics	EV charger	Weather station
Normal or standby						
Other normal state, e.g. active						
No connection to the device						
Device reports failure						
The device requests service						

A list of all states and icons for a given device can be seen at the very bottom of its configuration window in the euLINK gateway. The number of 'normal' states is often greater than 1, so that you can quickly show, for example, what the fan speed of the air conditioner is or what phase of the cycle the heat pump is currently in.

The appearance of the icons is continuously updated on the home screen of the euLINK gateway and is also transmitted to the Home Center controller, so it should even be visible in the app on users' smartphones.

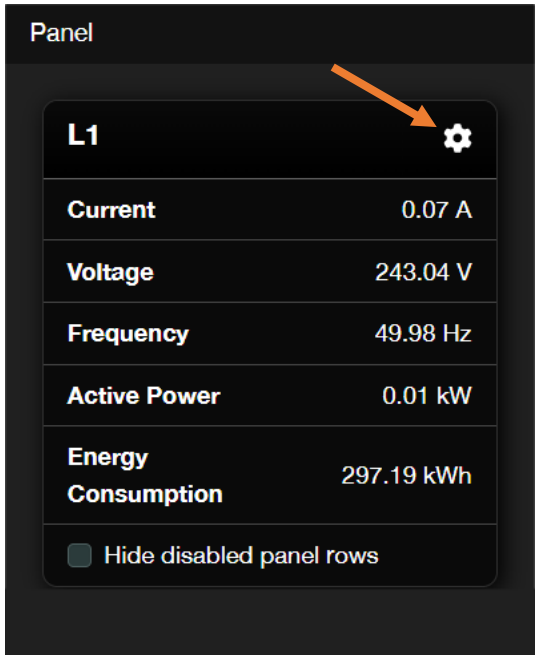
The "*exclamation mark*" icon means that the device requires your attention, e.g. the filters in the air conditioner or recuperator need to be changed. This is not yet a failure, but it is better not to underestimate such messages and to call the service in advance, as the lack of response may lead to a failure of the unit.

If the icon indicates a malfunction or the need for service, the reason for the lack of readings is not a loss of communication with the device. The continuity of the MODBUS should then not be checked, but the device itself. Often, the readings on the euLINK gateway panel show the error code reported by the faulty device, which is particularly appreciated by HVAC installers. This is because if the ordinary user can already communicate this code at the time the fault is reported, the service technician will already be well prepared at the first visit, e.g. bringing an unusual spare part necessary for the repair.

A connection failure icon indicates a communication problem on the path between the euLINK gateway and the device in question. There may be several causes, so it is worth starting the search with the most mundane ones, e.g. check the continuity of the MODBUS RTU bus cabling and the status of the terminators at the beginning and end of the bus as described in chapter 8.d) on page no. 31. Sometimes a mundane error in the factory wiring diagrams of the devices happens and it is enough to swap the wires on terminals A and B of the device to restore communication. Such a test is safe, as the transmitted signals have a low voltage (max. +/-12V) and the devices are protected against polarity reversal - in the event of an incorrect connection, transmission will fail, but no component of the devices will be damaged. To be sure, such a test can be performed on a separate RS-485 bus. Switching the suspect device to a separate bus can be done using a USB⇔RS-485 converter. This is advantageous because then the device tests do not interfere with the communication of the other, functioning devices in the system. A malfunction of, for example, the USB⇔RS-485 converter cannot be ruled out either, so it is worth having a spare converter with you. The power supply of the devices or their communication module (if separate, e.g. battery operated) can also be a problem. If all simple causes are ruled out, the problem is almost certainly an incompatibility of configuration parameters on either side, or setting the same Slave_ID address on two devices. It is very helpful to be able to verify the behaviour on the suspect bus of another previously checked device. In the case of MODBUS TCP connections, the local LAN needs to be verified, as configuration errors occur here too. Although it cannot be ruled out, faults in factory-new HVAC equipment also happen - fortunately extremely rarely. Instead, simple installation or configuration errors are almost 99% of the cause.

However, the cause of the problem must be found and eliminated, because not only can the integration not be completed, but an improperly configured device can interfere with the communication of other devices. If the problem proves more difficult to locate, please read chapter 16 on page 83, where we have gathered all the available sources of information to help with advanced problem diagnosis.

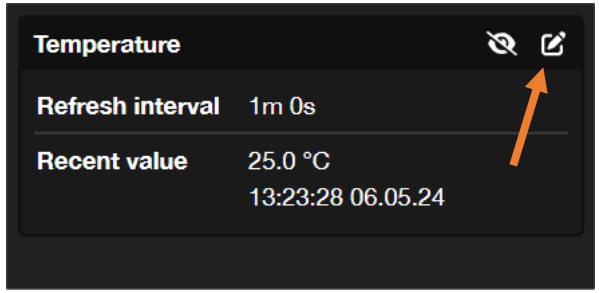
b) Reading tests



If the device is communicating correctly, it is worth reviewing the downloaded readings in its configuration and comparing them with the actual state. In many cases, integrated devices have their own LCD displays or the manufacturer provides a smartphone app, for example. If you encounter an unexplainable discrepancy, please send a description of the phenomenon to support@eutonomy.com.

The readings are presented in the configuration of the device in the form of a panel, the appearance of which is shown on the right in the example of a simple electricity consumption meter. The individual readings occupy successive rows, and each reading consists of the name, value and unit of the quantity read. If the device accepts commands, rows will also appear in the panel, containing command buttons. Some rows are hidden by default, others are optional and, in a particular installation, can be disabled by clicking on the 'cog' icon and selecting the rows to be disabled. Rows that have been disabled will not be visible on the FIBARO Home Center controller side. This applies to both readings and rows with command buttons. Checking the box at the bottom of the panel will hide the disabled rows, improving the idea of how the panel will look on the FIBARO Home Center side and in the Yubii smartphone app. The panel with its numerous readouts and buttons looks a bit like a "*remote control*" on smartphones and is used as such.

It is worth remembering that MODBUS devices are periodically polled by the euLINK gateway, so changes in values will only be observed on the panel after the next reading. When creating device templates, a polling interval of about 1 minute is generally set, which works well for most HVAC/PV/EV equipment. However, if an important parameter changes its actual value much more frequently, the interval for reading this parameter can be shortened. To do this, scroll down the configuration window to the '*Variables*' section, where you will find a list of all readings. The 'tile' of each reading shows the name of the parameter, its last value and the time and date of the last reading. Clicking on the icon opens the window for editing the interval of the read-out, while clicking on the *eye* icon will exclude the variable from the panel and from the read-out. The polling interval can be extended or shortened - to a dozen or even a few seconds. However, it should be remembered that the transmission on the MODBUS RTU bus is not very fast, and also the device itself needs some time to prepare and send a response to the query. If there are a lot of devices on a given bus (and there may be up to 30 of them), the time required to poll all of them may be several seconds. The polling interval should therefore be selected according to the number and specific characteristics of the connected devices. This also applies to devices with the MODBUS TCP protocol, because although the LAN is tens of thousands of times faster than RS-485, the response time of a device can still be considerable.



c) Command tests

Correct readings are a good confirmation that the communication parameters of the integrated device and the euLINK gateway are correctly configured. However, sending commands is a little more complex - both in the MODBUS protocol itself and in the way the integrated HVAC/PV/EV device responds. So you have to use **each** button on the panel in turn for the euLINK gateway to send the appropriate command to the device. If the command is accepted, the respective row on the panel will be briefly highlighted in green. If for some reason the device does not acknowledge receipt of the command or even rejects the command, the euLINK gateway will briefly highlight the line in question in red and display a clearly visible failure message. This may affect one, several or all buttons on the panel. The causes can be as follows:

- The command codes or MODBUS register addresses in the euLINK template are incompatible with the device software - causing the command to be rejected already by the MODBUS controller in the device. It is possible that since the template was created, the device manufacturer has changed something in its software and thus the template has lost its original compatibility. If this is the case, we will ask you to send an email with a description of the phenomenon and the **device model** to support@eutonomy.com. Accurate information about the device model will enable us to reach its current documentation and make changes to the template.
- The model of the integrated device is not fully compatible with the list of models for which the template was prepared - and is, for example, a successor or a model from a neighbouring device family of the same manufacturer. We will also ask you to email us and try to quickly prepare a template variant for the new model.

If the euLINK gateway has highlighted the button in green, confirming correct delivery of the command, then check that the device has indeed executed the command. This is because it occasionally happens that, despite correct communication, the device does not respond at all or responds differently to the description of the button on the euLINK gateway panel. Several reasons for this behaviour are possible:

- Incompatibility of command definitions in the euLINK template with the device software - meaning that commands are not understood by the device and are ignored. As before, this may be due to changes in the software of newer devices of the type in question and a minor correction of the template will be required.
- Sometimes a template is created for an entire family of equipment, but not all its models have all the functions described in the documentation built in. For example, the documentation may describe how to set the angle of the fins in the air conditioner, or to turn fanning on and off with them. This is a very useful function in wall-mounted and some ceiling-mounted air conditioners, but generally completely unavailable in so-called ducted air conditioners, which are fixed deep in the ductwork and are structurally devoid of fins at all. For such models, the fin control commands will be ignored by the unit. In such cases, it is best to hide on the panel of the euLINK gateway the line or lines that contain the command buttons not supported by the model in question.

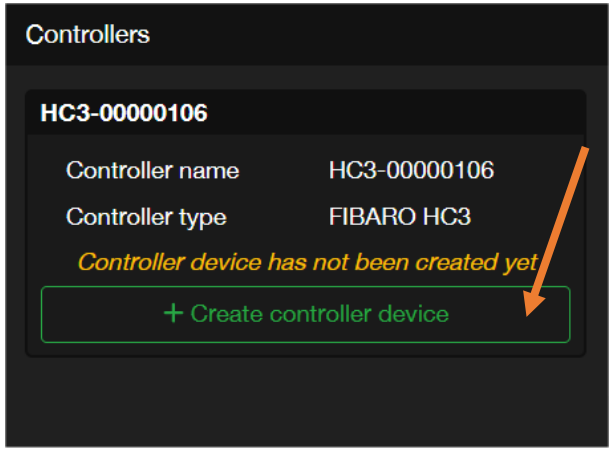
- The response to some commands is immediately clearly visible (e.g. switching the device on or off), but there are also commands with effects that are difficult to see immediately (e.g. switching on ECO mode, changing the temperature setting). If such a command has been issued and the response of the device is not visible, and the reading status on the euLINK gateway panel does not change, then it is necessary to wait about a minute until the next reading. If the expected change of reading is shown on the panel, it means that the command was de facto accepted, but only euLINK asked for its effects too early - when the device had not yet had time to change its state and was not yet ready to confirm the change. A minor correction of the template for the euLINK gateway would then suffice, so it is customary to ask for an email describing the phenomenon to support@eutonomy.com.

d) Remote testing

Particular attention should be paid to tests performed in remote access mode to the euLINK gateway or to the FIBARO Home Center controller. There are no contraindications to testing information readings from devices. But the testing of the issuing of commands should only take place when the installer or device owner is physically present on site. HVAC/PV/EV equipment produces or consumes significant amounts of energy (electrical or thermal), sometimes calculated even in kilowatts. When starting up or changing the operating mode of these units (especially newly installed ones), serious faults may become apparent, which, when processing such large energy streams, may prove to be the start of a dangerous fire. Regardless of who performs the remote test - the installer or (at his request) the staff of the euLINK gateway manufacturer - this must be done with the knowledge and consent of the person on site, who knows how to physically shut down the device under test and where the fuses are located to protect the power supply circuit of this device.

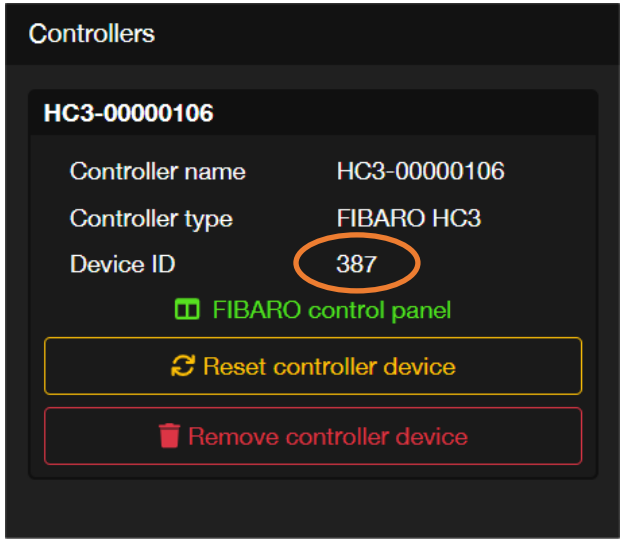
13. Import of a MODBUS device into the FIBARO Home Center

If all the above tests have confirmed operation as expected, you can now import the device into the FIBARO Home Center controller.



Scrolling down the MODBUS device configuration window will reveal the '*Controllers*' section, with a green '*Create Controller Device*' button.

A few seconds after pressing this button, the MODBUS device should already be visible on the FIBARO Home Center configuration page. But before leaving the euLINK screen, it is worth noting the number circled on the right. This is the identifier assigned by the FIBARO Home Center to



the newly created object and, in our example, it is **387**. Knowledge of this number will come in handy in a moment.

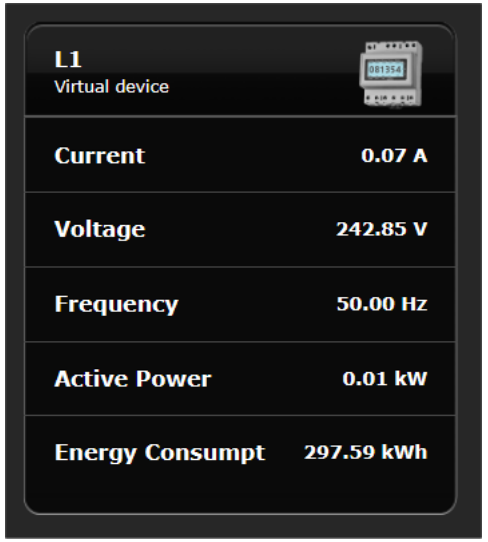
The yellow '*Reset Controller Device*' button is used to restore the device configuration in Home Center to the state it was in immediately after it was created. This function is helpful if someone has changed the device configuration in the controller (e.g. name, room assignment, QuickApp code) and it needs to be restored to its original state.

The red '*Delete Controller Device*' button deletes everything that the euLINK gateway has contributed to Home Center in connection with a particular device. There can be a surprisingly large amount of this information (including status icons, the device object, a dozen or so global variables), so if you need to delete a device, it is not worth doing it manually from the Home Center controller's configuration side - it is better to do it on the euLINK gateway's side, because then it will automatically clean everything up after itself.

On the other hand, if there is ever a need to definitively remove a device from the euLINK gateway, it is essential to use the red "*Remove controller device*" button before doing so, because once a device is removed from the euLINK gateway, automatic "cleaning" of all objects brought into the Home Centre will no longer be possible.

The import of a MODBUS device is possible both to the FIBARO controller from the older HC2 / HCLite family and to HC3 / HC3Lite / Yubii, but the two implementations are slightly different.

a) Import into **HC2 / HCLite** as a "*Virtual Device*" - readings and commands



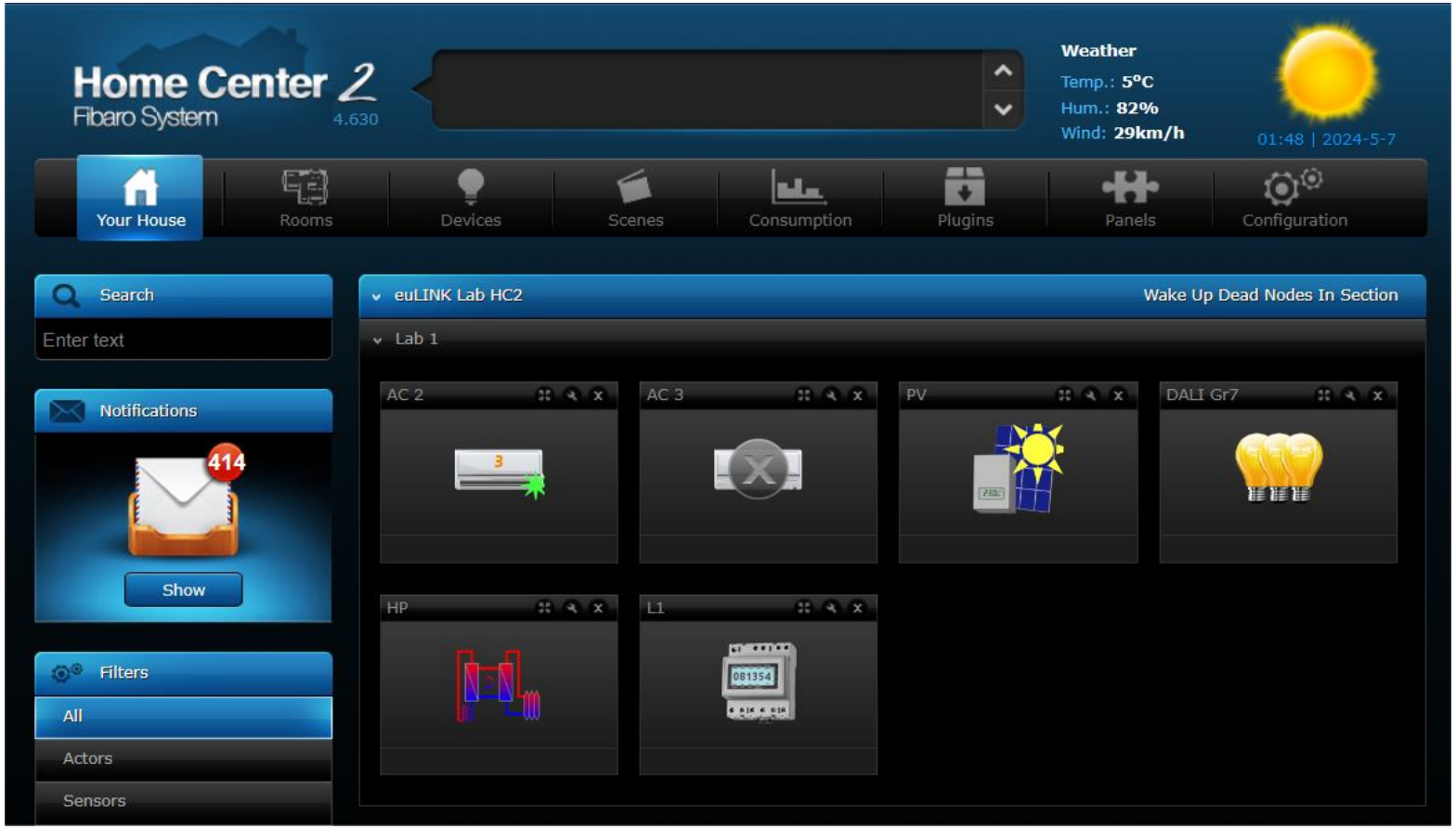
The integrated devices are brought into the HC2 environment via the euLINK gateway as '*Virtual Device*' objects.

The device panel on the HC2 side looks identical to the panel on the euLINK gateway. The euLINK panel, as well as the entire configuration screen of the euLINK gateway, was in fact modelled on the appearance of HC2 (with the knowledge and consent of FIBARO). This was intended to make the configuration of the euLINK gateway easier to navigate for installers familiar with the HC2 environment.

All individual line labels and button descriptions are brought in in the language to which the HC2 controller is set. However, they are brought in as static texts, so if the language is changed in the HC2 configuration, the texts will remain in the previous language in force at the time of device import. However, it is sufficient on the euLINK gateway side of the device configuration to press the yellow button "*Reset controller device*" in order for the euLINK gateway to check the language of the HC2 and infer the corresponding labels and button descriptions in the current language.

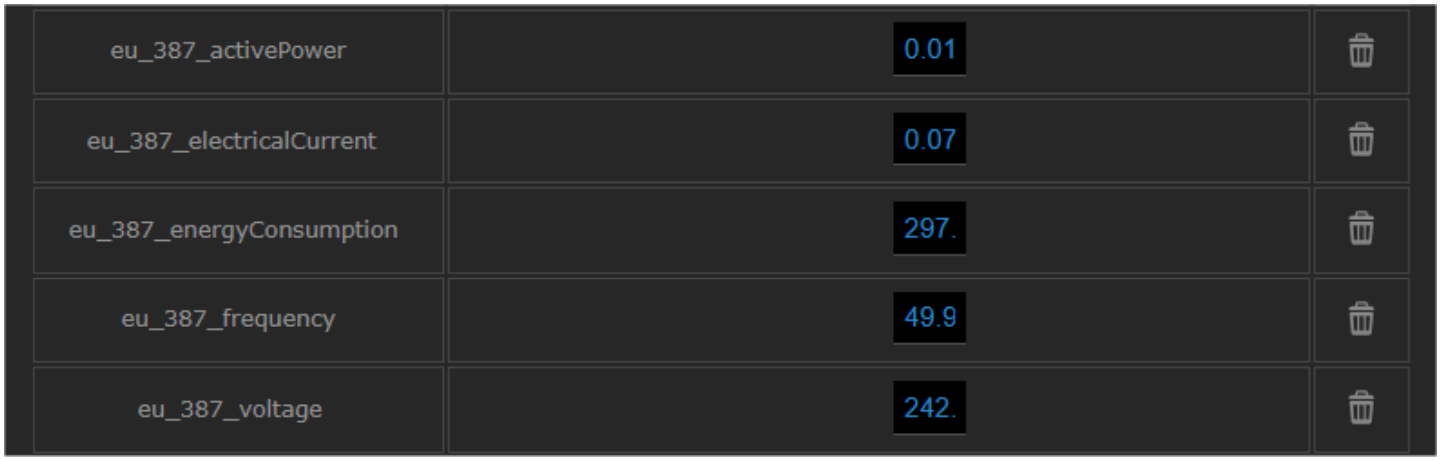
The principle of selecting the current controller language for import also applies in HC3.

Familiar to us from the previous example on (page 50) devices when brought to the HC2 controller look as follows:



The possibilities for exchanging data between different virtual devices and also between scenes were very limited on the HC2 platform. In fact, the only effective solution was to use global variables. Therefore, in addition to importing the device panel, the euLINK gateway brings in a group of global variables storing the values of all parameters of the integrated device. These global variables are constantly updated by the euLINK gateway with the actual readings, so the values contained in them can be used in block and LUA scenes for numerical calculations and for control, should the installer intend to create such scenes to integrate the entire system.

In order to use global variables, you will need to know the Device ID in HC2. We read and noted this information when we imported the device from the euLINK gateway into HC2 (in our example it is **387**). In the list of global variables in the HC2 variables panel, you should see several variables with names starting with "*eu_**387**_* ", containing the values of the individual MODBUS device parameters:



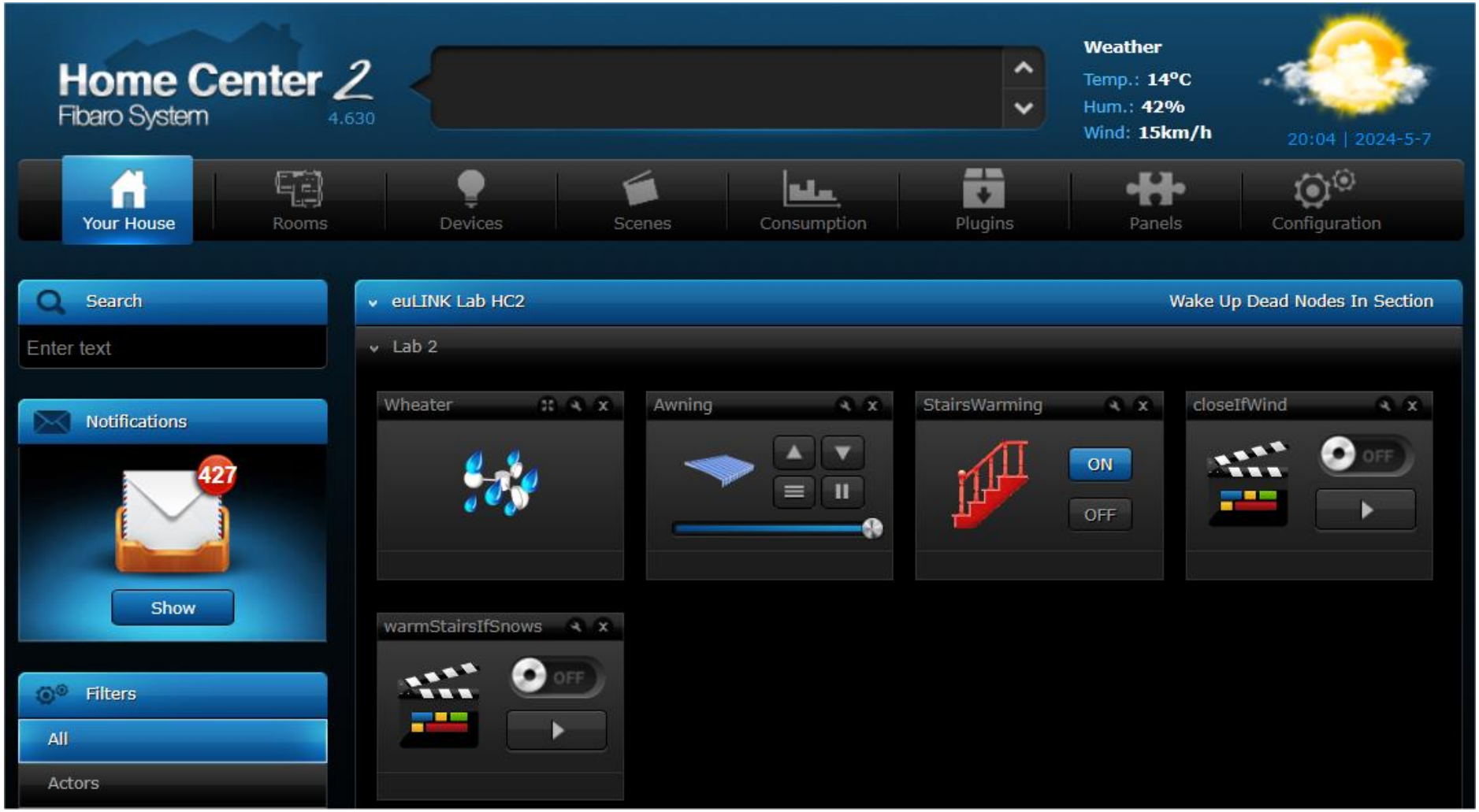
In LUA scenes in the HC2 environment, the *getGlobalValue* function can be used to read the value of a global variable. For example, reading the voltage value from the example counter requires the following construction. It is worth remembering that all global variables are stored as text in the FIBARO Home Center environment, so if the read value is to be used for numerical calculations (e.g. for comparison with a constant value), the retrieved global variable should be converted into a numerical value using the tonumber() function:

```
local L1_voltage = tonumber(fibaro:getGlobalValue("eu_387_voltage"))

--fibaro:debug("L1 voltage = " ... L1_voltage ... " V")
```

Sometimes euLINK needs to add a few characters (numbers or letters) to the end of a global variable name to make the name unique. Therefore, it is always a good idea to check in the list of variables what the name of the variable you need is in that particular installation. You should do the same if you are transferring a positively verified scene in LUA to another Home Center controller.

Creating block scenes using global variables is relatively straightforward. Let's use an example of an installation where there is a local MODBUS outdoor weather station, a garden awning and an electrically heated porch staircase. The HC2 screen then looks as follows:

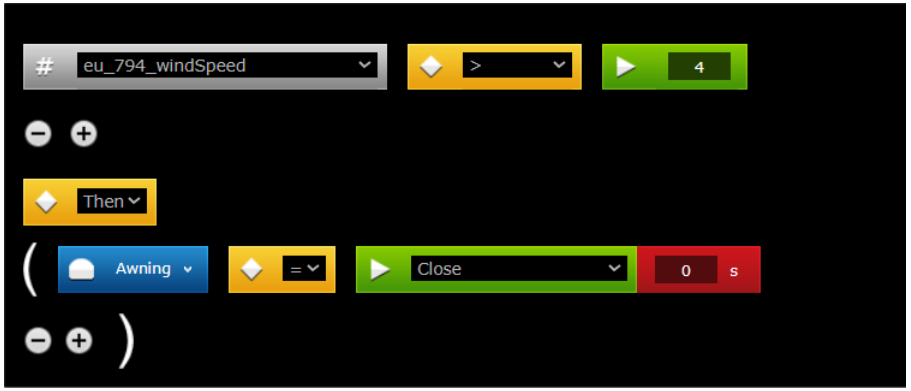


Awning manufacturers recommend that an awning should be retracted if wind speeds exceed 5m/s. Videos from CCTV cameras can be found online showing the destructive effect of a local glare on such an awning, acting as a sail. If the house is truly 'smart', in the absence of the occupants, the awning should automatically be retracted by an appropriate scene.

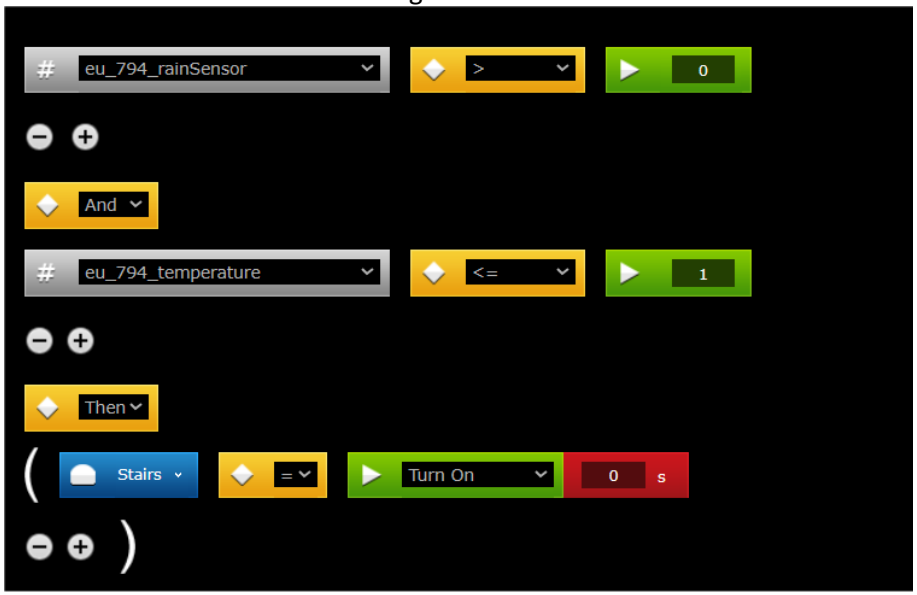
Icing on the entrance stairs can be dangerous, but it can be prevented by heating the stairs with a low-power heating cable (of the order of 10-15W/m). Not all year round, of course, and not even all winter - you only need to switch the heating on when it snows. When the snow stops falling, the heating can be switched off, as no ice will form even if it is very cold. This can be taken care of by a simple block stage, or rather two, as one can switch on and the other **must switch off the** heating. If this second stage were missing, the staircase would be heated 365 days a year, which the investor would only notice when he received his electricity bill. Greater care is therefore required when integrating higher-power devices!

The nearest weather server may be tens of kilometres away and have different data, so it is better to entrust these duties to your own external MODBUS weather station. Block scenes using it might look like the following:

Retract the awning if variable *eu_794_windSpeed* exceeds 4m/s



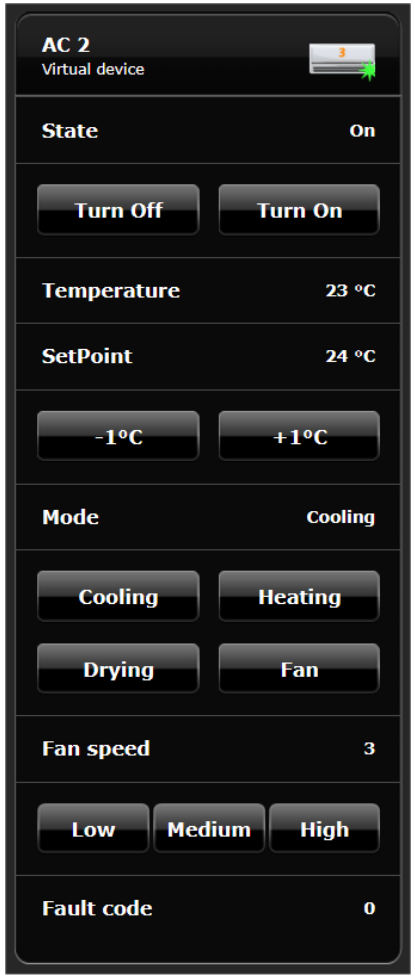
If the variable *eu_794_rainSensor* > 0 (precipitation is in progress) and the temperature has dropped below +1°C (so it is snowing), turn on the heating of the stairs



The trigger for the right scene should only be the variable *eu_794_rainSensor*, not *eu_794_temperature*, because then the scene would be triggered every time the temperature changed - even by 1°C - all year round. And it would be sufficient to wake this scene only when precipitation starts.

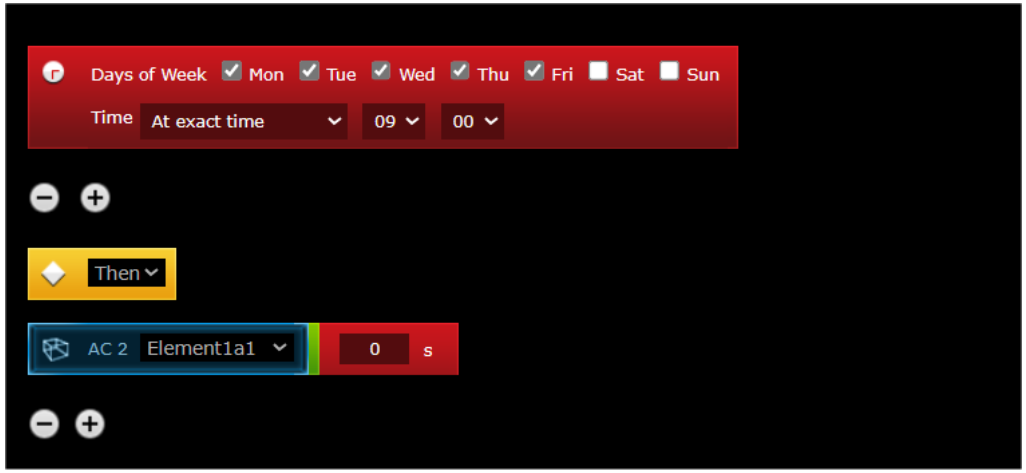
Communication with the MODBUS weather station is somewhat easier, as it is a "read-only" device. And, at least in the current software version, it is not possible to send a command to the weather station that would affect the weather (e.g. "Make it stop raining immediately!").

However, most HVAC/PV/EV devices accept a variety of commands and scenes in Home Centre should be able to enforce these commands. Unfortunately, in the HC2 environment, devices imported by the API as 'Virtual Devices' can only be given commands in one way - 'virtually' by pressing buttons on their panel.



This can be done from both the block scene and LUA, only you need to know the identifier of the specific button. You cannot use the visible description of the button, because it can change and, moreover, is different in every language. Button IDs can be read by opening the "*Advanced*" tab in the virtual device configuration. In general, the button ID contains the row number on the panel and the button number on the row. The row and button numbers are counted from zero, so in the example of the air conditioner panel shown on the left, the '*Off*' button has the identifier **Element1a0**, the '*On' button* has **Element1a1** and the '*Cooling' button* has the identifier **Element6a0**.

Knowing the identifiers of the buttons, you can build a scene that sends any command, e.g. turns on the air conditioner at 9 a.m. each working day: ->



However, the above identifiers only apply to block scenes. For LUA scenes, the HC2 controller uses button numbering as an object on the panel. An object is both a single button and a row with a label (without buttons). The numbering of objects starts from 1, so the "*Status*" row is numbered 1, the "*Off" button is numbered* 2 and the "Switch *on*" button we need is numbered 3. In a LUA scene, the command to switch on the air conditioner would therefore be implemented using the following function:

fibaro:call(797, "pressButton", "3")

where the number 797 is, of course, the virtual device identifier assigned by the HC2 controller during import.

So, if you integrate the HVAC/PV/EV unit with the Home Centre 2 system, you can get all the readings and send any command to the unit.

b) Import into **HC3 / HC3Lite/ Yubii** as "QuickApp" - readings and commands

If a device imported as a QuickApp is assigned to one of the categories that the HC3 family controllers are familiar with, it will be possible to use HC3's built-in support for elementary functions for devices in that category. For example, an HC3 controller 'knows' how to send '*On*' and '*Off*' commands to a device in the **Binary Switch** category, how to change the temperature of a device in the **Thermostat** category, and how to plot energy consumption from the readings of an **Energy Meter** device. These elementary functions become easily accessible in block scenes, and the **Thermostat** is even available in zone heating and cooling schedules. The problem, however, is that devices assigned to a category are, as a rule, single-function, whereas actual HVAC/PV/EV devices are always multifunctional and cannot be limited to a single category. Fortunately, the architecture of the HC3 platform allows the creation of hierarchical devices, with one **parent** device and several **descendant slaves**, each of which can belong to a different category. Therefore, when importing a complex device, the euLINK gateway creates a hierarchical group of QuickApp objects, of which **the Parent** contains the control panel of the device that we are already familiar with, and each subordinate object (**Descendant**) represents one elementary function of the complex HVAC/PV/EV device. The table below illustrates what types of sub-objects can be part of a single composite HVAC/PV/EV device:

Equipment category	Air conditioners	Heat pumps	Recuperators	Smart meters	Photovoltaics (PV)	Electric vehicle (EV) chargers	Energy storage facilities	Thermostats	Weather stations	Lighting sensors	Relays	SPA
Binary sensor		Option	Option		Option							
Binary switch	Option	Option	Option			Option		Option			Yes	Option
Colour controller												Option
Electric meter		Option		Option	Option	Option						
Energy meter				Option	Yes		Option					
Gas meter				Option								
Humidity sensor												
Multilevel sensor	Option	Option	Option	Option			Option		Yes	Yes		
Multi-level switch	Option	Option	Option			Option						Option
Power meter	Option			Option	Option	Yes	Yes					
Rain detector									Yes			
Temperature sensor	Yes	Yes	Yes						Yes			Yes
Thermostat auto	Option							Yes				
Thermostat cooling	Yes							Option				
Thermostat heating								Option				
Water meter				Option								
Wind sensor									Yes			
Enumerated switch	Option	Option	Option		Option	Option	Option	Option				

All sub-objects are imported into the same room to which the **Parent** is assigned, and their name contains a prefix associated with the **Parent's name**. Not all optional sub-objects are associated with every appliance, but there can still be quite a few - especially for air conditioners, heat pumps and some smart meters (e.g. mains parameter analysers).

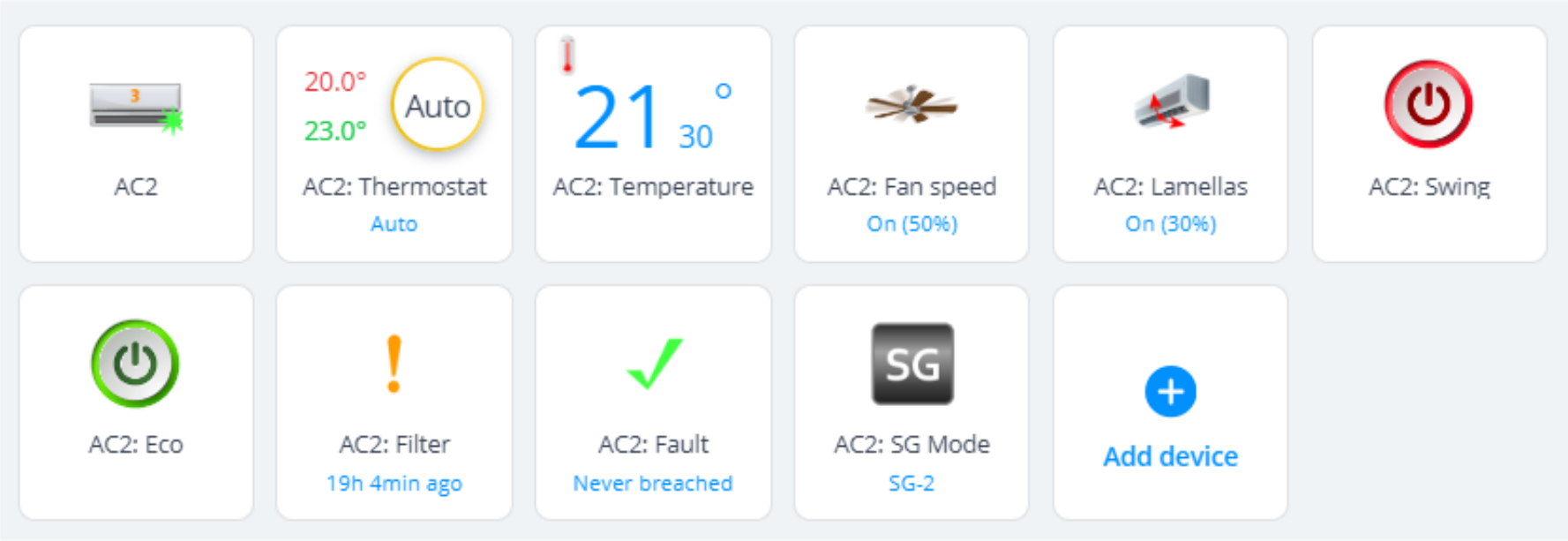
The following table describes examples of **air conditioner** functions and how they are implemented using QuickApp subobjects:

Function	Subordinate object	Required?	Comments
Setting the target temperature	Auto thermostat	Yes	Commands to switch on heating, cooling or 'auto' mode are implemented by the euLINK gateway as a sequence of commands (switch on, and after a second, set the desired mode).
Switching the air conditioner on and off	Binary switch	Optional	The commands to switch the air conditioner on and off are already implemented by a **thermostat-type** object, so there is rarely a need to add a separate object.
Display of room temperature	Temperature sensor	Yes	
Mode of operation	Enumerated switch	Optional	The **thermostat** facility does not have the option of enabling additional air conditioner modes (e.g. dehumidification or ventilation), so a separate mode switch is useful
Fan speed	Multi-level switch	Yes	Slider (from minimum to maximum speed)
Fins angle	Multi-level switch	Optional	Not every air conditioner has built-in moving fins (e.g. duct air conditioners generally do not)
Fanning with slats	Binary switch	Optional	
“ECO” mode	Binary switch	Optional	Not every air conditioner has an "ECO" mode
“Smart Grid Ready” mode	Multi-level switch	Optional	Described in more detail when discussing heat pumps (page 76)
Indication of dirty filter	Binary sensor	Optional	Can initiate notifications
Alarm (fault) signalling	Binary sensor	Yes	

To enable the implementation of all the above objects, the HC3 controller software must be version **5.142** or higher. If the software version is lower, the euLINK gateway will ask you to upgrade the HC3 before importing the device.

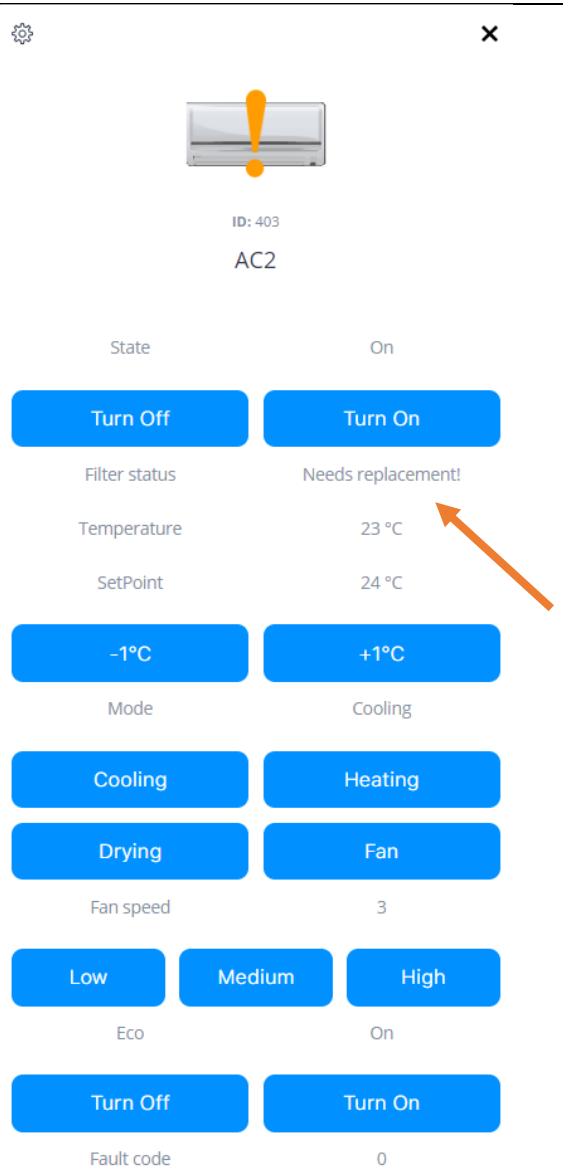
Such an example air conditioner, when imported into the HC3 controller, could therefore look as follows:

i. An example of an air conditioner - its components and their use in scenes



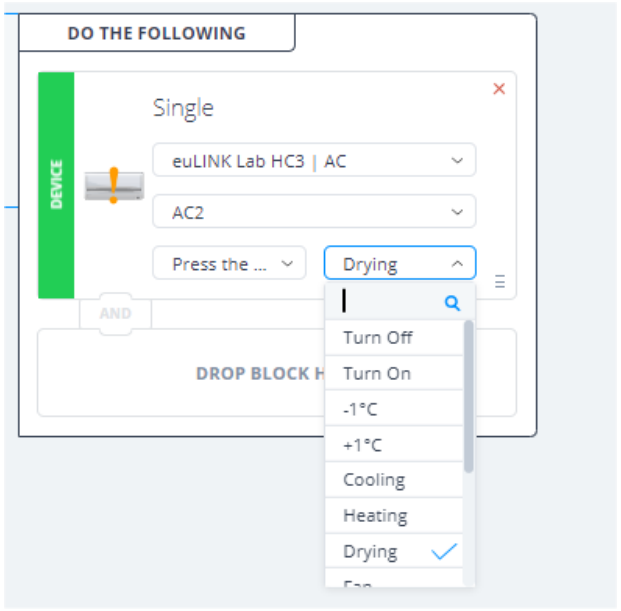
All of the above group actually represents just one appliance, the air conditioner, which in this example is called AC2. It is the first in the above list and is the superior device (*Parent*). The other devices represent various but very elementary functions of this one air conditioner. All these devices will be discussed in the table below, along with examples of how they are used in scenes.

In the left column of the table, there is a panel that is displayed when the cursor hovers over a 'tile' of the device and when the eye icon is pressed



AC2: Master (*Parent*) - the only mandatory object in this group, representing the status of the air conditioner, all its readings and giving the possibility of issuing any commands by simply pressing buttons. Each row of buttons is always placed directly below the row of readings whose value the buttons are supposed to change. For example, the -1°C/+1°C buttons change the temperature set point value, which is displayed on the line above the buttons.
In its normal state, the icon usually indicates the fan speed, but in the example it was deliberately forced to indicate that the filter needs to be replaced. This can be seen at a glance from the appearance of the icon, but it is also one of the text readouts (indicated by an arrow: "*Needs replacement!*"), placed by the template developer generally quite high up on the panel.

The pressing of each button can also be forced from the scene. For example, the action block shown opposite in the scene switches the air conditioner to dehumidification mode:

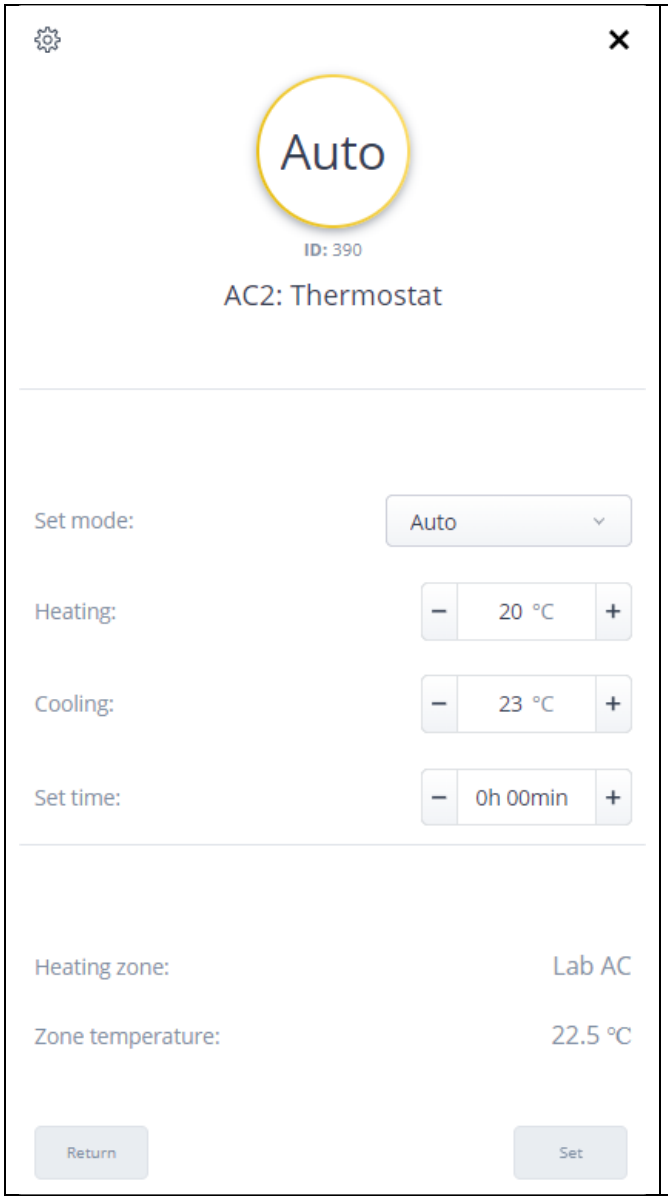


The LUA equivalent of this scene can be obtained by performing the conversion that HC3 offers. However, it will be more technically correct to use a direct reference to the function that is imported in the QuickApp code of this air conditioner:

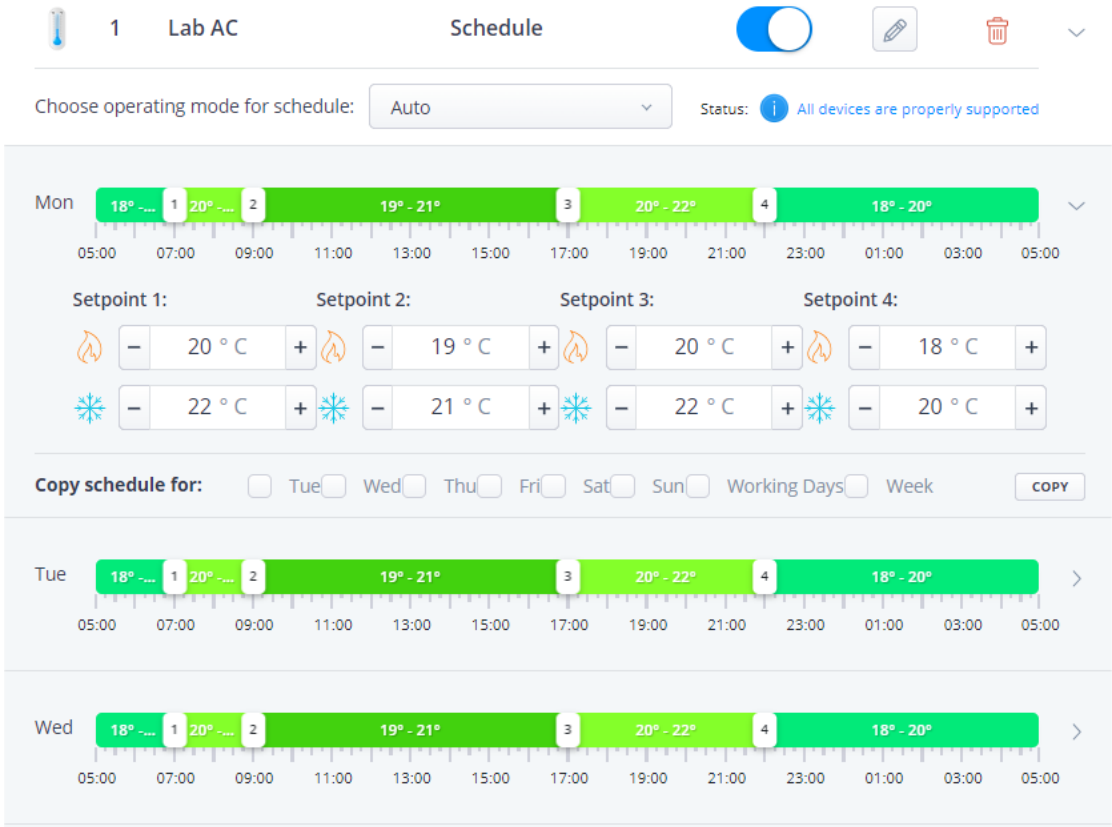
```
local AC2_ID=148
hub.call(AC2_ID, "eulinkCommand_setModeDry")
```

The corresponding function name for each command can be previewed by opening the QuickApp code of this air conditioner in the HC3 editor and looking for the function definition by pattern:

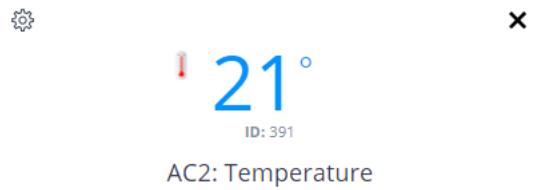
```
function QuickApp:eulinkCommand_setModeDry()
```



AC2: Thermostat - allows both manual control of the air conditioner's operating mode and target temperature setting, and allows the air conditioner to be designated as the executing device for a zonal cooling (and sometimes heating) schedule:



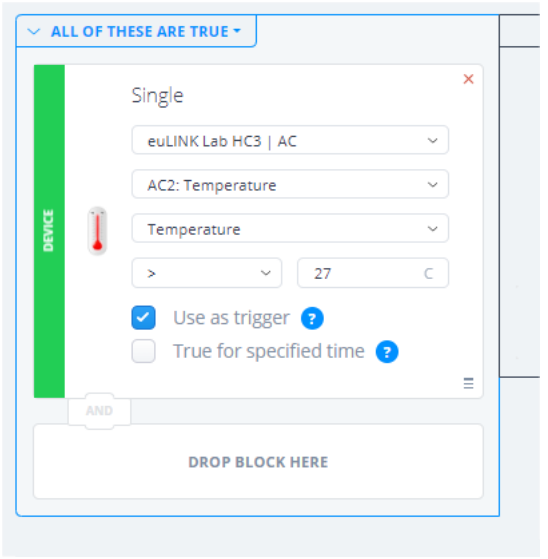
Some air conditioners have only one temperature setting (the same for cooling and heating), yet the thermostat software on the HC3 always sends both settings and in a sequence that is difficult to predict. So it is best to enter the same temperature in both fields to get the desired effect in the air conditioner. This is why we often create a template variant limited to the cooling function - even though the air conditioner also has a heating function. However, if the heating function is rarely used in a given climate, users like to choose this simpler template variant.

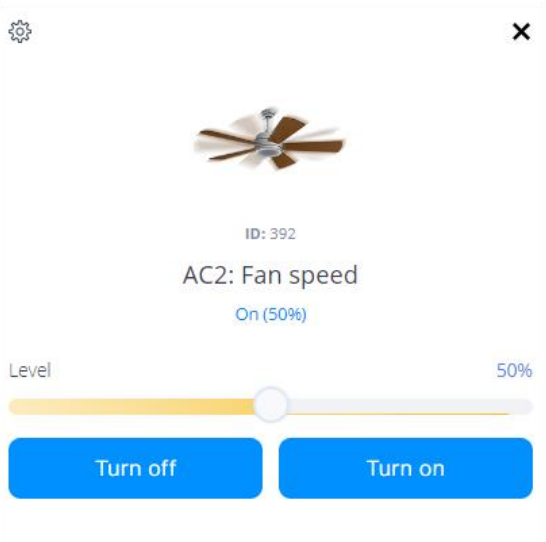


AC2: Temperature - the temperature reading in the room in which the air conditioner is located. Most air conditioners have an air temperature sensor - either this is located under the air conditioner casing as the air intake temperature sensor or, for example, built into the wall-mounted LCD controller in the room.
The block visible on the right will trigger the scene if the room temperature exceeds 27°C. The scene can, for example, send a notification or increase the air speed.

Reading this value in the LUA scene:

local ac2_temp = hub.getValue(391, "value")

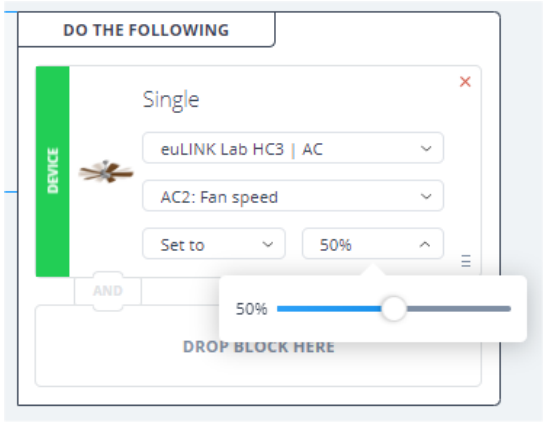


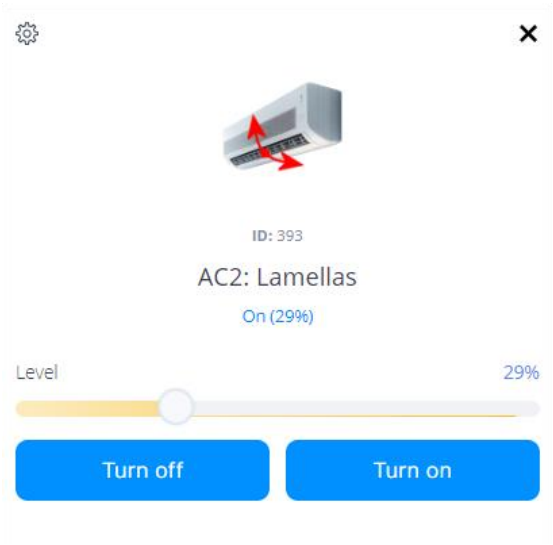


AC2: Fan speed - fan speed control, usually in several stages. Some air conditioners have 3 speeds (minimum, medium and maximum), others even 5 or 6. The euLINK gateway assumes that a value of 0% of the slider corresponds to the lowest possible fan speed, and a value of 100% corresponds to the maximum speed, and converts the speed set by the slider to the appropriate fan gear accordingly. The block visible on the right sets the average fan speed in the scene.

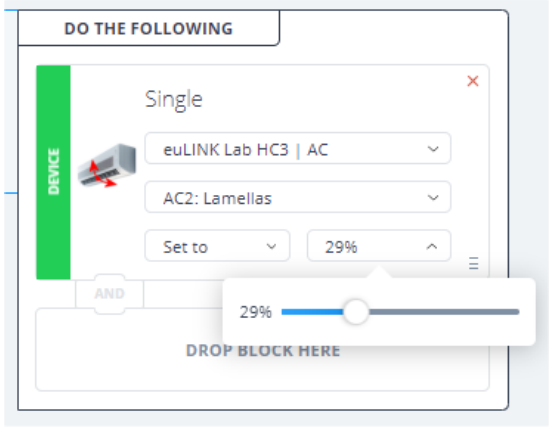
The speed reading in the LUA scene is of the form:
local ac2_fan_speed = hub.getValue(392, "value")

a command to set the average fan speed:
hub.call(392, 'setValue', 50)



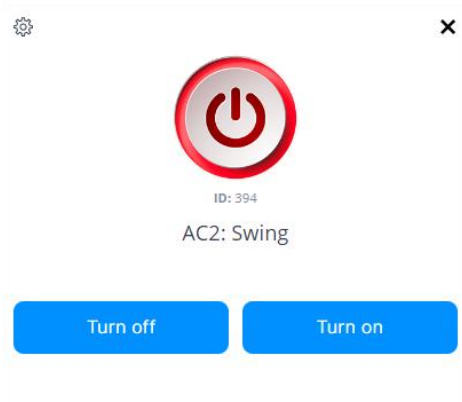


AC2: Lamellas - optional, as only some air conditioners have the possibility to adjust the angle of the slats, which direct the flow of cooled air more upwards or downwards. The euLINK gateway assumes that a value of 0% of the slider corresponds to directing the air stream downwards, while a value of 100% directs the air stream horizontally and converts the position of the slider into a slat angle setting command accordingly. The block visible next to it raises the slats to approximately 1/3 from the bottom.

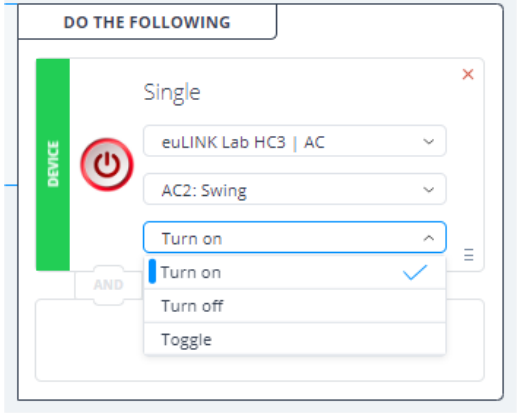


Reading the angle of the fins in a LUA scene:
local ac2_lam = hub.getValue(393, "value")

and the command to set the angle to 30% is of the form:
hub.call(393, 'setValue', 30)

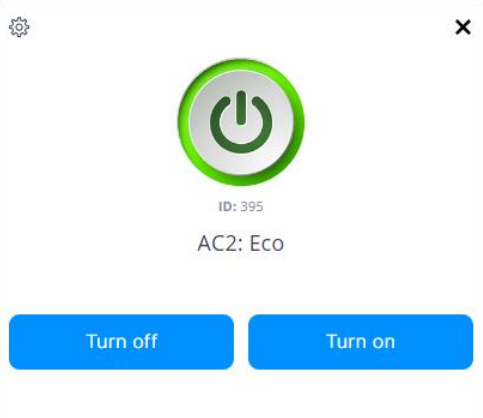


AC2: Swing - many air conditioners have a *Swing Function*, which allows cooled air to be more evenly distributed in the room. The block visible on the right activates this fanning.

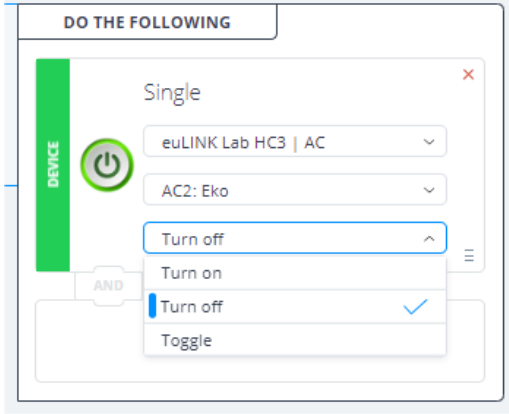


Reading the state of this function in the LUA scene (the result has the logical type *boolean (false|true)*:
local ac2_swing = hub.getValue(394, "value")

and the command to enable fanning takes the form of:
hub.call(394, 'turnOn')



AC2: Eco - Some air conditioners have an operating mode with reduced electricity consumption. The cooling of the room air may then take much slower, but it is more environmentally friendly. In certain circumstances you may need to automatically switch this mode off, for example when there is a large overproduction of energy from photovoltaics. The block shown opposite switches off the eco mode.



Reading the state of this mode in the LUA scene (the result has the logical type *boolean (false|true)*:

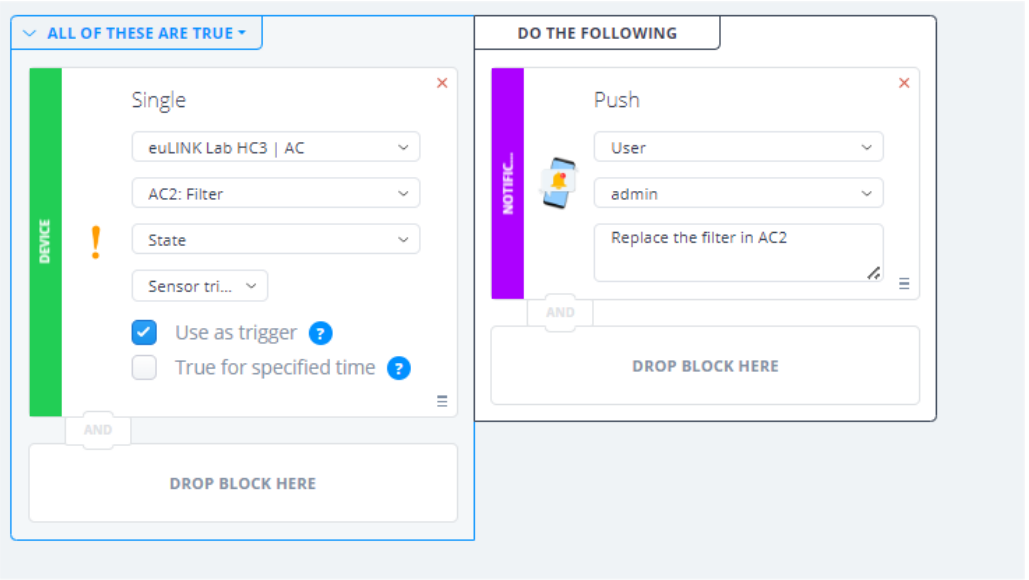
```
local ac2_eco = hub.getValue(395, "value")
```

and the command to deactivate eco-mode is of the form:

```
hub.call(395, 'turnOff')
```

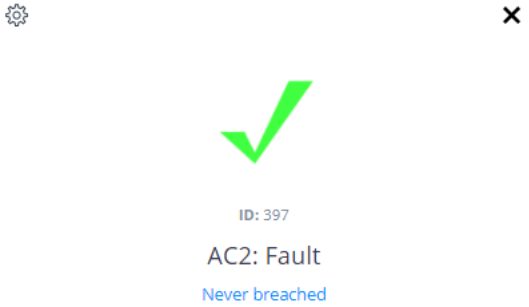
AC2: SG mode - a multi-level switch to switch the air conditioner into thermal (cooling) energy storage mode, as required by the *Smart Grid Ready* standard. A function so interesting and important that it is described in more detail in the next subsection, under the table describing the heat pump on page 76.

AC2: Filter - The air conditioner can inform you when the air filter needs to be changed, and this information can be used to send you a notification:



If the following reading of the filter status in the LUA scene returned a logical result of *true* (sensor breached), then the filter needs to be replaced:

```
local ac2_filter = hub.getValue(396, "value")
```

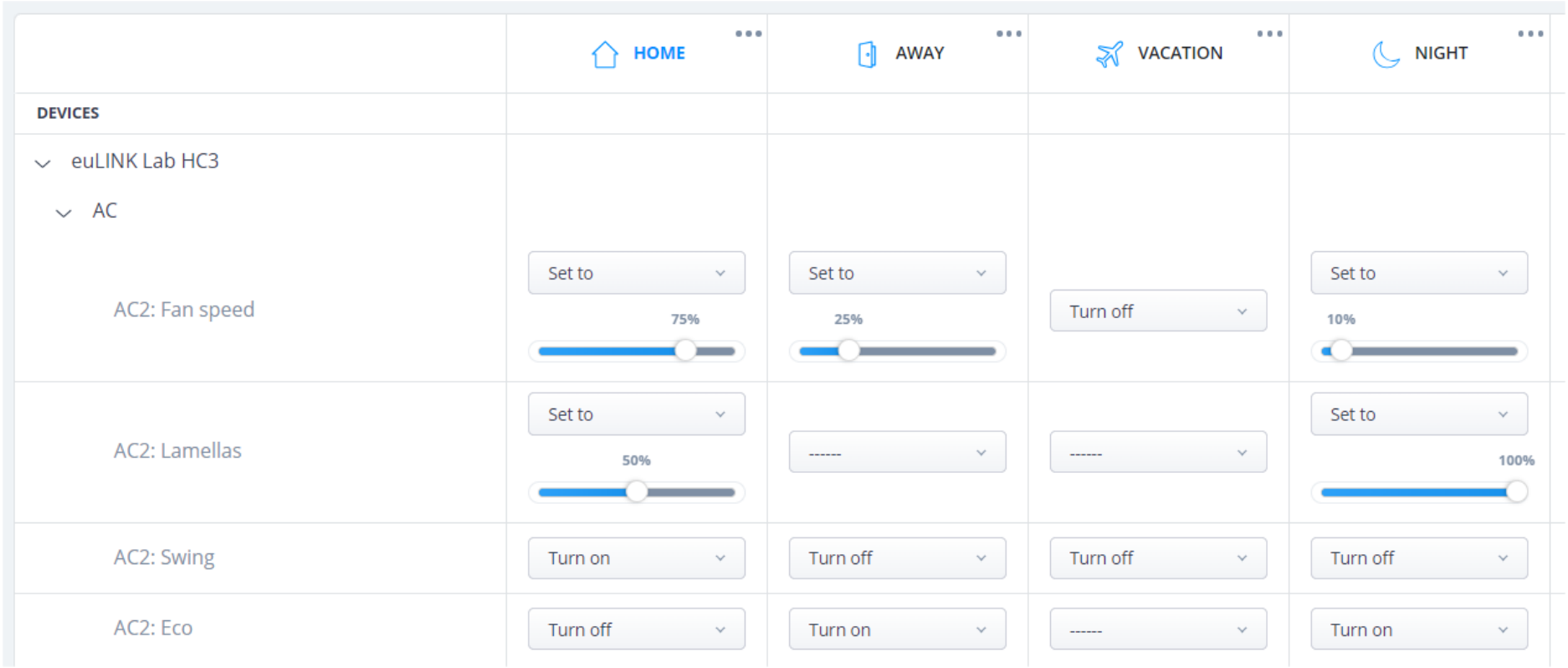


AC2: Fault - This is the same type of object as the above binary filter dirty status sensor, so block scenes and LUA are constructed identically for it (for **id=397**).
However, if the air conditioner has detected and signalled a fault, it is a good idea to read the fault code from the air conditioner's main panel, as this information can be useful when making a service appointment.
The fault code can also be read from a global variable in the LUA scene, using the ID of the parent device (*Parent*). For the example AC2 air conditioner, this identifier is **403**, so the corresponding global variable will be named:
eu_403_faultCode or similar. It can be read in the LUA scene as follows:

```
local ac2_faultCode = hub.getGlobalVariable("eu_403_faultCode ")
```

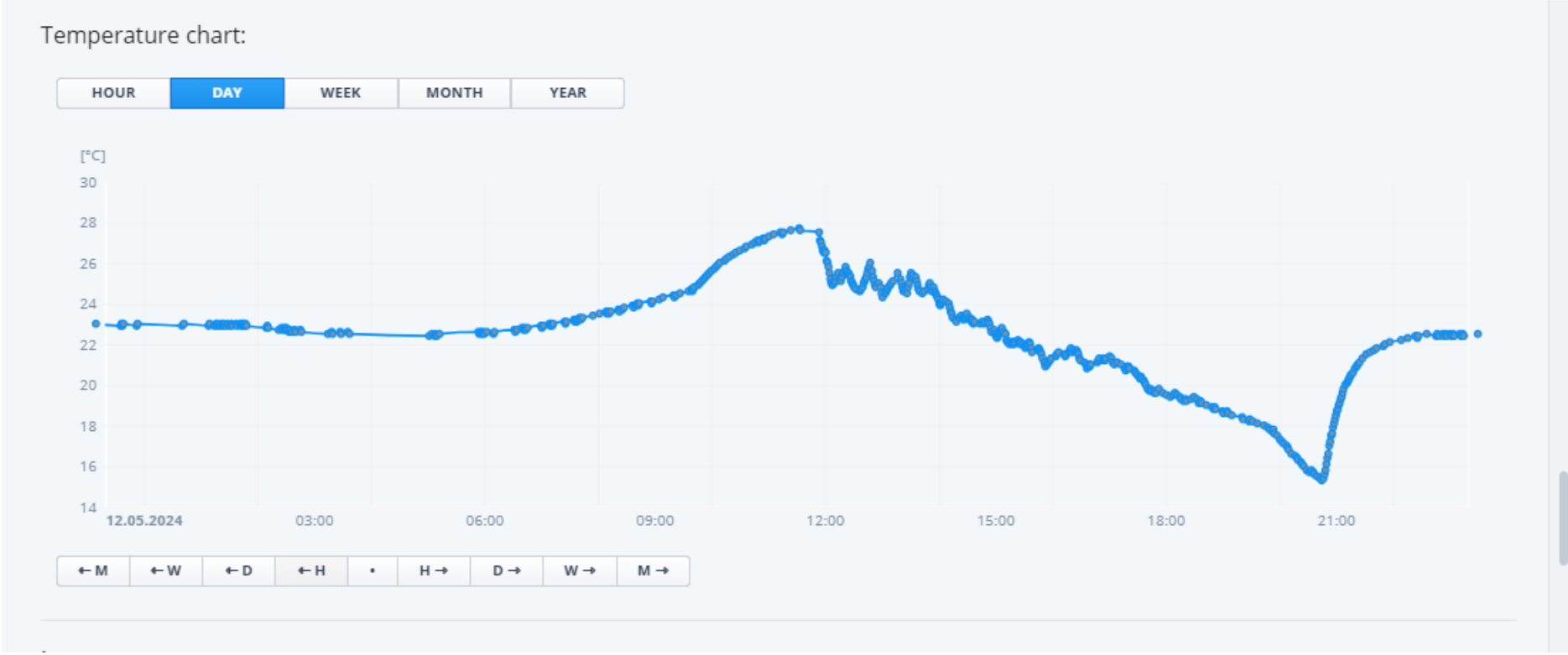
ii. Example air conditioner - use of its components in HC3 profiles

It is possible to use most of the described elementary devices in profiles, i.e. changing the profile can automatically give commands to the air conditioner. For example, it is possible to switch off fanning and reduce the intensity of the air supply at night or when the occupants are away, or to switch off unnecessary functions altogether for an extended holiday trip:

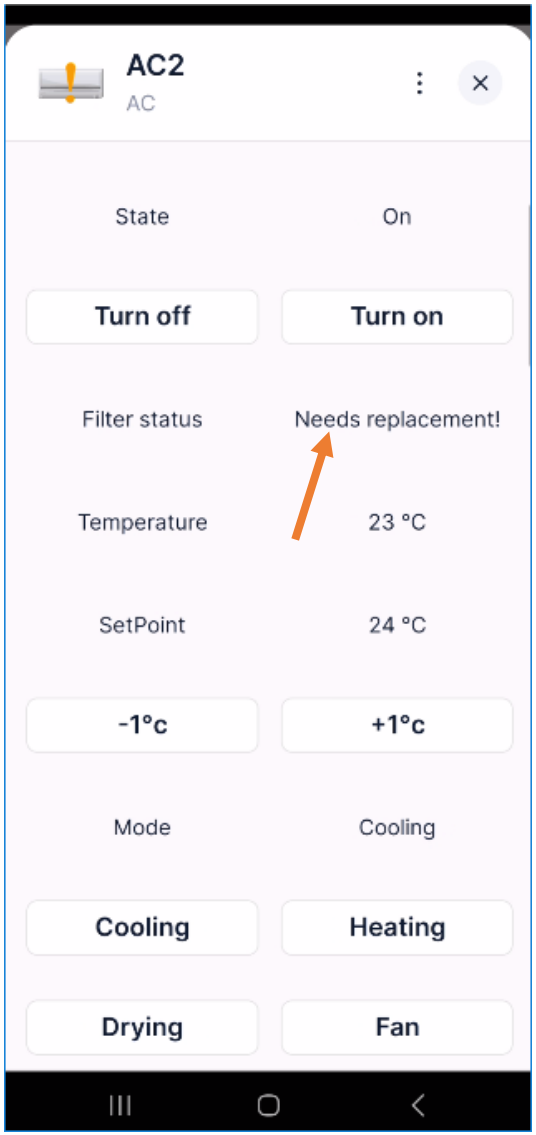


iii. Example air conditioner - graphs from built-in temperature sensor

The temperature sensor, which is built into the air conditioner, can provide useful information and can even be used to plot the room temperature. In the graph below (taken from the sensor of the example air conditioner with id=391), you can clearly see the increase in temperature at midday, when the sun was peeking into the room, as well as the evening drop, caused by ventilation. Just bear in mind that the temperature measurement from the built-in sensor is only reliable when the air conditioner is on and its fan is running (even if only at the lowest speed) to draw air from the room.



iv. Example air conditioner - control from the Yubii smartphone app



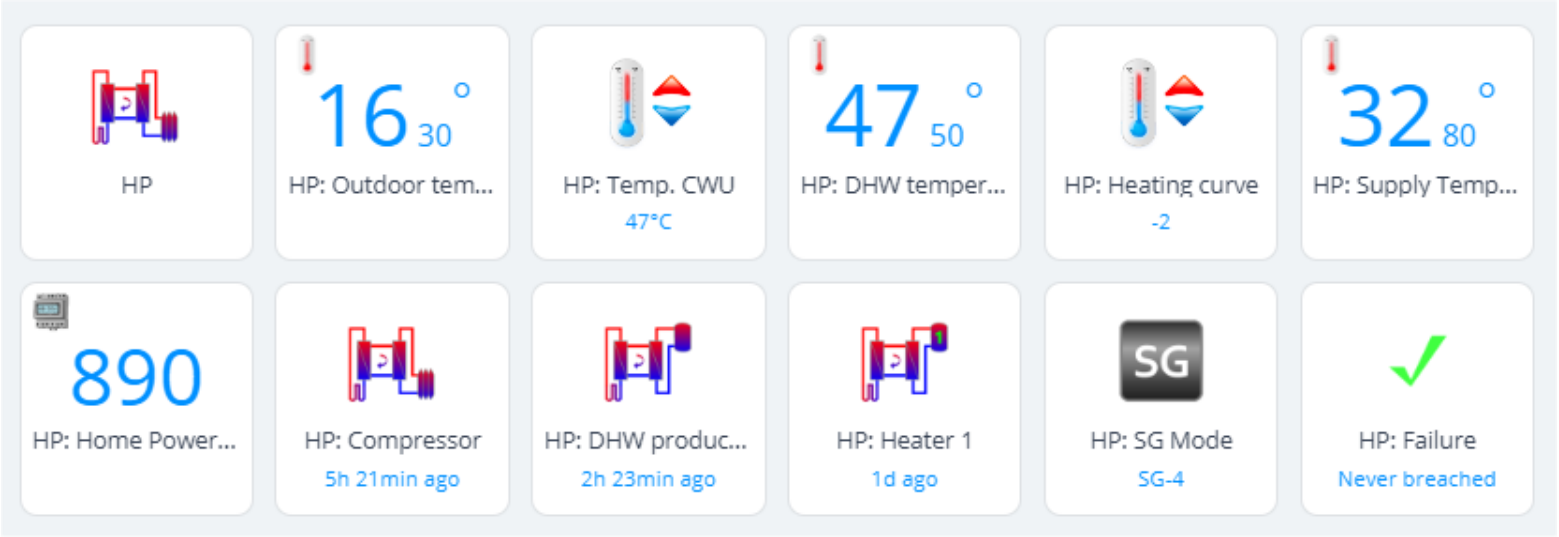
The appearance of the air conditioner panel in the Yubii app very much resembles that of the parent appliance (*Parent*). The more complex units usually do not fit all their readouts and buttons on the smartphone screen and require a scrolling view. But the most important information should be at the very top of the screen, including the status of the air conditioner, the temperature (current and target) and the status of the filter (marked with an arrow in our example). In the latest versions of the Yubia app, all status icons of the unit are also already shown correctly.

Although the euLINK gateway can bring many different devices to the Home Center controller, their appearance in the controller and in the Yubii app will always be similar and standardised to some extent. As a result, users quickly get used to it and become proficient in managing all their home equipment.

Many installers also appreciate this unification, because not only is the appearance of the panel similar, but the creation of scenes or the management of profiles looks very similar - even for devices as different and complex as air conditioners, heat pumps, car chargers or photovoltaics.

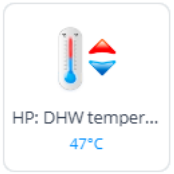
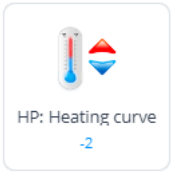
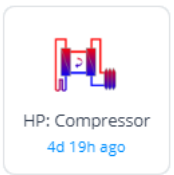
v. Example heat pump and its components

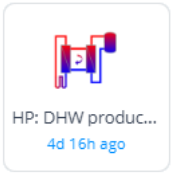
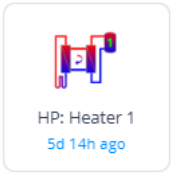
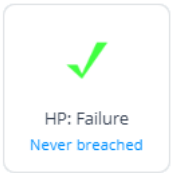
The following group represents an example of a heat pump and its components:



These components will be discussed in the table below, but no longer in as much detail as for the air conditioner, as the component types are the same and so are the scene controls.

	**HP:** Master (*Parent*) - representing the main control panel of the heat pump (in the preview it looks like a "remote control"). The status icon is important, as the heat pump can have quite a few of these states - so the icon can signal, among other things: idle, circulation pump operation, compressor operation, underfloor heating mode, DHW tank heating and, of course, a fault or no connection to the pump. In block scenes, commands can be given by 'virtual' button presses. The example device name comes from the term *Heat Pump*.
	**HP: Outdoor temperature** - each heat pump has an outdoor air temperature sensor connected and automatically selects its operating mode based on this: Summer / Winter. The water in the DHW tank is heated throughout the year and, in addition, hot water is pumped into the underfloor heating system during the cold months.

	**HP: DHW temp** - the slider controller allows you to set the desired average DHW tank water temperature. The assumed hysteresis width is 5°C, so setting the target temperature to 47.5°C means that the euLINK gateway will program the lower threshold at 45°C and the upper threshold at 50°C. If the heat pump measures that the water temperature in the DHW tank has fallen below 45°C, it will start to heat it up until it reaches 50°C. Of course, the euLINK gateway ensures that the set temperature does not exceed the manufacturer's recommendations for the heat pump type.
	**HP: DHW temp** - a sensor that measures the actual temperature of the water in the DHW tank. In this example, the reading should not be outside the range of 45-50°C. A graph of these readings is possible, but not very interesting as the temperature oscillates in a small range. A more interesting graph will be possible if the pump acts as a thermal energy store, because then the water temperature in the DHW tank can rise significantly - even above 60°C.
	**HP: Heating Curve** - a slider control that allows the so-called *Heating Curve to be shifted* within a small range. The creator of the template for the euLINK gateway should assume that the middle position of the slider (50%) means no shift of the curve, while the extreme positions are a shift of the curve by -3°C or +3°C and recalculate the commands sent to the pump accordingly. In this way, users gain the ability to regulate the temperature in the house by deviating 1, 2 or 3 degrees down or up from the optimum central position. Usually these 3 degrees are sufficient to adjust the comfortable temperature in the house.
	**HP: Supply temp** - sensor for the temperature of the water, pumped into the underfloor heating system. In the summer months, this temperature is practically equal to room temperature, while in the autumn and winter months it can exceed 30°C.
	**HP: Home Power Consumption** - the heat pump may have the option to connect 3 current transformers (*CTs - Current Transformers*), mounted on each phase conductor of the building's main power line (HV). This allows the heat pump to measure the current power consumption of the house and calculate whether it is allowed to run the immersion heaters in the DHW tank. These heaters can have such a high power consumption that switching them on without taking into account the current power consumption of the house (which can also sometimes be high) can risk tripping the main overcurrent fuse of the whole house.
	**HP: Compressor** - a binary sensor that indicates the operating status of the heat pump compressor. The compressor is a significant load on the system, as its power consumption can reach 2-3kW.

	**HP: DHW production** – a binary sensor, indicating hot water production in the DHW tank.
	**HP: Heater 1** - a binary sensor, indicating the operating status of the first heater in the DHW tank. There may be more than one heater, and each may have a power consumption calculated in kilowatts, so their switching on has a significant impact on the energy balance of the home. Under normal heat pump operating conditions, the heater is switched on occasionally - e.g. a few times a year, when the building's hot water demand increases rapidly and the heat pump's inertia does not allow it to heat up quickly.
	**HP: SG mode** - a multi-level switch that causes the heat pump to switch to heat storage mode, as required by the *Smart Grid Ready* standard. A function so interesting and important that it is described in more detail just below this table.
	**HP: Fault** - a binary sensor, customarily indicating that the heat pump has reported some internal fault. It can be used to issue a notification. The fault code can be read from a global variable containing the ID number of the master (*Parent*) of the heat pump.

As well as heating the home cleanly, quietly, safely and ecologically, a heat pump can also be a simple and effective **store of thermal energy**. After all, you can force the heat pump to heat the water in the DHW tank to a much higher temperature than normal - which, after all, is always some compromise between needs and economy. The euLINK gateway can reprogram the target temperature in the DHW tank, forcing the heat pump to dramatically increase its electricity consumption, sometimes even forcing the heat pump to switch on the immersion heater in the water tank. During the heating season, the temperature of the water pumped by the heat pump into the underfloor heating system can also be raised a little. If there is a photovoltaic installation in the building, the euLINK gateway can detect that a large overproduction of energy from the sun is being fed back into the grid and can automatically switch the heat pump into an increased electricity demand mode. In this way, even in the absence of householders, the autoconsumption of energy can be improved, which is always more cost-effective and technically correct than feeding energy back into the grid.

Many heat pump manufacturers are equipping their latest products with the *Smart Grid Ready* function[5] (*SG-Ready*) for, among other things, tariff control. Using this function, the electricity supplier can remotely control the operating mode of the heat pump at the energy consumer. Four basic operating modes have been defined for the *SG-Ready function*:

1. blocking action (hereinafter: **SG-1**)
2. normal operation (**SG-2)**
3. high energy consumption mode (**SG-3**)
4. maximum energy consumption mode (hereinafter: **SG-4**)

In the latter mode, the heat pump heats the DHW tank water with all its available capacity up to the maximum temperature allowed by the manufacturer. With some modern heat pumps, the euLINK gateway does not even need to reprogram the target DHW tank water temperature, as sending a mode change command to **SG-4** is sufficient to increase the heat pump's power requirement. However, if the heat pump is not factory-equipped to support *SG-Ready modes*, an appropriately prepared euLINK template can provide the function of setting **SG** modes as a 'multi-level switch' in HC3. This function is then implemented by the euLINK gateway in the form of a command, increasing the DHW tank water temperature setting to the maximum allowed level, e.g. 60°C. When selecting the value for this maximum temperature, it is a good idea to check what temperature the manufacturer recommends for the periodic water heating mode to eliminate legionella bacteria from the water.

Whichever method is chosen to convert a heat pump into a thermal energy store, it is always advisable to consult the heat pump supplier who, after all, has the knowledge and vast operational experience.

The following page describes an example of an advanced scene, controlling the SG modes of domestic appliances to improve the self-consumption of solar energy.

c) Once integration into the Home Centre is complete

As a final step, the ability to control all elementary functions of the MODBUS device from the Home Center controller website and from the Yubii smartphone app should be tested. If the tests were successful, the basic integration can be considered successful ☺

[5] The term *'Smart Grid Ready'* and the associated label was developed by the association Bundesverband Wärmepumpe (BWP) e.V.

14. Examples of scenes using integrated HVAC/PV/EV equipment

One of the most necessary functions of the Smart Home is the management of the energy flow in the home. Energy is consumed by household appliances, but can also be produced at home by RES (e.g. photovoltaics, wind turbines, etc.) and fed back into the grid or consumed on site (self-consumption). Optimising the flow of this energy from sources to consumers is not an easy task. A PV installation, even in a small house, can provide energy measured in kilowatts (5-10kW installations are the most popular), while the energy requirements of ordinary household appliances tend to be in the tens or hundreds of watts. The few appliances with higher power consumption (e.g. heat pump, air conditioners, electric car charger) reach their maximum demand when householders return home after work, i.e. in the afternoons and evenings - when photovoltaic energy production "wanes somewhat". It is possible to invest in the purchase of electricity storage, or to use the SG-4 function in the heat pump, as described in chapter 13.v on page no. 74. However, none of these devices with higher energy consumption know themselves when they should go into increased power consumption mode. Some energy storage units and heat pumps have factory-built support for photovoltaic cooperation, but this is subject to a number of restrictions, e.g. they must be specific appliances from a particular manufacturer. However, it is difficult to find air conditioners or washing machines with this function on the market. And none of these appliances take into account the user's preferences in terms of a priority list. After all, if there are several energy consumers and the current surplus of energy from photovoltaics (overproduction) is insufficient, the question of which appliances should be switched off and in which order they should be switched off and on has to be answered. Of course, in the algorithm under discussion, the term 'Switch on' or 'Switch off' does not include a straightforward disconnection of its power supply, but only the sending of a subtle request in the MODBUS protocol to smoothly change the operating mode, as far as the autonomous capabilities of the integrated HVAC/PV/EV device are concerned.

So let us consider the following list of popular devices, ranked in exemplary order of attachment according to user preference:

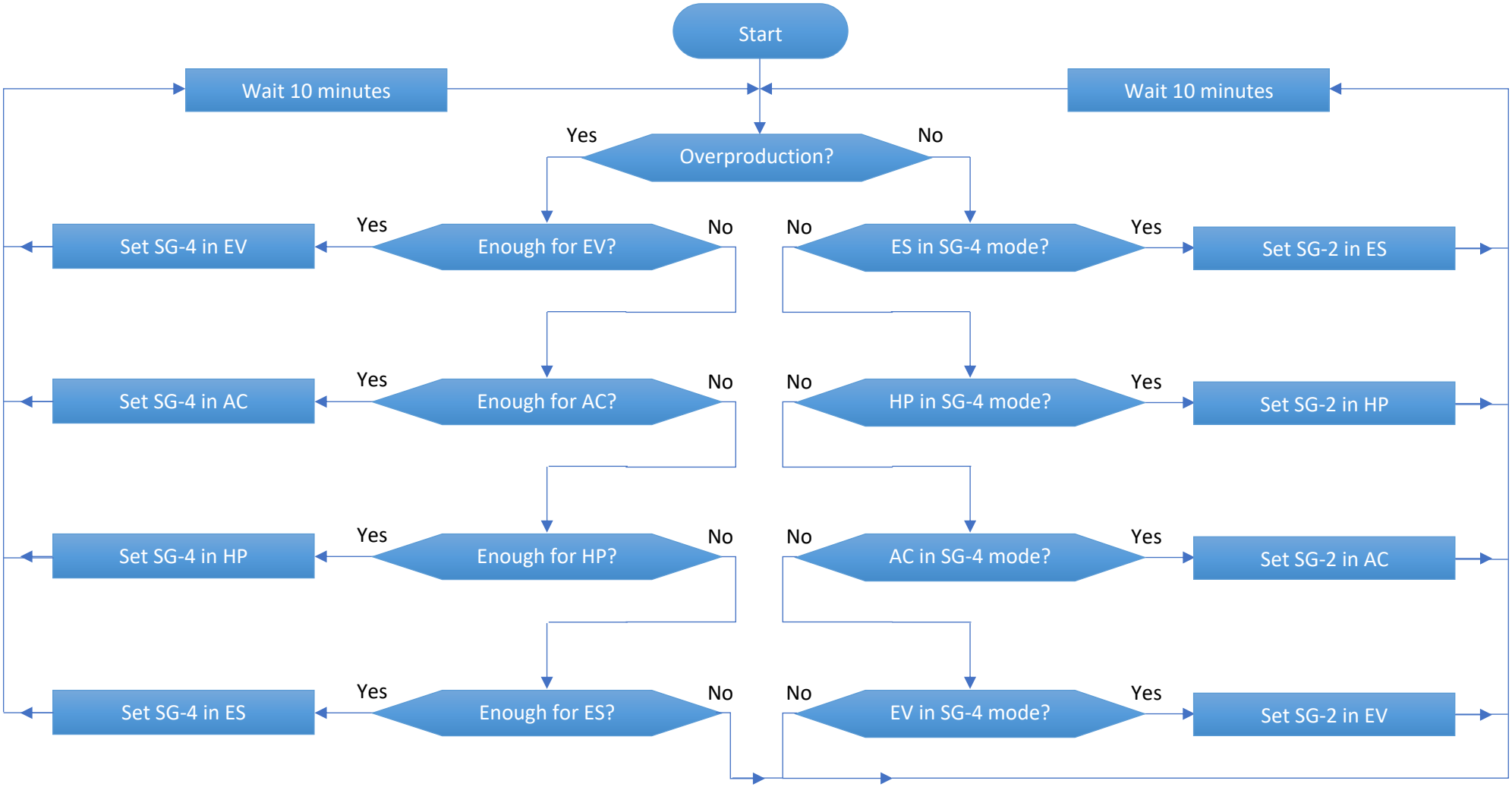
- EV - charger (or several chargers) of an electric car
- AC - air conditioning (one or more air conditioners)
- HP - heat pump
- ES - electrical energy storage (battery bank)

Of course, another user may prefer a different order, but this can easily be changed.

Every photovoltaic inverter knows the amount of energy currently produced, and most also work with a two-way energy meter mounted at the interface between the home installation and the grid, so the amount of overproduction is known and should be included in the euLINK template.

So you can build a scene in HC3 that periodically reads the amount of overproduction from the PV or the meter and compares it with the power demand of the following devices from the list above: EV, AC, HP and ES. If there is sufficient overproduction and the unit was in the usual SG-2 mode, the unit is switched to the enhanced power mode (SG-4). And if there is no overproduction, SG-2 mode is successively restored.

The operation of the '*Smart-Grid Manager*' scene can be described graphically using the following algorithm:

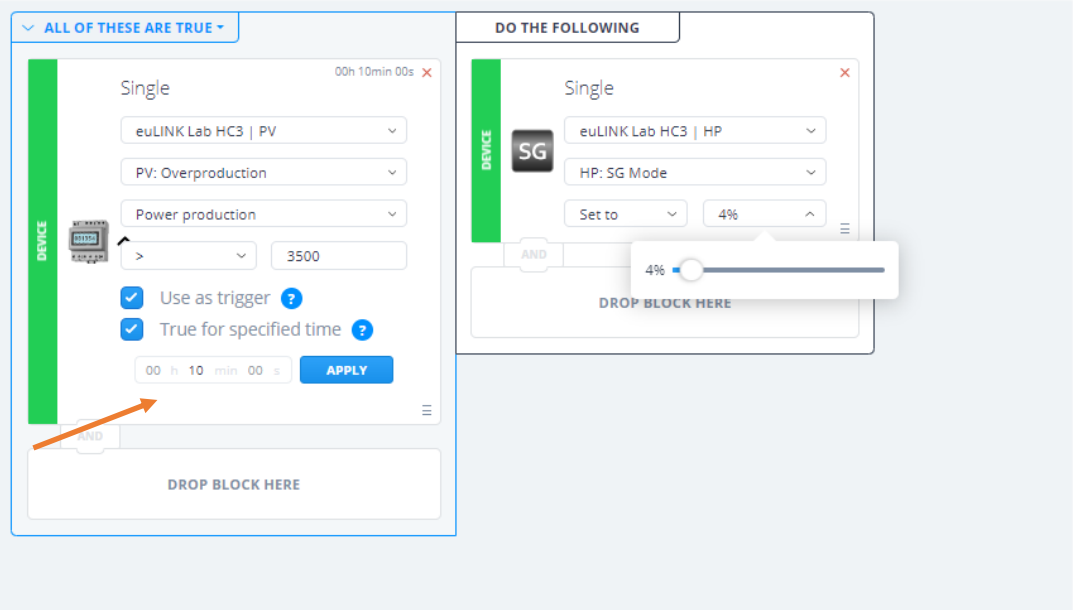


EV - car charger
AC - air conditioning
HP - heat pump
ES - energy storage

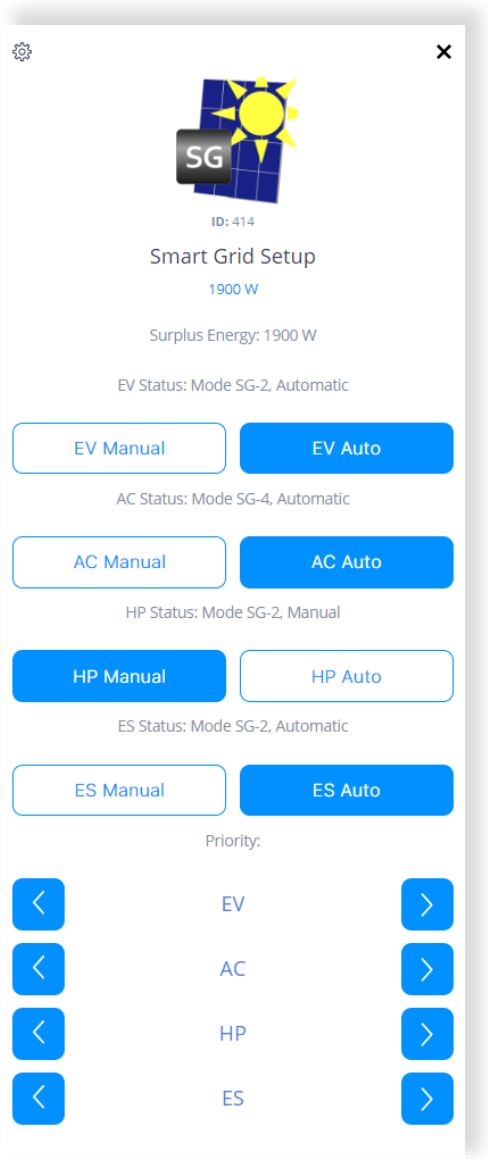
When the device is set to SG-4 mode, the algorithm returns to the start, waits 10 minutes, re-reads the PV overproduction value and the cycle repeats. There is no point in waiting any shorter, as it can take several minutes for some EV/AC/HP/ES devices to start up and only then will you see their impact on the energy measurement result. Also, short-lived cloudiness does not cause any nervous reactions of the integrated devices. Furthermore, the *Smart Grid Ready* standard requires that changes occur no more frequently than every 10 minutes. If a device is already in SG-4 mode, it is skipped and the scene moves on. It is to be expected that the overproduction value will drop by the amount of additional consumption of a device in SG-4 mode. If the reading indicates no overproduction, the devices are individually returned to SG-2 mode in reverse order of the defined priority list. As a result of the operation of such a scene, it is certain that on a sunny day, the most important devices for users will be running "for free" for many hours ☺

Knowing the increase in SG-4 power demand of a given EV/AC/HP/ES device (expressed in watts) can be compared with the instantaneous power reading indicated by the two-way meter, mounted at the grid interface. The scene maker just has to remember to treat the amount of overproduction as a negative number, because a positive reading in turn indicates energy intake from the power grid rather than local production. Of course, it is worth adding some reasonable margin, e.g. increase the appliance demand by 10% before comparing the output, to make the system immune to minor fluctuations in overproduction from PV and momentary fluctuations in the energy consumption of individual appliances. It is worth observing the graph of overproduction in HC3 for a few sunny days, as it will be possible to deduce from this graph, for example, the need to increase the margin slightly.

Of course, not every installation will have all the devices mentioned in the example, often there will be fewer. If there were only one such appliance - for example, a heat pump with thermal energy storage function (**SG-4** mode) - then such a scene could even be created from building blocks. If we estimate the increase in power requirement of a heat pump in SG-4 mode to be about 3200W, this value can be increased by about 10% and entered explicitly in the scene configuration. It is only necessary to add the condition of meeting the requirement of sufficient overproduction for at least 10 minutes to avoid switching the pump mode with short-term increases in overproduction. This condition is added as can be seen on the right =>



When creating a complementary scene (which restores SG-2 mode heat pumps in the absence of overproduction), meeting the constancy condition is no longer necessary and SG-2 mode can be restored immediately.



However, if there are more devices, it would be difficult to ensure the correct sequence from within the block scenes. Such a task should be entrusted to a LUA scene. We have therefore prepared a file in ZIP format containing a set of components of such a LUA scene. However, please note that this is an example scene, which should be adapted to your own system configuration. So please treat the example provided as an inspiration rather than a ready-made "recipe" to be used indiscriminately. The file can be downloaded from here:

https://www.eutonomy.com/download/eulink/hc3/smart-grid-mngr.zip

File content:	File name:
Instruction (last 2 letters of file name indicate language)	smart-grid-mngr-doc-pl.pdf
Definition of a QuickApp object, controlling the operation of the "*Smart Grid Manager*" scene, to be uploaded when creating a device from a file	smart-grid-mngr-setup.fqa
Scene triggers to paste in the DECLARATIONS window of the LUA editor	smart-grid-mngr-scene-trigger.lua.txt
LUA code of the scene to paste in the ACTIONS window of the LUA editor	smart-grid-mngr-scene-actions.lua.txt
Optional icon that can be assigned to the QuickApp object	smart-grid-mngr.png

The included **instructions** describe the step-by-step process of installing a scene in HC3.

Householders should be able to configure the operation of this scene, e.g. to exclude any of the devices from scene operation (Manual Mode), to include a device in the scene (Automatic Mode) and to change the order of operation of EV/AC/HP/ES devices (priority). A small QuickApp panel (shown on the left), included as a **smart-grid-mngr-setup.fqa** file, is used for this purpose. This file should be used to create a new QuickApp device, and can also be provided with a shared icon. The variables of the QuickApp object store the values of the power demand of the individual devices in **SG-4** mode. It is necessary to adjust these values, adjusting the example values to the actual demand of the involved EV/AC/HP/ES devices in **SG-4** mode. Further variables contain the identifiers of the EV/AC/HP/ES devices, which are individually assigned by HC3 when these devices are imported:

Variable name	Device ID	Variable name	Power consumption in **SG-4** mode [W].
EV_ Device_ID	418	EV_Power_Demand	5000
AC_Device_ID	419, 422, 425	AC_Power_Demand	in total: 1300
HP_Device_ID	420	HP_Power_Demand	3500
ES_Device_ID	0	ES_Power_Demand	4000

3 air conditioners

energy storage is not installed there

15. Energy Manager and *Smart Grid Ready* standard

It is worth checking the legal status before starting design work, as many EU governments are stimulating rational energy management through significant subsidies for the construction of energy flow and balance management systems. Particular emphasis is placed on maximising the self-consumption of energy produced by RES, which significantly reduces the load on the power grid across the country. Experts from even the most developed countries in the European Union acknowledge that today's electricity grids are not well equipped to absorb the amount of energy prosumers are already producing today - and the rate of growth is increasing! This is why subsidies are significant, sometimes even sufficient to build an entire Smart Home system - if, of course, it is a reasonably priced implementation - such as the one we have just discussed.

This may change the approach of Smart Home designers from the previous 'medieval' approach, expressed by the maxim:

- Let's install Smart Home devices that control lighting, roller shutters, Audio/Video equipment and other gadgets, and if there is still something left in the budget, let's integrate heating and photovoltaic control "by the way"

For a more modern approach:

- Let's use the grant to build a Smart Home PV energy spread management system, and if there's still something left in the budget (and there probably will be), let's 'incidentally' add control of all lighting and blinds.

This innovative approach, much appreciated by investors, is made possible by using the euLINK gateway in cooperation with the FIBARO Home Center system.

Furthermore, the euLINK gateway can become a so-called 'Energy Manager' in line with the *Smart Grid Ready* standard mentioned above, i.e. it can adapt the operating modes of all household HVAC/PV/EV equipment to commands sent out by grid operators - even if the individual HVAC/PV/EV equipment does not have such functions built in at the factory. Even if a heat pump is state-of-the-art enough to support *SG-Ready mode, an* electric car charger or air conditioners may not have this mode. And yet, the total energy consumed by these devices can be really significant. These devices should therefore be taken into account by an energy flow optimisation mechanism - and thanks to the euLINK gateway, this is already possible.

The scene discussed in the previous chapter reacts to a reading of the local PV overproduction value, increasing the power demand of the integrated equipment. Work is also already underway to equip the euLINK gateway with a function that would react to the sending of **SG-3** or **SG-4** commands by the power grid operator, affecting the power consumption of all equipment in the building.

The ability of the euLINK gateway to influence the power consumption of HVAC/PV/EV equipment can be particularly beneficial in those electricity networks where so-called 'Hourly Energy Billing' is already in force. Simple scenes on the HC3 can help optimise energy consumption from the grid at different times of the day, which can mean noticeable savings in electricity bills ☺

16. If something is not working...

The most useful practical tool for troubleshooting the euLINK gateway is the manual we have developed:

1. Problem diagnosis

This manual is the result of many years of observing the operation of euLINK gateways in various environments and on the basis of 4 decades of accumulated experience in the Smart Home and IT industry.

It is also worth using other information sources published on our servers, including:

2. MODBUS connection diagrams
3. euLINK Datasheet
4. euLINK User Manual
5. AI-supported chatbot retrieving information from the product knowledge base
6. CE Declaration of Conformity

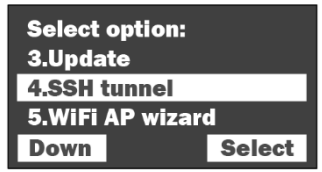
If you encounter problems that are not explained in this guide or in the description of the diagnostics, we encourage you to post your questions on our forum.eutonomy.com. There you can count on the help of a growing number of enthusiasts of our solution.

Alternatively, you can always email our technical department at support@eutonomy.com. The easiest way to do this is to use the report form that is built into every euLINK gateway. To do this, simply go to *Menu => Help => Report a problem*. The *Help* section also presents the above list of links to useful publications.

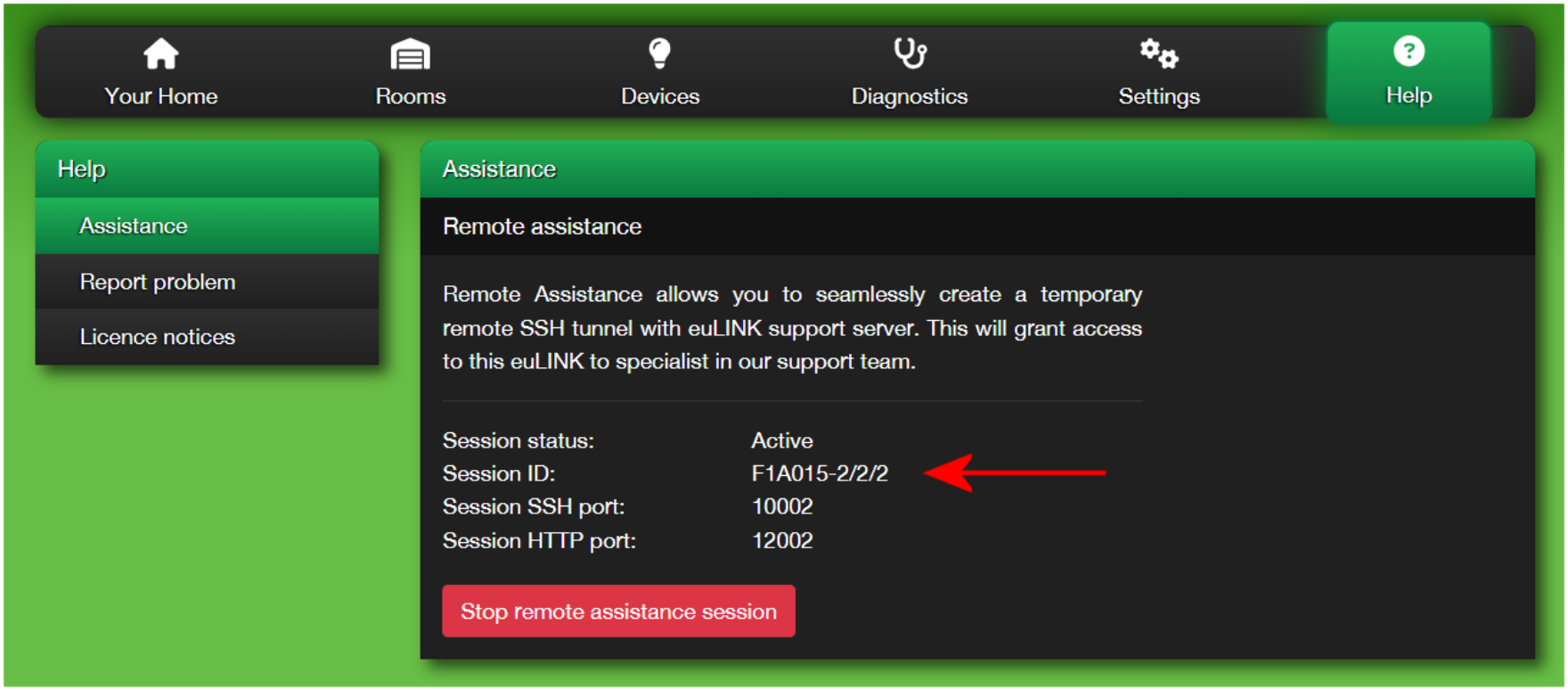
If the euLINK gateway is not working properly or its installation presents unforeseen difficulties, assistance can be sought from the manufacturer's specialists. The euLINK gateway is equipped with the ability to establish an encrypted connection to the manufacturer's server, so its engineers can carry out the diagnostic procedure at the lowest, hardware level and suggest an effective solution to the problem. The connection is secure as it does not require any TCP ports to be opened on the FireWall. The euLINK gateway does not accept external connections; instead, it simply establishes a connection to the server as a client, making the 'SSH tunnel' a secure, encrypted connection out of the local LAN. An important feature of this approach is that no one - not even the manufacturer's representative - can access the euLINK gateway without the knowledge and consent of the person who owns it.

A remote support session can be started in 2 ways:

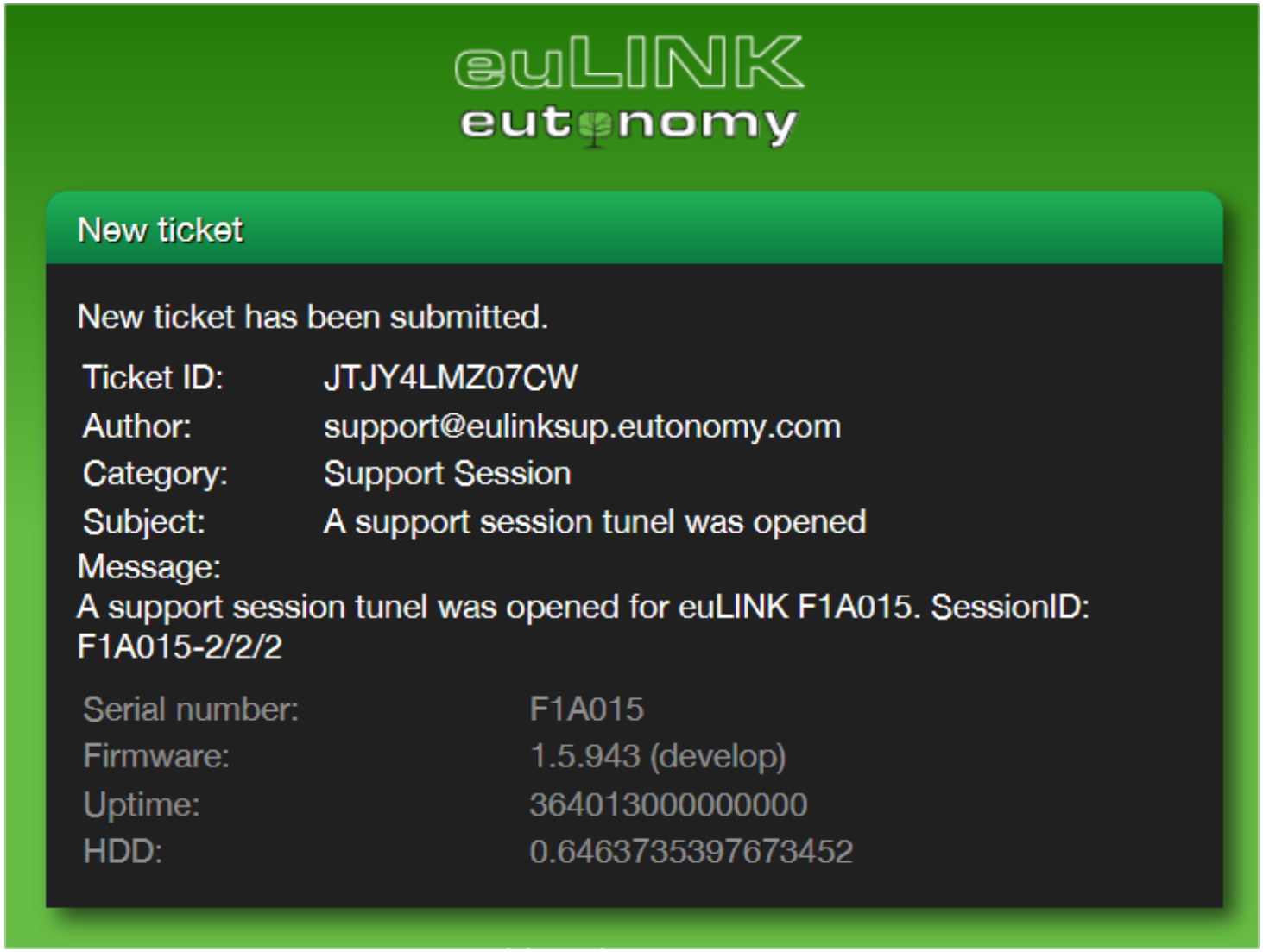
- From the OLED display on the euLINK panel, scroll down with the left button to the option "*4.SSH tunnel*" and accept it with the right button or
- Connecting to the euLINK gateway via a browser and navigating to *Menu => Help => Remote support => Start remote support session*.



To agree on a further procedure, please contact the manufacturer's service by e-mail at support@eutonomy.com and provide the SSH session ID, as read from the OLED display or from the *Menu => Help => Remote assistance* of the euLINK gateway:



The same information will be sent by the euLINK gateway to the email address provided during its installation. The message will have a similar design to the one below:



Once the work is completed, the remote support session can be stopped so as not to maintain unnecessary connections. Restarting the euLINK gateway will also disconnect the session, so if remote support is to continue, further approval from the euLINK gateway keeper will be required.

17. Summary

It is worth noting that, thanks to the euLINK gateway, the integration of a MODBUS device into the FIBARO system does not require any knowledge of LUA programming or the QuickApp technique for building complex objects. All necessary objects, icons and variables are automatically created by the euLINK gateway and then imported into the Home Center controller thanks to the FIBARO REST API mechanism. This should make the work of the MODBUS device integrator very easy.

The scale of difficulty of integrating MODBUS devices and diagnostic methods does not differ significantly from what installers of other advanced home automation systems (e.g. KNX, BACnet) have to deal with. For a skilled installer, the installation of MODBUS RTU technology connections does not present any difficulties - after all, it is just "two wires" ;-) More attention needs to be paid to configuration and careful documentation of the project.

Of course, integration is a little more difficult than installing individual sensors or lighting dimmers. But after all, there are more than just lamps or blinds in modern homes, there are also serious and expensive HVAC/PV/EV infrastructure devices, and householders have a keen interest in their integration and consistent control.

And with this, all major home appliances can finally work together as one cohesive organism - a true Smart Home ☺

We wish you good luck!

Maciej Skrzypczyński

CTO @ Eutonomy