SpaceLogic KNX

LSS100100 Wiser for KNX
LSS100200 spaceLYnk

User Guide

Firmware R 3.0.0

Release date 10/2024

www.se.com

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this document are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owner.

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety information

Important information

Read these instructions carefully and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this manual or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of either symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that accompany this symbol to avoid possible injury or death.

⚠⚠DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

Failure to follow these instructions will result in death or serious injury.

⚠WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

For Your Safety

⚠⚠DANGER

RISK OF FATAL INJURY FROM ELECTRIC SHOCK, EXPLOSION OR ARC.

Safe electrical installation must be carried out by qualified professionals.

Qualified professionals must demonstrate an in-depth knowledge of:

- Connecting to installation networks
- Connecting multiple electrical appliances
- Installation of electric cables
- Connection and setup of KNX networks
- Commissioning KNX installations
- Safety standards, local connection rules and regulations

Failure to follow these instructions will result in death or serious injury.

The devices and the associated ETS application must not be used to control safety-related applications.

Qualified Personnel

This document is aimed at personnel who are responsible for setting up, installing, commissioning and operating the device and the system in which it is installed.

Detailed expertise gained by means of training in the KNX system is a prerequisite.

Introduction

This document outlines the features and programming interface for the spaceLYnk and Wiser for KNX controllers (referred to as the “controller”).

The software programming interface is integrated within the controller and can be accessed via a web browser.

It is essential to note that pre-programming and configuration tasks are only possible with the controller. Before proceeding with installation, operation, or maintenance, carefully review the instructions and become acquainted with the device

Product Features

Connectivity

- IP LAN connection 10/100 Mbit
- USB 2.0 (for GMS modem, EnOcean…) 5V, 500 mA max.
- RS-232
- Modbus (RS-485)
- Wi-Fi through IP connection and wireless router
- KNX/EIB TP Bus

Security Recommendations

To enhance the security of your controller, consider the following best practices:

1. **Network security**:
   - Set up network security at an appropriate level.
   - Ensure that your controller is part of a secure network with limited access.
   - If connected to the Internet, strictly recommend using either a VPN or an HTTPS communication.
2. **Secure protocol access**:
   - Use the secure protocol HTTPS://IP:Port to access your controller.
3. **Security measures**:
   - Evaluate the security capabilities of other network elements, such as firewalls and protection against viruses and malware threats.
   - Store backup files in a safe location inaccessible to unauthorized individuals.
4. **Public IP address**:
   - Verify that your controller does not have a publicly accessible IP address.
   - Avoid using port forwarding to access your controller from the public Internet.
5. **Network segmentation**:
   - Place your controller on its own network segment.
   - If your router supports a guest network or VLAN, consider locating controller there.
6. **Cybersecurity incidents and vulnerabilities**:
   - Report any cybersecurity incidents or vulnerabilities through this page: https://www.se.com/ww/en/work/support/cybersecurity/security-notifications.jsp.
7. **HTTP communication warning**:
   - If HTTP communication is detected, switch to HTTPS (encrypted mode).
   - Note that your controller comes with a self-signed SSL certificate, which encrypts information. Web browsers may display a warning message when confirming the exception to proceed.

8. **KNX installation security**:
    - When accessing the KNX installation via the Internet, be aware that data traffic can be read by third parties.
    - Always use a VPN connection with secure encryption for all data packets.
    - Hardware requirements for VPN routers and features offered by mobile service providers may vary significantly.

For additional details on system hardening, refer to Schneider Electric's document: System Hardening Guidelines for Wiser for KNX and spaceLYnk Controllers.

Password Guidance

When creating and managing passwords, follow these best practices:

1. **Complexity**:
    - Your password can be any combination of **uppercase and lowercase characters**, **numbers**, and **special characters**.
    - Use a **minimum of 8 characters**.
2. **Security**:
    - Make your password **hard to guess** or find in cybercriminal dictionaries.
    - Prefer using **phrases** over single words.
3. **Regular changes**:
    - Change your password **frequently**, at least once a year.
4. **Default admin password**:
    - After receiving the default admin password or performing a factory reset, **immediately change it**.
5. **Avoid reusing passwords**:
    - Never re-use your passwords.

The VPN Access

The VPN access (VPN = Virtual Private Network) authorises the portable device to access the local network and, therefore, the KNX installation via the Internet.

Benefits of VPN

- Only authorised users have access to the local network.
- All data is encrypted.
- The data is not changed, recorded or diverted during the transfer (often referred to as a VPN tunnel).

Requirements for setting up a VPN connection

- Internet connection.
- The portable device and the router are enabled for a VPN connection (VPN client installed).

Maintenance

If you encounter any issues or have questions related to the operation of the controller, please reach out to your supplier or the Schneider Electric helpdesk in your country.

Additionally, exercise caution when accessing your local network remotely, as it may pose higher security risks

Patch Management

Effective patch management is crucial for maintaining the security and performance of your systems. Here are some essential guidelines:

1. **Install updates:**
   - Refer to Install Updates, page 29 to install patches and firmware add-ons.
   - Always perform upgrades **manually**.
2. **Backup before upgrading:**
   - **Backup, page 29** outlines the backup procedure.
   - Prior to any upgrade, ensure you have a reliable backup.

Factory Reset

See chapter Reset/Clean-up, page 27 for an instruction on how to reset the controller.

Firmware Upgrade

See chapter Upgrade Firmware, page 33 for an instruction on how to upgrade the firmware of your controller.

Differences between spaceLYnk and Wiser for KNX

Feature	Wiser for KNX	spaceLYnk
Modbus GUI	Up to 10 Modbus devices	Up to Modbus addressable range ≥ 31 by default
BACnet Server	Up to 150 exported objects	*No limit (≤ 2000 recommended)
User Administration	Up to 8 users	No limit (≤ 20 recommended)
One click adding to the filter table (object filtering)	N/A	Fully supported

*Performance testing was measured on HW3.0 when the object value change interval was set to 60 s. The recommended limit for HW1.x remains 500 objects.

New Features – FW Version 3.0.0

A full list of new features with descriptions is available in the changelog.

Feature	Link to the chapter
KNX data secured group address feature	Add New Object, page 52: **KNX data secure key**
Read-only trends	Trend Logs, page 60
IP settings in KNX connections	KNX Connection, page 36
802.1X EAP TLS network setting added	802.1X EAP–TLS, page 47
User detail - new feature - Home page setting	Add New User, page 82
Acceptance of the NTP server from DHCP server	Interfaces, page 34

Getting Started

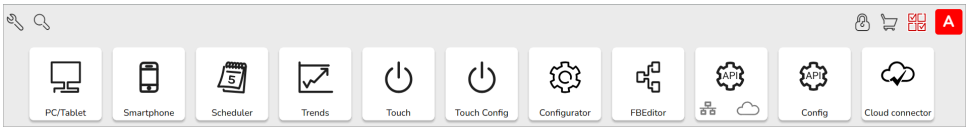
Follow these steps to set up your controller:

1. **Mounting:**
   - Securely mount the controller on the DIN rail.
2. **Cable connections:**
   - Connect the necessary bus cables, including KNX, Modbus, and/or RS232.
   - Optionally, connect a flash drive.
3. **Power supply:**
   - Connect a **24V power supply** to the device.
   - Ensure the positive conductor is connected to the **red clamp**, and the negative conductor to the **blue clamp**.
   - Consider using the recommended accessory: Power supply REG/24V DC/0.4A (article No.: MTN693003).
4. **Ethernet connection:**
   - Connect an Ethernet cable from your PC to the controller.
5. **IP address configuration:**
   - The default IP address for the controller is **192.168.0.10**.
   - Adjust your computer's IP address to be in the same range, e.g., **192.168.0.9** with a subnet mask of **255.255.255.0**.
6. **Web browser access:**
   - Open **Google Chrome** or **Mozilla Firefox** (for Windows), or **Safari** (for macOS).
   - Navigate to **192.168.0.10**.
7. **Default login credentials:**
   - User name: **admin**
   - Password: **admin**

Upon login, you will be prompted to change your password. Refer to chapter Default Configuration, page 17 for details on default configuration settings.

Start Page

The start page offers a dashboard-like view, highlighting key areas within your controller. Here are the available options:



1. **User name and password setup:**
   - Upon the first login or after a factory reset, you'll be prompted to set your user name and password.
   - The default Admin login credentials are as follows:
     - User name: **admin**
     - Password: **admin**
   - Note that initially, only an Admin account exists. Additional users must be created. Refer to chapter User Access, page 81 for details on user access.

2. **PC/Tablet**:
   - Click this icon to access a rich visualization interface with plans containing individual objects.
   - Ideal for use on PCs, iPads, and Android tablets (preferably with a display size of 10 inches or larger).
3. **Smartphone**:
   - This icon leads to a simplified list visualization designed for iPhone/iPad/ Android smartphones and Android tablets (with a display size of 7 inches or smaller).
   - All objects added in controller visualization are visible here (unless the **Hide in Smartphone** option is enabled). Different icons may be set for smartphone visualization.
4. **Scheduler**:
   - Navigate to a user-friendly interface for managing scheduler tasks.
5. **Trends**:
   - Access a user-friendly display of **Trend logs**, allowing you to compare values over time.
   - Trends can be displayed for up to 10 years.
6. **Touch**:
   - Explore the visualization created in the **Touch Config** environment.
7. **Touch Config**:
   - This icon takes you to the premade widgets visualization creator.
8. **Configurator**:
   - Navigate to the programming, settings, and configuration interface.
   - Access is restricted to administrators.
   - Using the **Configurator** on mobile devices is not recommended.
9. **Function Blocks Editor**:
   - Click this icon to access the **FB Editor**, which provides a graphical and user-friendly alternative to LUA scripting.
10. **KNX IoT 3rd Party API**:
    - Serves as an interface for third-party applications to interact with KNX systems. Enables communication between external software and KNX devices, allowing developers to create custom integrations, applications, or services.
    - Enable the API for cloud and/or local integration.
    - For **KNX IoT 3rd Party API** documentation, please reach out to the customer care support team.
11. **KNX IoT 3rd Party API Config**
12. **Cloud connector**:
    - Establish a connection to the KNX cloud.

    NOTE: Firmware (Upgrade Firmware, page 33) automatically installs the **Cloud connector**, **KNX IoT 3rd Party API**, and **Touch** visualization applications in the controller together with a new feature that allows the automatic update of the applications. It is highly recommended that you enable automatic updates. Then you do not have to manually update the applications in the controller´s Marketplace in the future.

Start Page Configuration

1. **Configuration**
   - Click this icon to navigate to the configurator page.
   - Select the **language** for the user interface.
   - Choose between **light/dark theme** of the interface.
   - Unlock the **menu for editing** by clicking . The menu will turn light red (admin mode only).
   - Click and click to:
     - Allow users to **show/hide applications** on the main screen. Users can modify the visibility of icons on the main screen (when permitted by the admin). This feature is available only in admin mode.
     - **Change admin password**: This option is available only when in admin mode.
2. **Search**
   - Filter menu applications based on typed letters.
3. **Lock / unlock grid**
   - Lock or unlock the grid to sort the order of icons in the menu.
4. **Show / hide applications**:
   - Allows showing or hiding applications on the main screen. This feature is available in admin mode or user view (when permitted by the admin).
5. **Save view**: Save the default view of the user's home page.
6. **Controller mode**
   - If the controller is in admin mode, clicking A will log off.
   - If the controller is in user mode, clicking U will also log off.

Marketplace

	This icon navigates to the Marketplace page. It is available only in admin mode. If there is no internet connection, only installed applications will be visible, and updates won't be possible.
	The icon on the left side opens the following sub-menu: 1. **Apps**: Displays all available applications. 2. **Installed**: Shows all installed applications (the yellow number indicates the count of installed applications). 3. **Install from file**: Installs an application directly from a file.
Search bar	Filters applications based on typed letters.
	Installs the selected application.
	Updates the selected application.
	Uninstalls the selected application.
Verified	Marks applications verified by Schneider Electric. Applications not verified are not officially supported by Schneider Electric, and their use is at your own risk.

Applications signing:

- Applications installed from a file should be verified by signature, similar to all applications on the Marketplace.
- Installation of applications without a signature is still possible, but ensure that you trust the source and specifically select this option in a dialog window.

Default Configuration

If the default password has not been changed, you will be prompted to enter the admin access password. This applies to brand new controllers or after a firmware upgrade.

Password complexity requirements:

- The password must be **8-20 characters** long.
- It should contain **uppercase letters** from European languages (A through Z, including diacritic marks, Greek, and Cyrillic characters).
- It must also include **lowercase letters** from European languages (a through z, including sharp-s, diacritic marks, Greek, and Cyrillic characters).
- Additionally, the password should have **at least one base 10 digit** (0 through 9).

LAN configuration:

- IP Address on LAN: **192.168.0.10**
- Network Mask on LAN: **255.255.255.0**

Change IP Settings

See Interfaces, page 34 for more details.

Discover Your Controller in IP Network

1. **Windows OS:**
   - Download the utility service browser **Bonjour** from here.
   - If the host is unchanged and there's only one controller on the network, type the following in your browser:
     http://spaceLYnk.local
     http://Wiser_for_KNX.local

   NOTE: The installed Bonjour service is required for this functionality.
2. **Linux OS:**
   - You can download the utility called **Avahi** from here.
3. **Android:**
   - Download the freely available app called **ZeroConf Browser** from Google Play.
4. **iOS/Mac OS:**
   - You can download the freely available app **Discovery DNS-SD Browser** from the iOS App Store.
   - For iPad, install the iPhone/iPod version of the utility.

Import KNX Project from ETS

ETS 3

To utilize controller with KNX TP UART/IP functionality and program other KNX bus devices, follow these steps to add your controller to the ETS **Connection manager**.

ETS programming through the controller is available only when KNX IP features are enabled.

NOTE: The function **Bus monitor** is not included in the controller.

1. Navigate to **Extras** > **Options** > **Communication** > **Configure Interfaces**.
2. Enter a name for the connection.
3. Choose the **Type** and select **KNXnet/IP** from the drop-down menu.
4. Click **Rescan** and then select **spaceLYnk**/ **Wiser for KNX** from the drop-down menu.
5. Click **OK**.
6. In the **Options** > **Communication** window, select the newly created interface as the **Communication interface** from the drop-down menu.
7. To test communication with ETS, click **Test**.
8. Ensure that the bus status is **Online** by clicking  in ETS.

ETS 4

To utilize the controller with KNX TP UART/IP functionality and program other KNX bus devices, follow these steps to add the controller to the ETS **Connection manager**.

ETS programming through the controller is available only when KNX IP features are enabled.

NOTE: The function **Bus monitor** is not included in the controller.

1. Navigate to **Settings** > **Communication**.
2. Newly added controller will be automatically discovered if it is connected to the same network as the PC running ETS 4 software.
3. Choose **Select** to move it to the **Configured connections**.
4. Set the controller KNX individual address and mask by clicking **Local settings**.
5. Select the **Use project connection if available** checkbox to make it the default project connection.
6. If direct communication in the IP network is desired, select the **Use direct KNX-IP connection if available** option.
7. Click **New** to manually add the controller and enter any name for the device.
8. Set the **IP address**, **Port**, and **NAT mode** (if needed).
9. Click **OK** to save the changes.
10. To test communication with ETS, click **Test**.
11. Finally, click **Apply changes** for the changes to take effect.

ETS 5

To integrate the controller with KNX TP (Twisted Pair) UART/IP functionality and program it alongside other KNX bus devices, follow these steps:

1. Add the controller to ETS **Connection manager**:
   - Open ETS 5 and navigate to **Bus** > **Connections** > **Interfaces**.
   - If your controller is on the same network as a computer running ETS5, it can be discovered automatically. If discovered, double-click on the item in the **Discovered interfaces** list to choose it.
   - If not discovered, proceed to the next steps.
2. Manually configure the interface:
   - Click the green “+” icon next to the **Configured interfaces**.
   - Select **IP Tunnelling**.
   - Click **New Connection** (0.0.0.0:3671) (which is created in the **Configured interfaces**).
   - In the settings tab on the right-hand side:
     - Set the **Name** of your connection.
     - Specify the **Server** (IP address of the controller).
     - Set the **Port**.
   - Select the interface you configured in the previous step.
3. Test communication with ETS:
   - Click **Test** in the lower-right corner to verify communication between ETS and the controller.
   - If the test is successful, double-click on the item in the list of **Configured interfaces** to set it as active.
   - The current interface is now set up.

NOTE: Port 3671 is blocked from any public IP address.

ETS 6

To integrate the controller with KNX TP (Twisted Pair) UART/IP functionality and program it alongside other KNX bus devices, follow these steps:

1. Add the controller to ETS **Connection manager**:
   - Open ETS 6 and navigate to **Bus** > **Manage configured connections**.
   - If your controller is on the same network as a computer running ETS 6, it can be discovered automatically. If discovered, double-click on the item in the **Manage configured connections** list to choose it.
   - If not discovered, proceed to the next steps.
2. Manually configure the interface:
   - On the **Manage configured connections** tab, click the green “+” icon.
   - Select **IP Tunnelling**.
   - Click **IP Tunnelling (0.0.0.0:3671)**.
   - In the settings tab on the right-hand side:
     - Set the **Name** of your connection.
     - Specify the **Server** (IP address of the controller).
     - Set the **Port**.

3. Test communication with ETS:
   - Click **Test** in the lower-right corner to verify communication between ETS and the controller.
   - If the test is successful, double-click on the item in the list of interfaces to set it as active.
   - The current interface is now set up.

   NOTE: Port 3671 is blocked from any public IP address.

KNX Data Secure

If you have data secure devices in your KNX installation and need your controller to communicate directly with these devices (sending and receiving data secure telegrams), follow these steps:

1. **Add the controller secure dummy device:**
   - Integrate the controller secure dummy device into your KNX installation.
   - You can use the **SpaceLogic KNX Secure Dummy** device provided by Schneider Electric available in the ETS catalogue.
   - Connect the individual group addresses of the data secure device to your controller dummy device.
2. **Controller as a router:**
   - If your controller functions as a router (facilitating communication between KNX devices), you do not need the controller secure dummy device for secured KNX communication.

KNX Specific Configuration

To configure KNX connection, follow these steps:

1. Navigate to **Configurator** > **Utilities** > **System** > **Network** > **KNX connection** for detailed configuration options.
2. After making changes, look for the **Apply changes** icon in the top-right corner and click it.
3. The controller will automatically reboot after changes are applied.

   NOTE: Starting from firmware version 2.6.1, the controller now supports KNX long frames with a maximum length of 55 bytes. This improvement significantly reduces the download time required for commissioning KNX devices.

Touch Visualization

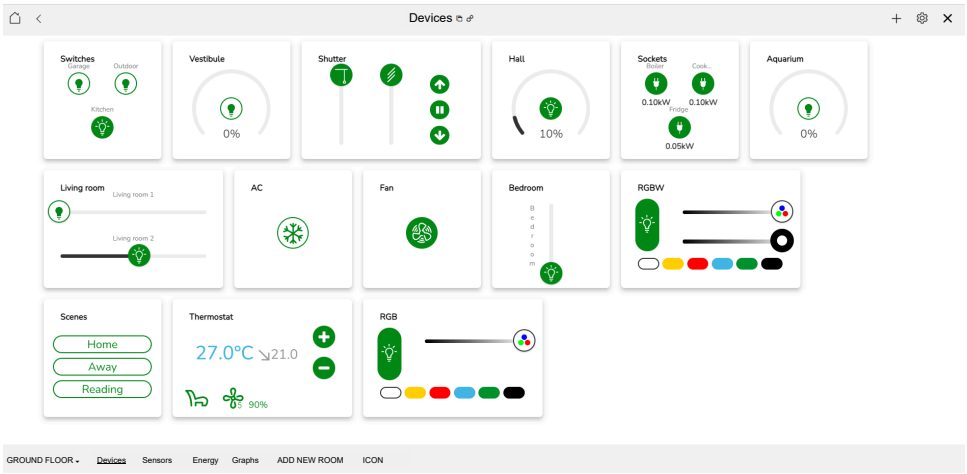
The Touch Visualization offers a streamlined and visually appealing way to create automation interfaces:

1. Ease of use:
   - Setting up the **Touch** visualization is straightforward.
   - You will spend significantly less time compared to visualization as described in Visualization, page 67.
2. Perfect for touch screens:
   - Designed with touch screens in mind, it provides an intuitive user experience.
   - Interact effortlessly with your smart home or building automation system.
3. Pre-made widgets:
   - The included widgets cover essential automation functions.
   - Quickly build your interface without starting from scratch.

Touch Configuration

When setting up your **Touch** visualization, keep the following points in mind:

1. Visualization structure:
   - Your visualization can include multiple floors and rooms.
   - Think of it as a hierarchical structure that represents your building or home layout.
2. Rooms and widgets:
   - Within each room, you have options:
     - **Pre-made widgets**: Use ready-made widgets to quickly populate your rooms with essential automation controls.
     - **Custom widgets**: Create your own widgets using the **Widget creator** tool.
3. Position display:
   - As you build your structure, the current position is shown at the bottom center of the screen.



Adding Widgets

To add a new widget, follow these steps:

1. Open the room you want to add the widget to in **Touch Config**.
2. Click the + button at the top right of the screen.
3. The **Add new widget** menu will open. Select the widget you want to place in the room.
4. A widget dialog will appear, allowing you to set a widget name, assign objects to your widget, and choose a display style for your widget.
5. Ensure that all required objects are linked for the widget to appear correctly in the visualization structure.

   NOTE: Objects marked with an asterisk (*) are mandatory. Additionally, some objects include an **alert** field. Fill in this field, and the object will generate an alert when the alarm condition is met.

Touch Config Menu

When you click in the top right corner of the screen, the **Touch Config** menu opens with the following options:

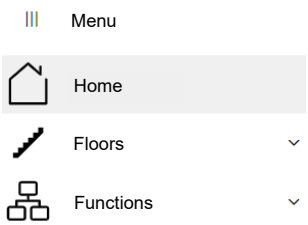
1. **Backup config**: Allows you to back up the **Touch** visualization settings.
2. **Restore backup**: Enables you to restore previous **Touch** visualization configurations.
3. **Settings**: Access general settings for the **Touch**.
4. **Styles**: Manage general styles for visualization and widgets.
5. **Clear all**: Clears settings for rooms, widgets, and styles.

Touch Application

In the **Touch** application, you can view all the features configured in the **Touch Config**, including floors, rooms, widgets, and functions.

On the left side of the screen, you'll find the main menu, which includes:

- **Menu**: This can be locked by clicking the menu icon , at which point the icon will rotate vertically.
- **Floors**: This displays a list of all floors and rooms.
- **Functions**: This section presents groups of widgets, categorized by their respective functions.



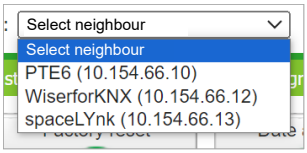
At the top right of the screen, you can notice a lock icon . When this icon is unlocked (click it), it enables you to edit individual widgets using the editing pencil. This allows you to change their names, set widget visibility in the mobile application, initiate voice control, and configure schedulers for widgets.

Configurator – Main Page

Clicking the **Configurator** on the Start page will navigate you to the main page of the **Configurator**.

The top and bottom bars of the **Configurator** main page offer the following options:

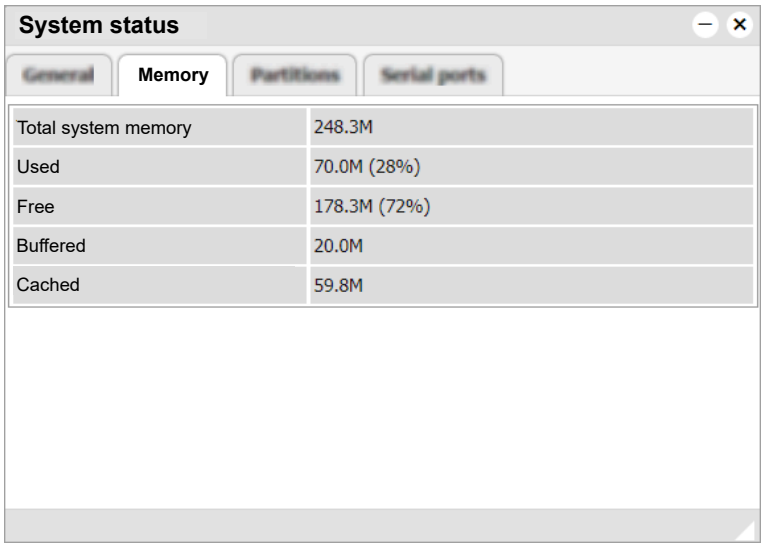
- **Neighbours**: This allows you to switch to the next controller within the same network. This option is only visible if other spaceLYnks or Wiser for KNX devices are detected in the network.



- **Language**: This enables you to change the GUI language. Available languages include English, Bulgarian, Chinese, Czech, Danish, Dutch, French, German, Greek, Hungarian, Italian, Polish, Portuguese, Russian, Spanish, Swedish, Turkish.
- **Start page**: This option redirects you back to the Start page.
- **Logout**: Use this for a secure logout.
- **Version**: This displays the current firmware version of the controller.
- **CPU/IO**: 0.43 0.60 0.69, **Memory** 14%: The load average values of 0.43 0.60 0.69 represent the average system load over one, five, and fifteen-minute intervals. The lower the value, the better the system performance. If any of these values exceed 0.7, the color of the CPU/IO values will change to red, indicating a high load.
  If the load surpasses 0.70, it is recommended to inspect your running tasks. LED1 and LED2 can also be used for an approximate load estimate. For more details, refer to the device operating Instructions.
  The memory usage is calculated as follows:

$$\text{Memory} = \frac{\text{Used memory} - \text{buffered} - \text{cached}}{\text{Total system memory}}$$

This represents the minimum occupied memory in percentage terms.
You can view this under **System → Status → System status → Memory usage**.



NOTE: In Linux terminology, cached and buffered memory is considered “used”, even though it can be freed up for new applications.

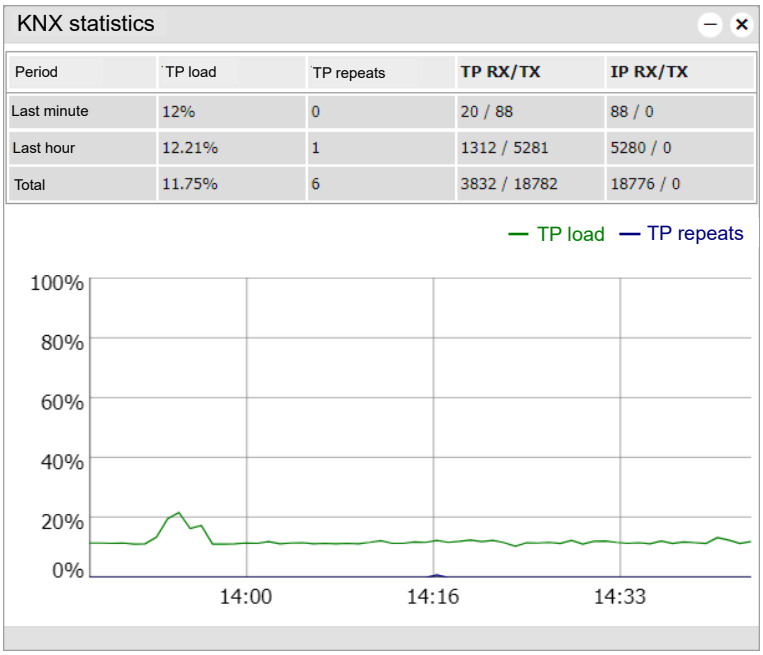
- **KNX/IP**: Each time the **Configurator** is opened, the controller checks if the KNX bus is connected. If not, an error message appears:
  "*Scripting, visualization, and other features will not work. Do you want to switch to KNX/IP instead?*"

The selected connection and its status are visible in the right bottom corner of the screen.

If a KNX/TP error message appears, it indicates that the controller has no connection to the bus. Once the KNX bus is connected, the KNX interface should be changed back to TP-UART under **System** → **Network** → **KNX connection** → **Mode**. This change must be confirmed by rebooting the controller, either manually or by clicking the **Apply changes** icon.

The KNX statistics graph displays the load on the KNX BUS.

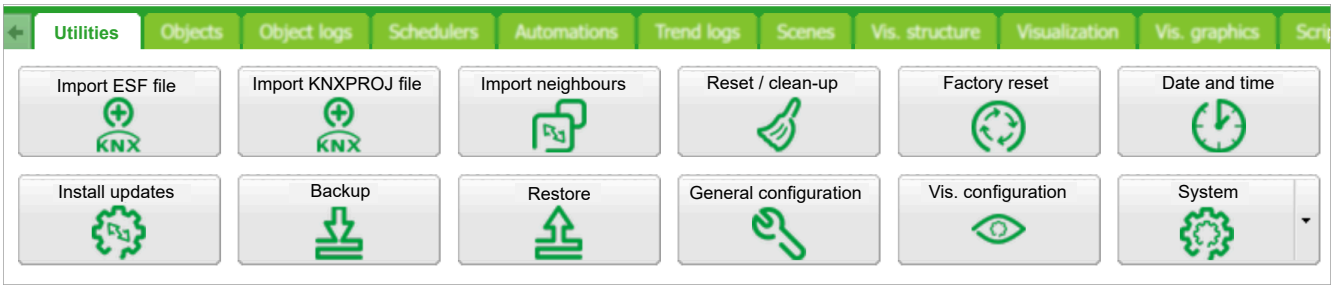
Clicking the graph picture opens detailed KNX statistics.



The **Save project** button is useful when making significant changes in the project. Clicking it immediately saves the project to the microSD card.

NOTE: The automatic synchronization is performed only every 15 minutes, and any unsaved changes and data may be lost in the interim.

Utilities



Import ESF File

Use the **Import ESF File** button to import your *.esf object files.

It is crucial to assign the correct data types to the objects you are importing.

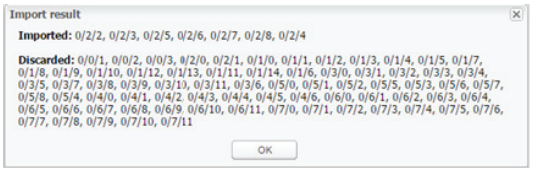
NOTE: Existing objects will not be overwritten.

Objects that share the same name are treated as duplicates and may not be imported. These objects will be marked as discarded.

To import *.esf files, follow these steps:

1. In the **Utilities** tab, click the **Import ESF File** button.
2. Select the file you wish to import.
3. Click **Open** → **Save**.

After each import, a list of both imported and discarded objects will be displayed in the **Import Result** dialog window.



Comments related to ETS imports will be displayed in the **Object comments** section for each imported ETS object:

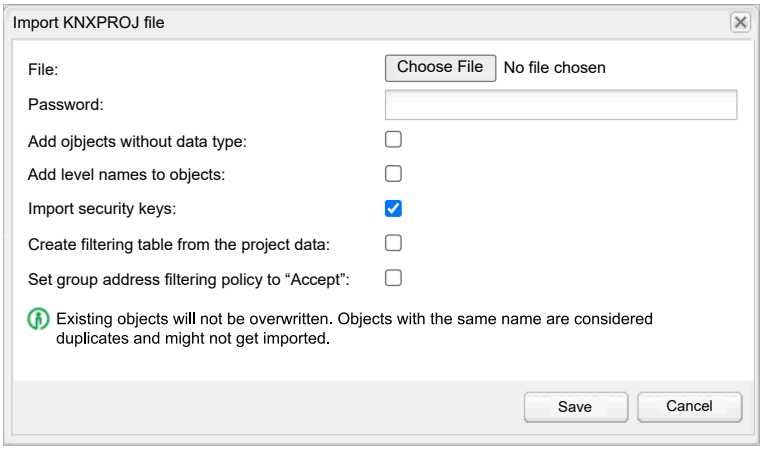
Objects tab → find your object → click it → see the comments.

For more details, please refer to chapter Import KNX Project from ETS, page 18.

Import KNXPROJ File

You can directly import the *.KNXPROJ file into the controller using the Import KNXPROJ file button.

This process preserves the structure of the project and the data point types (DPTs) of the group addresses, including automatic units and suffixes.



- **File**: Choose the file you want to import.

- **Password**: For password-protected *.knxproj files, you will need the password set in ETS. Without the correct password, the import process cannot proceed.
- **Add objects without data type**: The import process allows you to add objects without defined data types.
- **Add level names to objects**: You can assign structure level names to the objects.
- **Import security keys**: By default, this option is enabled. When enabled, the backbone key and KNX data secure keys for group objects will be automatically filled in after the import.
  See also Add New Object, page 52.
- **Create filtering table from the project data**: If you check this option, filtering tables from a KNX project will be automatically created.
- **Set group address filtering policy to "Accept"**: When enabled, the DST group policy will automatically accept selected group addresses in: **Configurator → Network → KNX Connection → IP > Local Filter** (IP > Local Filter, page 39) and **Local > IP Filter** (Local > IP Filter, page 40).
  If this option is unchecked, the DST group policy remains unchanged.

  NOTE: Objects with the same name are considered duplicates and might not be imported. These objects will be marked as discarded.

Import Neighbours

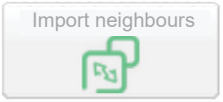
This feature allows you to import objects that have been marked for export from another Wiser for KNX/SpaceLYnk.

The system will prompt you for the remote password of the second device from which data will be exported.

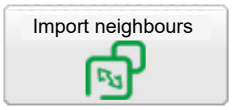
To access remotely, modify the IP and password according to your controller settings. For example: https://remote:remote@192.168.0.10/scada-remote?m=rss&r=alerts

NOTE: The export option must be activated for objects in order to share them between Wiser for KNX/spaceLYnk. Enabling this option makes objects visible via BACnet and remote services (XML/JSON).

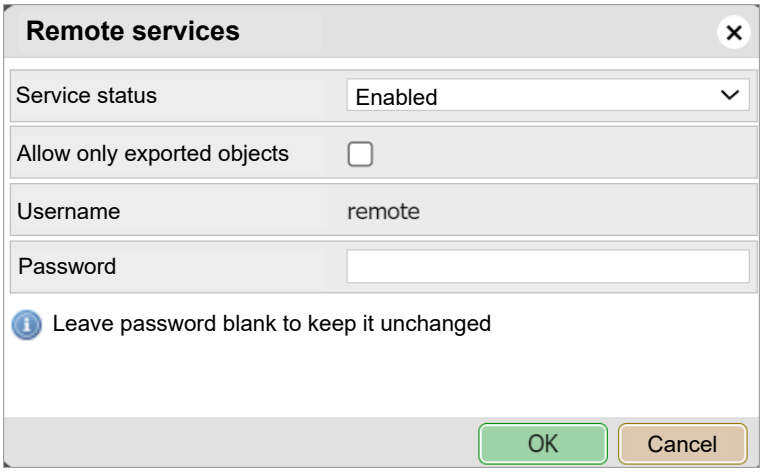
Single Wiser for KNX/spaceLYnk in IP network



Multiple Wiser for KNX/spaceLYnk controllers



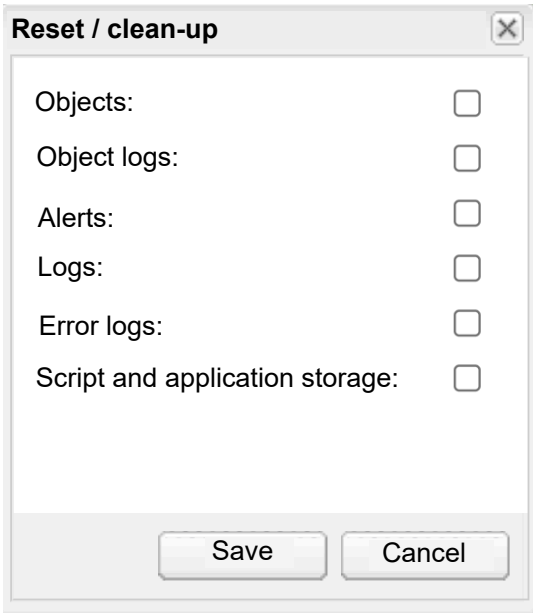
NOTE: Remote services must be enabled on both devices for object sharing: **System → Services → Remote services → Service status: Enabled**.



Reset/Clean-up

This function allows you to delete selected items from the controller.

If you choose to delete objects, they will also be removed from the visualization section.



IMPORTANT: Please ensure that you have backed up all important data before proceeding with the Reset/Clean-Up function.

Factory Reset

The factory reset feature erases all configurations and restores the device to its factory default settings, while leaving the system settings unchanged.

Software factory reset

The software factory reset, which can be initiated from the main menu, does not change the IP address. It performs the following actions:

- Erases visualization, applications, graphics, and all data.
- Resets the device name to the default (SpaceLYnk/Wiser for KNX).
- Resets the KNX settings to default.
- Deletes FTP and Nginx certificates.
- Sets the FTP password to empty.
- Deletes Lua scripts, including planned and init commission.

To reset the IP address to the default (192.168.0.10) while preserving the project, press the HW reset button for 10 seconds. This feature is helpful if you've forgotten or incorrectly set the IP address.

Hardware factory reset

The factory reset with the hardware button is particularly useful in situations where the controller is inaccessible due to incorrect settings.

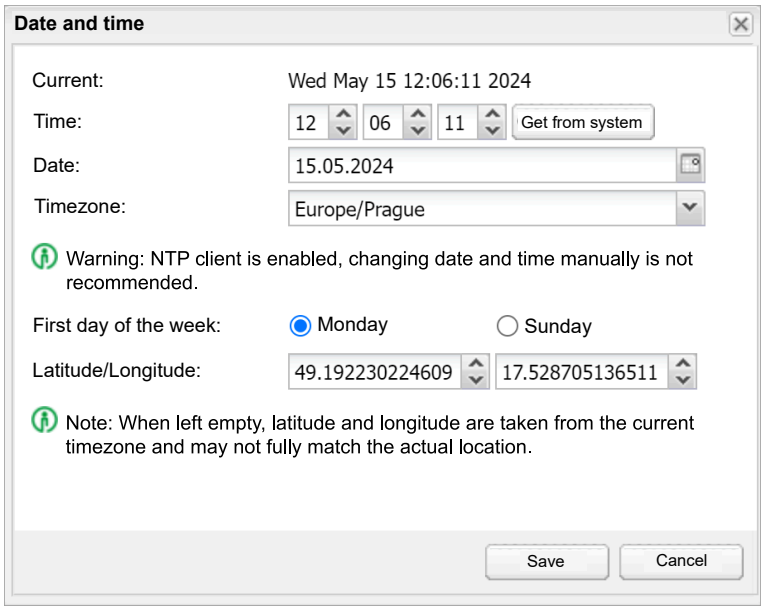
To perform a hardware (HW) factory reset, locate the RESET button on the front side of the controller. Press and hold this button for 10 seconds, release it, and then press and hold it again for another 10 seconds.

After a HW factory reset, the IP address will always be reset to 192.168.0.10.

Date and Time

The controller implements the Network Time Protocol (NTP). With an internet connection, controller automatically updates the time from the servers defined in Utilities/Services/NTP Client/Server:

- 0.schneider.pool.ntp.org
- 1.schneider.pool.ntp.org
- 2.schneider.pool.ntp.org
- 3.schneider.pool.ntp.org



Timezone: It is crucial to select the correct time zone.

Get from system: If there is no internet connection, click on **Get from system** to adopt the time from your PC.

First day of the week: You can select the starting day of the week according to your preference.

Latitude/Longitude: Set up the latitude and longitude coordinates of the geographic position of the controller for accurate calculation of sunrise and sunset times. If not set, these times are calculated from the time zone, which may result in inaccuracies.

The exact geographic position can be easily found on Google Maps by right-clicking on your location on the map.

NOTE: Without power, the controller will maintain time and date settings only for a limited time (approximately 5 minutes).

Install Updates

Click the **Install updates** button to apply partial updates or patches, or to add pre-made solutions.

To update Wiser for KNX, install the update file with the *.lmup extension.

NOTE: Wiser for KNX will reboot after a successful update.

Package file signature:

It is important to note that updates installed from the file are verified by a signature. Updates cannot be installed without the correct signature for security reasons. This ensures the integrity and authenticity of the updates.

Backup

To create a backup, follow these steps:

1. **Backup all data**: This includes objects, trends, logs, scripts, icons, images, backgrounds, visualizations, and the KNX filter table. The backup will be saved as a file named **Project-device name-dd.mm.yyyy-hh.mm.tar.gz**. The date and time used in the filename will correspond to the actual time and date when the backup is generated on the controller.
2. **Choose a password**: During the backup process, you will be prompted to select a password. This password will be required to access the contents of the backup file in the future.
3. **Locate your backup file**: Once the backup process is complete, you can find the backup file in your browser's Downloads folder.

Please note the following important points regarding the backup process:

- **Maximum backup size**: The maximum allowable size for a backup file is 32 MB. After creating the backup, ensure to check its size. If the backup file exceeds 32 MB, you will not be able to restore it in the future.

- **Exclusions from backup**: The backup process does not include system configuration, network settings, installed applications, passwords, or KNX settings. However, the KNX filter table is included in the backup.

Restore

The restore function allows you to reinstate the configuration from a backup file.

To restore the system, follow these steps:

1. In the **Utilities** tab, click the **Restore** button.
2. Select the **Choose File** option and locate your backup file.
3. After selecting the file, click the **Save** button to initiate the restore process.

The maximum backup size is 32 MB. Projects larger than this cannot be restored.

When creating a backup, you will be prompted to choose a password. This password will be required when you wish to restore the project from the backup file.

During the restore process, if LED 1 is flashing red/green, it indicates that data are being recalculated. It is important not to switch off the controller until the process is complete. This ensures the integrity of the restored data.

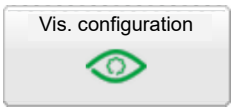
General Configuration

1. **Interface language**: Choose from a variety of languages including English, Bulgarian, Chinese, Czech, Danish, Dutch, French, German, Greek, Hungarian, Italian, Polish, Portuguese, Russian, Spanish, Swedish, and Turkish.
2. **Automatic address range start**: Define the address range for new group objects.
3. **Virtual address range start**: Set the address range for new virtual group objects.
4. **Discover new objects**: Enable the KNX object sniffer to automatically add new objects to the objects list. The bus sniffer is enabled by default, but it's recommended to disable it when not in use, especially if multiple spaceLYnk/ Wiser for KNX controllers are connected in the same network.
5. **Object log size**: This displays the count of object logs. The maximum limit is 50.000.
6. **Default log policy**: Select the log status for all objects or only for checked objects.
7. **Alert log size**: This informs you about the count of logged alerts. The maximum limit is 5.000.
8. **Log size**: This provides information about the count of logs. The maximum limit is 5.000.
9. **Error log size**: This displays the count of logged errors. The maximum limit is 5.000.
10. **Save object values in storage**: The script storage is logged and updated when an object value changes.

11. **Code editor tab size**: Pressing TAB in the scripting editor inserts a defined number of spaces.
12. **Block unsafe functions in scripts**: Enable this option to block potentially harmful functions such as os.exec, io.writefile, etc.

    NOTE: The controller maintains the log objects above the limit for 10 minutes. After this period, all records above the limit are cleared. Excessive object logging can degrade the controller performance.

Vis. Configuration



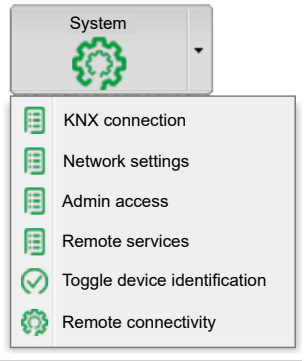
These settings allow you to customize your visualization experience to best suit your needs:

1. **PC/Tablet sidebar**: Enable the sidebar to display a list of plans in the visualization. You can choose to have it docked, auto-hidden, or hidden.
2. **PC/Tablet view**:
   - **Align plans to the top left with no size limits**
   - **Center plans with size limits**
   - **Center plans with auto-sizing enabled**
   - **Center horizontally with auto-sized width**

   Auto-sizing is only compatible with WebKit engine browsers (Chrome, Safari) and Firefox.
3. **PC/Tablet page transition**: Select the transition effect for page changes in the visualization. Options include:
   - **No transition**
   - **Flip X/Y**
   - **Shrink**
   - **Expand**
   - **Slide up/down/left/right**
   - **Slide up/down/left/right big**
4. **PC/Tablet auto-size upscaling**: Enable automatic rescaling to accommodate multiple screen resolutions.
5. **PC/Tablet background color**: Choose your preferred background color for the visualization.
6. **PC/Tablet background image**: Select a background image for the visualization.
7. **Custom font**: Choose a custom font for the visualization.
8. **Use dark theme**: Invert the color, fonts, graphs, and controls to match dark palette visualizations.
9. **Enable swipe gesture**: Enable swipe gestures for touch screen devices, such as zooming with two fingers.
10. **Disable object click animation**: Disable icon animation, which can be useful for slower devices.
11. **Hide Home button in smartphone**: Hide the Home button to prevent exiting the visualization (kiosk mode).
12. **Dim inactive visualization after**: Set a time in minutes after which the inactive visualization dims to save energy on battery-powered devices.
13. **Dimming level**: Set the brightness level of the dimmed screen.
14. **Show alerts in PC/Tablet**: Enable pop-up alerts in the PC/Tablet visualization when new alerts are triggered.

15. **Allow external access via iFrame**: Enable access to the controller's content via iFrame.
16. **Allow external resources (JS/CSS)**: Enable access to external resources. This will disable protection against certain attacks.

System: Quick Menu



By clicking the arrow located on the right, you will open a menu that contains the system settings most frequently used.

KNX Connection

KNX Connection, page 36

Network Settings

Interfaces, page 34

Admin Access

Admin Access, page 33

Remote Services

Remote Services, page 48

Toggle Device Identification

By selecting the **Toggle Device Identification** option, you will activate a red/green flashing sequence on LED 2 of the device.

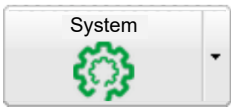
This feature is designed to assist in the easy identification of the controller within the cabinet.

Remote Connectivity

This feature allows you to enable or disable the possibility of remote connectivity, such as for cloud connections.

By default, remote connectivity is **disabled** for spaceLYnk/Wiser for KNX.

System: Service Page

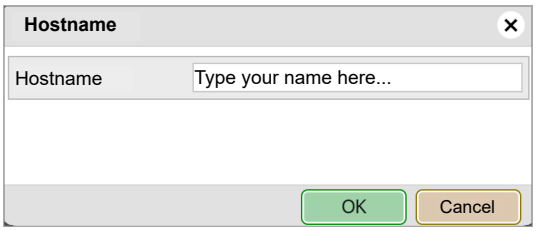


Click the **System** button to access a new page with system settings:

System

Hostname

For easy identification, consider changing the name of your controller.



This name will be displayed in both the neighbour list and the backup file.

Admin Access

The default username is ‘admin’. You have the option to set a new password for the ‘admin’ account.

Login: To login, use ‘admin’ as the default username.

Current password: Enter your current password.

New password: Create your new password. It should be between 8 and 20 characters.

Repeat password: Re-enter your new password for confirmation.

Upgrade Firmware

1. Navigate to **System** → **Upgrade Firmware** to upgrade the system.

   NOTE: The controller firmware downgrade is not recommended.
2. Choose your firmware and signature files and click **OK**.

   NOTE: Starting from firmware version 3.0.0, you are required to add a signature file. Without this file, the firmware upgrade will not be performed.

During the firmware upgrade, the device may not respond as the controller undergoes several reboots. The upgrade process can take up to 5 minutes, especially when a significant amount of trend data is used in the project.

While the upgrade is in progress, LED1 will flash red/green. Please ensure that the controller remains powered on until LED1 ceases to flash red/green.

IMPORTANT: After each upgrade, it is strongly recommended to clear your browser cache to ensure optimal performance.

Backup Configuration

If you wish to create a backup of your configuration, do the following:

1. Navigate to **System** → **Backup configuration**.
2. Choose your password, confirm it by re-entering, and then click **OK**.

The controller will backup the system configuration and generate a backup file.

See also Backup, page 29.

Restore Configuration

If you wish to restore your system configuration from a backup file, do the following:

1. Navigate to **System** → **Restore configuration**.
2. Choose your backup file, enter your password, and then click **OK**.

The controller reboots automatically after configuration has been restored.

See also: Restore, page 30.

Reboot

By selecting **System** → **Reboot**, you can initiate a restart of your controller.

Shutdown

Initiating the **System** → **Shutdown** command will cause the controller to restart.

IMPORTANT: It is strongly advised to execute the shutdown process before powering off the unit. This ensures the database is saved securely.

You can confirm the system is shut down when LED 1 stops blinking and LED 2 is turned off.

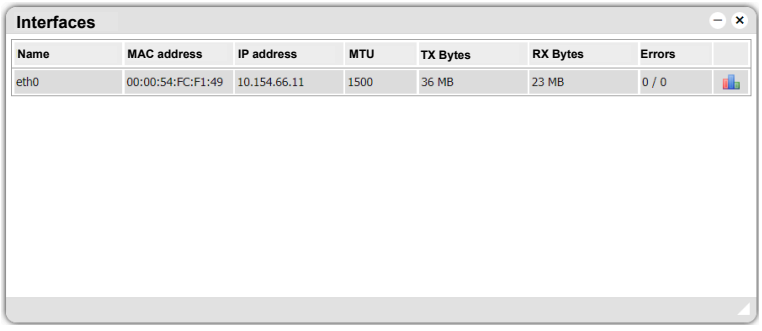
To restart the controller, you will need to physically disconnect and then reconnect the power supply.

NOTE: The controller cannot be switched on remotely.

Network

Interfaces

If you click **Configurator** → **Utilities** → **System** → **Network** → **Interfaces**, the dialog box will open displaying the Ethernet interface.



NOTE: Click  at the end of the interface line to open its traffic flowchart. This displays a real-time graph of the traffic flow passing through the interface (both In and Out).

Clicking the interface in the first dialog will open the configuration window. Set up the parameters as follows:

Protocol: This displays the specific protocol for addressing. The options are:

- **Static IP**: This displays the static IP address, which is 192.168.0.10 by default.
- **DHCP**: This option uses the DHCP protocol to get IP configuration.

Current IP: This field shows the IP address assigned by the DHCP server. It only appears if an IP address has been assigned, otherwise it remains hidden.

Use DHCP NTP Servers: If the DHCP interface protocol is set, the controller will ignore NTP servers. However, you can enable the controller to accept NTP servers from the DHCP server.

IP address: The default IP address is 192.168.0.10.

Network mask: The default network mask is 255.255.255.0 (/24).

Gateway IP: This field is for the Gateway IP address.

IPv6: If IPv6 processing is enabled, the controller will generate an IPv6 address/prefix (a 128-bit alphanumeric value). You can also manually enter the address, giving the controller two IPv6 addresses.

IPv6 gateway: This is optional and can be used if you want your controller to have an Internet connection.

Please note, the IPv6 feature is not supported for HW1.xx.

Duplicate IP messages: This is the number of checks for duplicate IP addresses that are performed before the controller sets its IP address.

DNS server 1: This field is for the primary DNS server IP address.

DNS server 2: This field is for the secondary DNS server IP address.

MTU: This stands for Maximum Transmission Unit, which is the largest packet size that can be passed in the communication protocol. The default is 1500.

After making changes, click the **Apply Changes** icon in the top-right corner for the modifications to take effect. The controller will automatically reboot after applying the changes.

Routes

A routing table is a data structure that itemizes the paths to various network destinations. It encapsulates details about the immediate network topology.

The system routing table can be found under the **System → Network → Routes**.

The window is divided into two sections:

- Dynamic routes
- Static routes

Dynamic routes

This section displays a list of network destinations that the system has learned on its own, along with the automatic selection of the ‘optimal route’.

It includes:

- **Interface**: This is the name of the local interface responsible for reaching the gateway.

- **Destination**: This is the IP address of the destination subnet, which, in conjunction with the network mask, defines the network ID.
- **Gateway**: This is the IP address of the gateway through which the network can be accessed.
- **Network mask**: This shows the network mask address.

Static routes

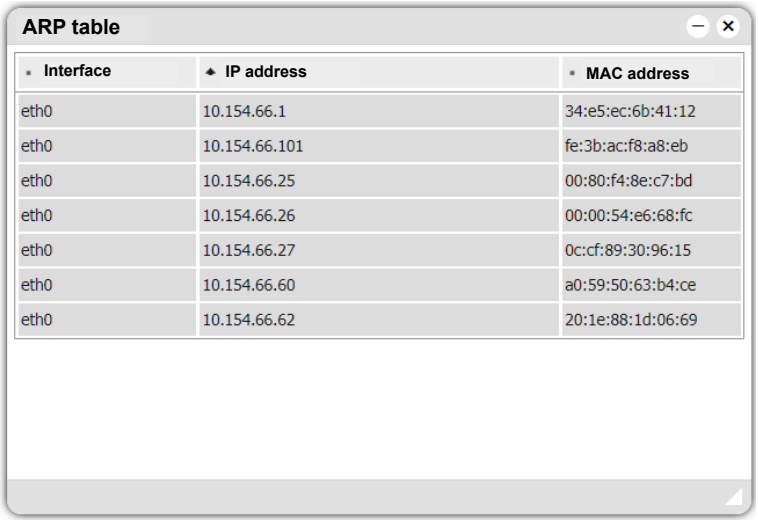
These are routes manually entered into the controller routing table. They do not change automatically.

It includes:

- **Interface**: This is the name of the interface.
- **Destination**: This is the IP address of the destination.
- **Gateway**: This is the IP address of the gateway.
- **Network mask**: This is the network mask address.
- **Flags**: These are useful for troubleshooting network issues.

ARP Table

The Address Resolution Protocol (ARP) table can be found under the **Network** → **ARP table**.

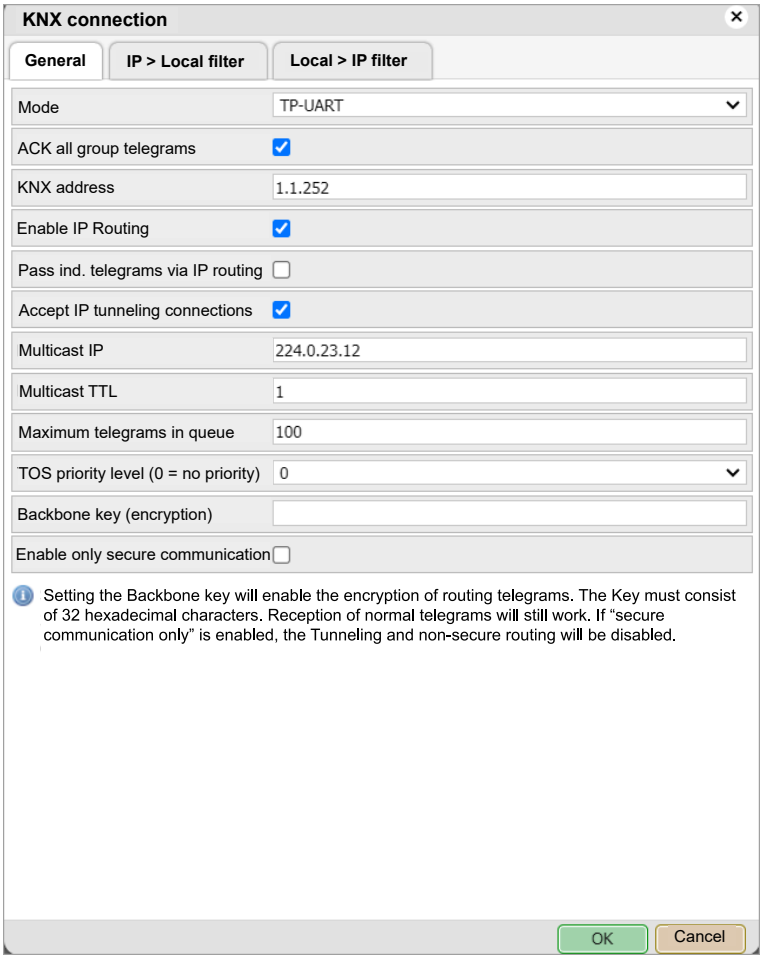


Its primary function is to resolve network layer addresses into link layer addresses.

In simpler terms, it transforms an IP address into a corresponding physical address.

KNX Connection

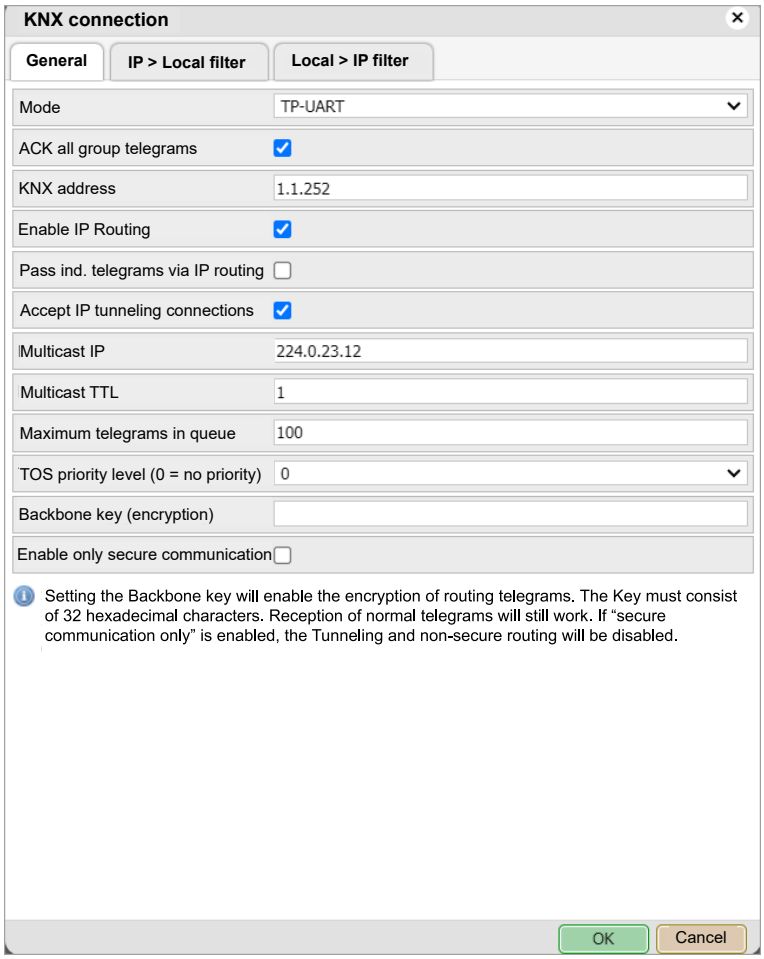
The specific configuration for KNX can be found by navigating to the KNX connection window. The path is as follows: **Configurator** → **Utilities** → **System** → **Network** → **KNX Connection**.



The KNX connection dialog has three setting tabs:

- **General**
- **IP Local filter**
- **Local IP filter**

General



Mode: This section displays the KNX connection mode. By default, the controller comes with a built-in TP-UART interface. The various modes include:

- **TP-UART**: This is a twisted pair connection via a black/red plug. It has a transfer rate of 9.6 kB/s.
- **EIBnet/IP tunneling**: This is an IP connection that is 1,000 times faster than TP-UART. In this mode, the controller behaves as a server, allowing unicast and acknowledged data exchange, with an additional individual address per tunneling connection.
- **EIBnet/IP tunneling (NAT mode)**: Also known as Network Address Translation mode, this allows multiple devices to connect to the public network using the same public IPv4 address. It modifies the IP address information in the IPv4 headers while in transit across a traffic routing device.
- **EIBnet/IP routing**: This mode allows multicast and unacknowledged data transfer. In this mode, the controller behaves as a line or a backbone coupler.
- **ACK all group telegrams**: If the controller communicates directly with another KNX device, it has to acknowledge received telegrams. Unselect this option if the controller operates as a sniffer of group addresses only.

KNX address: This is the individual KNX address of the device.

Enable IP routing: This option allows the receiving/sending of telegrams from/to IP routing (multicast).

Pass individual telegrams via IP routing: This option allows the receiving/sending of individual telegrams from/to IP routing. You must also enable IP routing (for example, for commissioning KNX devices).

Accept IP tunneling connections: This option allows the acceptance of IP tunneling (unicast) connections. This is most often used for commissioning from ETS or diagnostics. For example, you can also create a tunnel from a second controller in IP tunneling mode.

Multicast IP: This is the Multicast IP address.

Multicast TTL: The default value is 1, which allows communication between different sub-networks.

Maximum telegrams in queue: This is the count of maximum telegrams in the queue.

TOS priority level: This is the priority of KNX telegrams, ranging from 0 to 7.

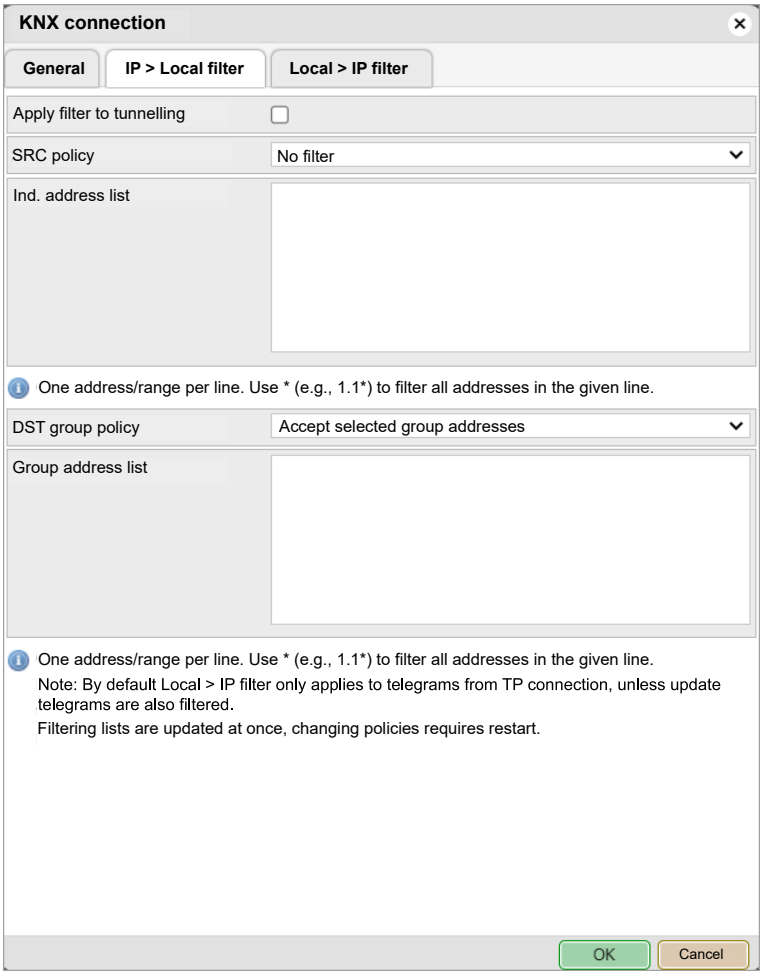
Backbone key (encryption): This is the backbone key for decrypting secured telegrams for IP routing. The backbone key can be exported from ETS software in **Reports** → **Project security**. You can import the backbone key during the KNXPROJ import process.

Enable only secure communication: This option disables tunnelling and non-secure routing.

NOTE: The device will reboot after applying the changes.

IP > Local Filter

This filter either accepts or rejects received telegrams from the defined KNX devices or physical addresses. Please note that all outgoing telegrams are not subjected to this filter.



Apply filter to tunneling: This filter, designed to offer enhanced functionality compared to a standard KNX router, can be applied to the tunneling mode. By default, it allows all telegrams to pass. This option applies to both directions (IP > Local Filter & Local > IP Filter).

SRC policy: This is the policy applied to the list of source addresses. The options include:

- **No filter**
- **Accept selected individual addresses**
- **Reject selected individual addresses**

Individual address list: This list contains the individual or group addresses (one address per line). Use an asterisk (e.g., 1.1.* or 1/1/*) to filter all the addresses in the given line.

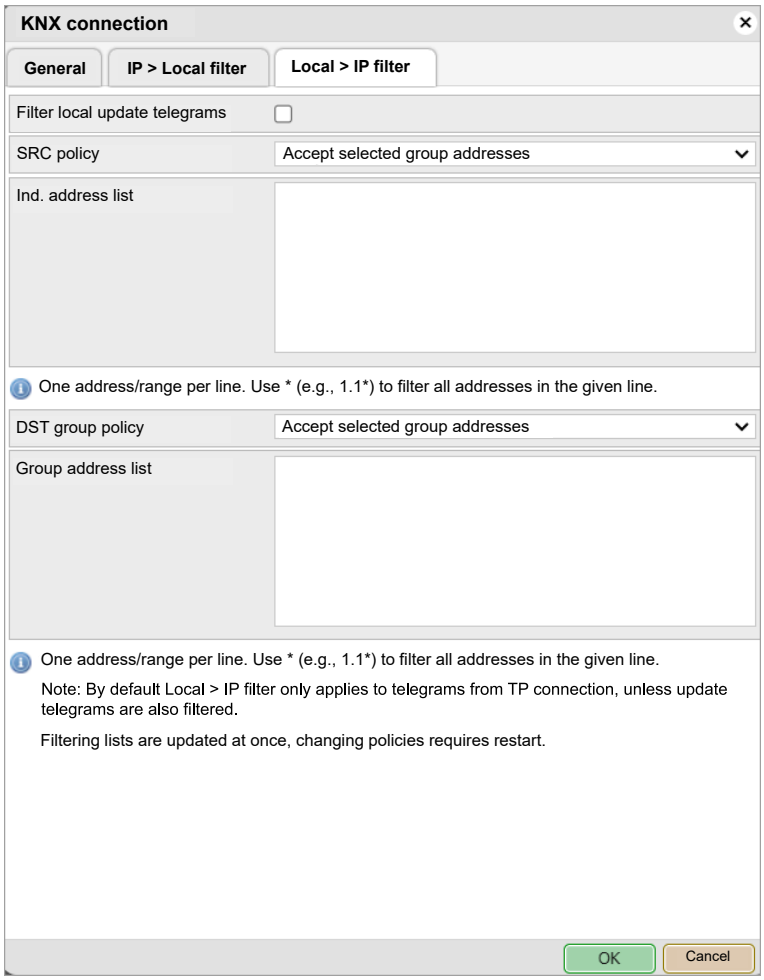
DST group policy: This is the policy applied to the list of destination group addresses. The destination group filter either accepts or rejects received telegrams belonging to one group (e.g., 1/2/3) or subgroup (e.g., 1/2/*). All outgoing telegrams are not subjected to this filter. The options include:

- **No filter**
- **Accept selected individual addresses**
- **Reject selected individual addresses**

Group address list: This list contains the group addresses (one address per line). Use an asterisk (e.g., 1/1/*) to filter all the addresses in the given line.

Local > IP Filter

This filter either accepts or rejects received telegrams from the defined KNX devices or physical addresses. Please note that outgoing telegrams are not subjected to this filter.



Filter local update telegrams: Virtual objects are used for internal data exchange within the controller (e.g., from Modbus to Visualization). If the **`grp.update()`** command is used in LUA, the group address is not written to TP but is written to IP only. If this option is selected, the listed groups are filtered (i.e., not written) from IP and thus become virtual.

SRC policy: This is the policy applied to the list of source addresses. The options include:

- **No filter**
- **Accept selected individual addresses**
- **Reject selected individual addresses**

Individual address list: This list contains the individual addresses. Each address is listed on a separate line. Use an asterisk (e.g., 1.1.* or 1/1/*) to filter all addresses in the given line.

DST group policy: The destination group filter either accepts or rejects received telegrams belonging to one group (e.g., 1/2/3) or subgroup (e.g., 1/2/*). All outgoing telegrams are not subjected to this filter. The options include:

- **No filter**
- **Accept selected individual addresses**
- **Reject selected individual addresses**

Group address list: This list contains the group addresses. Each address is listed on a separate line. Use an asterisk (e.g., 1/1/*) to filter all addresses in the given line.

NOTE: By default, **Local > IP filter** only applies to telegrams from TP connection, unless update telegrams are also filtered. Filtering lists are updated at once, changing policies requires restart.
The KNX IP features must be enabled for these filters to function properly. This applies only to incoming telegrams.

BACnet

BACnet is a communication protocol specifically designed for Building Automation and Control Networks. It is recognized by ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers), ANSI (American National Standards Institute), and ISO (International Organization for Standardization). Here are its key characteristics:

1. Open source standard:
   - BACnet is an open standard, and no license fees are required for its implementation.
   - Many manufacturers have adopted these standards, reducing dependence on specific vendors.
2. Application scope:
   - BACnet facilitates communication among building automation and control systems.
   - It covers applications such as heating, ventilation, air conditioning control, lighting control, access control, fire detection systems, and associated equipment.
3. BACnet standardized device profile:
   The controller has undergone rigorous testing at the BACnet testing labs (BTL) and successfully meets all necessary interoperability requirements. You can find additional details and the results of the BTL testing on the BACnet website.
4. The controller configuration:
   - Interconnection between the controller and other BACnet devices occurs over the Ethernet physical layer.
   - The controller serves as a BACnet server, allowing data to be read by BACnet client devices and written to the server.
   - The connection to the BACnet network originates from KNX group objects in the controller.

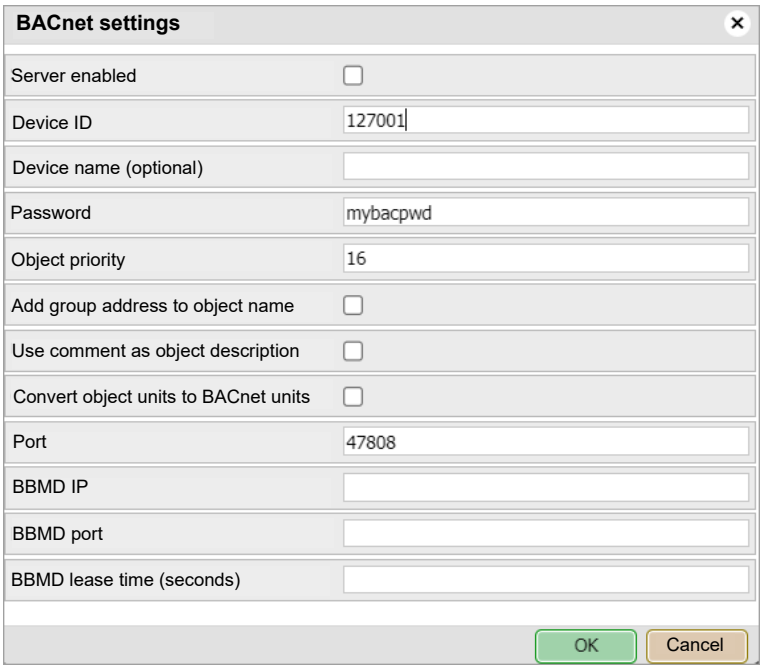
5. Object export:
   - In the controller object list (**Configurator** → **Objects**), each KNX object has an **Export** parameter.
   - Selecting the **Export** checkbox makes the specific KNX object visible in BACnet as a BACnet object.
   - Binary objects appear as binary values, while numeric objects appear as analog values.
   - Other data types are not supported.

BACnet Settings

Configuring BACnet involves setting up BACnet server parameters in the controller. The Building Management System (BMS) discovers the exposed data through this configuration.

Here are the steps for configuring the controller as a BACnet server:

1. Navigate to **Configurator** → **Utilities** → **System** → **Network** → **BACnet settings** dialog.



2. Configure the following parameters:
   - **Server enabled**: Enable or disable the BACnet server.
   - **Device ID**: Ensure that the BACnet device ID is unique on the BACnet network.
   - **Device name (optional)**: By default, the BACnet name is composed of the controller´s hostname followed by the **Device ID**. If you fill in the **Device name**, the BACnet name will match it.
   - **Password**: Set the BACnet password.
   - **Object priority**: Define the priority array to which the controller will write. The controller writes to the Relinquish Default (RD) property by first reading (upload from BMS) only. It takes the current value of the object. The value of the RD property cannot be changed afterward. If an object read from the controller has a higher value than the RD property, it raises the *Overwritten* flag.
   - **Add the KNX address to the object name**: KNX address will be included in object's name.
   - **Use comments as object description**: E.g., ETS import will be used as an object description.
   - **Convert object units to BACnet units**: When enabled, it ensures that the units used by the object (e.g., temperature in Celsius or Fahrenheit) are correctly translated to the standard BACnet units (e.g., degrees Celsius).
   - **Port**: Specify the BACnet port (default is 47808).
   - BBMD Configuration: Set the BACnet router IP (BBMD IP) and port (BBMD port). Configure the **BBMD lease time (seconds)** for registration resend intervals.
3. Click **OK** to save your settings.

BACnet Objects

You can find a list of BACnet objects along with their parameters in the following location:

Configurator → Utilities → System → Network → BACnet objects.

To download BACnet objects as CSV files, simply click the **Download CSV** button. This button will be hidden if your browser does not support this feature.

BACnet COV Settings

When exporting objects to BACnet, they can be subscribed by the BACnet client using **Change of value** (COV) subscriptions.

To find all analog values active for COV subscription, navigate to:

Configurator → Utilities → System → Network → BACnet COV settings.

Each analog value that is active for COV subscription has a parameter called **COV increment**. This parameter defines the minimal change of value (delta) that triggers a value update on the client side. By default, the COV increment is set to 1. This means that unless the value in the controller changes by more than 1, the client-side value remains unchanged.

If you modify the COV increment parameters in **Configurator → Utilities → System → Network → BACnet COV settings** and save the changes, all existing COV subscriptions will be canceled. To resume COV subscriptions, they must be restarted from the client side.

> **IMPORTANT:** Changing COV increment values also resets the priority array values of all objects.

Supported BACnet Interoperability Building Blocks (BIBB)

Data Sharing	ReadProperty-B	DS-RP-B
	ReadPropertyMultiple-B	DS-RPM-B
	WriteProperty-B	DS-WP-B
	COV-B	DS-COV-B
Device and Network Management	Dynamic Device Binding-B	DM-DDB-B
	Dynamic Object Binding-B	DM-DOB-B
	DeviceCommunicationsCon-trol-B	DM-DCC-B
	TimeSynchronization-B	DM-TS-B
	UTCTimeSynchronization-B	DM-UTC-B
	ReinitializeDevice-B	DM-RD-B

BACnet Object Types Supported

- Device
- Analog Value
- Binary Value

Data Link Layer Options

- Media: BACnet IP
- Option: Register as a Foreign Device

Building Operation WorkStation

Schneider Electric's EcoStruxure is a BACnet-certified Building Management System. The Building Operation WorkStation software is used to configure and commission the Enterprise Server and the Automation Server, enabling them to retrieve and send data to the controller.

For detailed information about the interoperability between the controller and Building Operation Workstation over BACnet, visit Schneider Electric website.

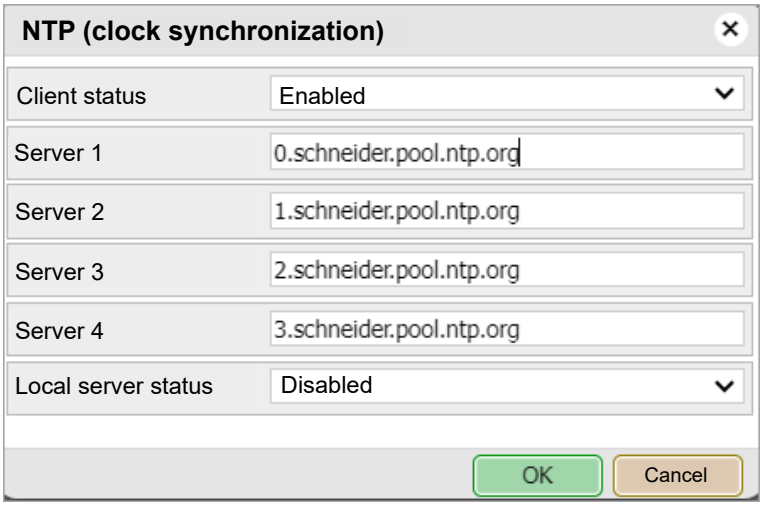
Services

NTP/Client/Server

Client status: Upon enabling the client status, the controller fetches data from a maximum of four selected servers.

Network time protocol (clock synchronization) – Servers 1 – 4: Specify the server from which the controller should retrieve the date and time.

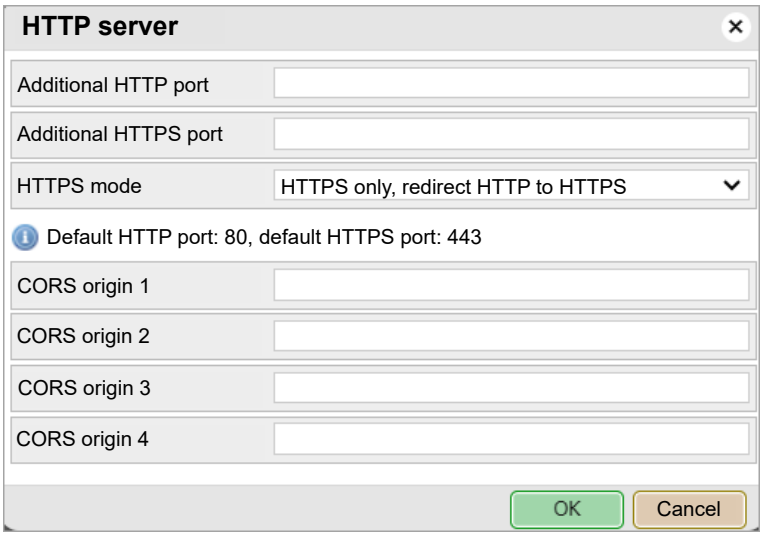
Local server status: By enabling this option, the controller can function as a local NTP server for other devices in the network.



IMPORTANT: A system reboot may be required for changes to take effect. If necessary, use the ping tool to verify the availability of the NTP server.

HTTP Server

The HTTP server allows the use of additional ports for both HTTP and HTTPS.



The default ports are as follows:

- HTTP Port: 80
- HTTPS Port: 443

Available **HTTPS modes**:

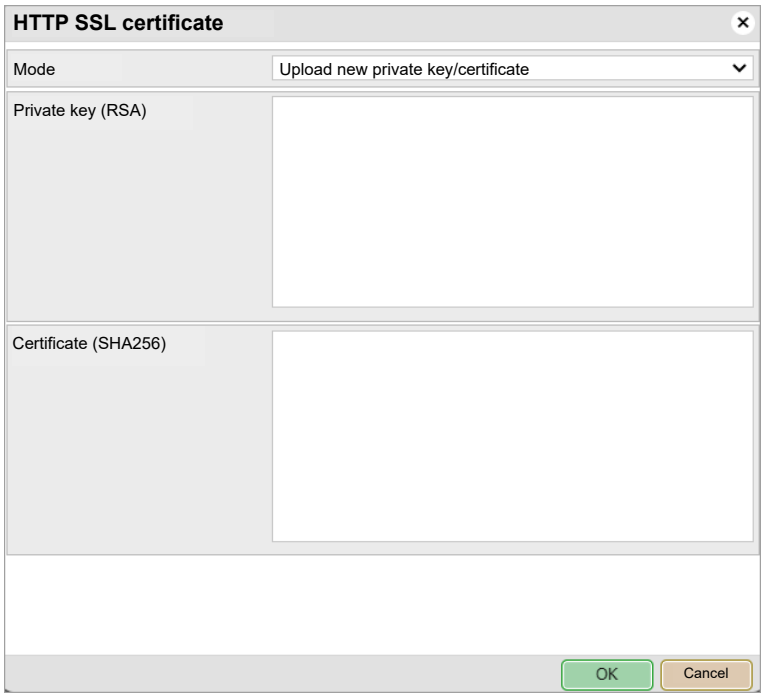
- **HTTP and HTTPS enabled**
- **HTTPS only, redirect HTTP to HTTPS**
- **HTTPS only, HTTP port is disabled**

CORS Origin (1 – 4): You can specify up to four IP addresses or hosts that are permitted to use Cross-Origin Resource Sharing (CORS).

IMPORTANT: A system reboot may be required for these changes to take effect.

HTTP SSL Certificate

SSL certificates are small data files that digitally link a cryptographic key to a device's details. When installed on a web server, it activates the padlock and the HTTPS protocol, allowing secure connections from a web server to a browser.



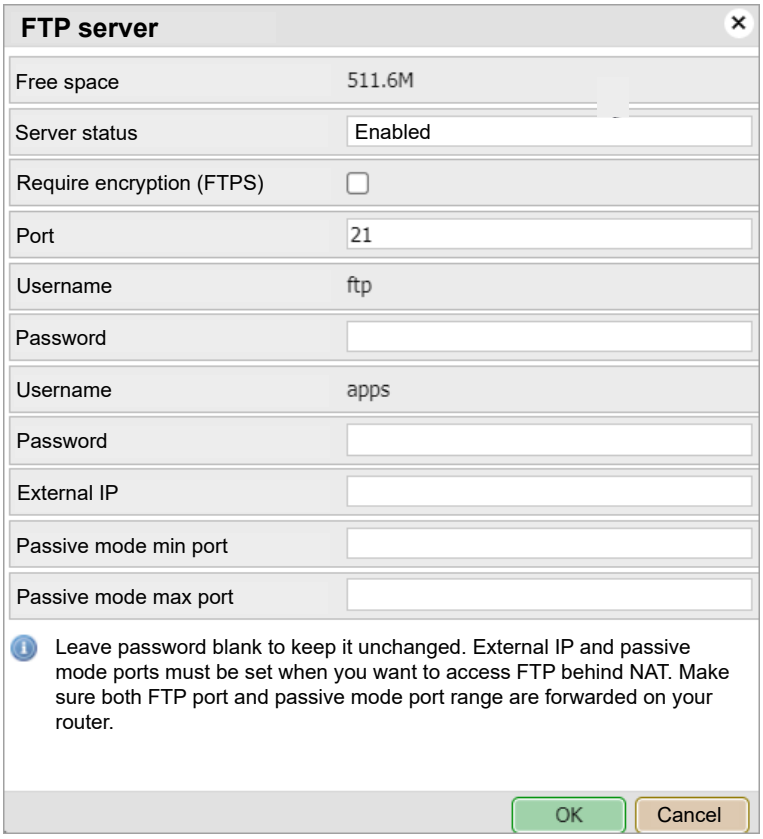
There are numerous online SSL certificate providers, offering both free and paid options.

Mode:

- **Upload new private key/certificate**: This mode is for uploading an existing RSA key or SSL certificate.
- **Generate new private key/certificate**: This mode allows you to generate a new RSA private key or SSL certificate from one that is already installed.

FTP Server

The controller FTP server can be accessed by enabling the **FTP server** under the **Service** menu.



Free space: This indicates the remaining free space on the built-in USB card.

Server status: This refers to the current status of the FTP server.

Require encryption (FTPS): When FTP encryption is enabled, the FTP server will only accept secure connections. If it's not required, the FTP server will accept both FTP and FTPS connections.

Port: This is the port used by the service.

Username: This is the login name (default is 'ftp' and 'apps' for use with SE services).

Password: This is the password, which should be between 6 and 20 characters. The default password is empty and must be changed before using FTP.

External IP: This is the IP address used for external connections.

Passive mode min/max port: These are the minimum and maximum ports for passive mode.

> **NOTE:** The maximum number of simultaneous FTP connections is limited to 5.

802.1X EAP–TLS

EAP-TLS (Extensible Authentication Protocol – Transport Layer Security) authentication is more secure and efficient than credential-based authentication. It automatically occurs without user involvement.

When the controller is within range of the secure network, it initiates and completes the connection autonomously.

To configure EAP-TLS, follow these steps:

1. Navigate to **Configurator** → **System** → **Services** → **802.1X EAP–TLS**.
2. Enable the **Service status**.
3. Fill in the parameters (provided by a network administrator).
4. Click **OK**.

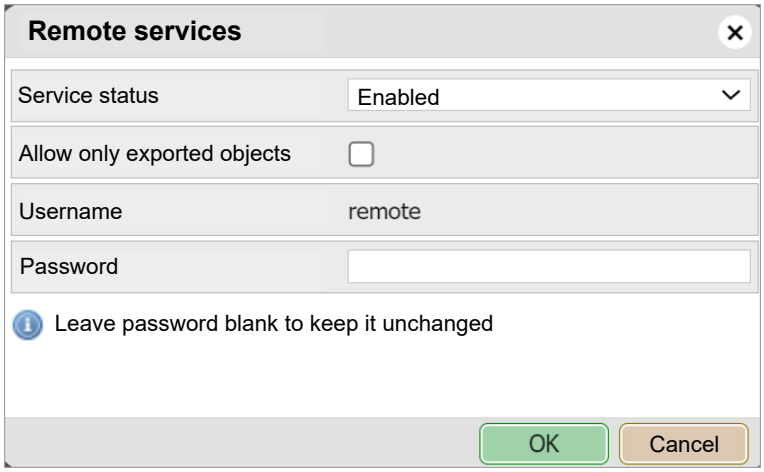
Once configured, your controller can securely connect to an EAP-secured network. This feature is not supported for HW1.xx devices.

Remote Services

You can enable or disable remote access for controller maintenance, control, and export purposes.

Here are the details:

- **Service status**: In this section, you can enable or disable remote services.
- **Allow only exported objects**: If you select this option, only objects marked as exported will be available for remote services.
- **Username**: The default username for remote access.
- **Password**: The password should be 8-20 characters long.



URL configuration: Change the IP address and password according to your controller settings. The URL format is as follows: https://remote: remote@192.168.0.10/scada-remote?m=rss&r=alerts

Request parameters:

m – set the return value format

- json
- xml
- rss

r – requested function name

alerts – for 50 newest alerts

- **alert**: alert text
- **time**: alert time (UNIX timestamp)
- **date**: alert date (RFC date)

errors – for 50 newest errors

- **error**: error text
- **script**: error script name
- **time**: error time (UNIX timestamp)
- **date**: error time (RFC date)

objects – list of return values of export

Marked objects ordered by their update time:

- **address**: object address: e.g., 1/1/1
- **name**: object name: e.g., *My object*
- **data**: decoded object value: e.g., 42 or 01.01.2012
- **datatype**: object datatype: e.g., **1** or **5.001**
- **time**: object update time (UNIX timestamp)
- **date**: object update time (RFC date)
- **comment**: object comment: e.g., *Second floor entry lights*

tags: optional array of object tags: e.g., *Light*, *Second floor*

grp: executes one of grp functions

fn: function name, required

- **getvalue**: returns current object value if found
- **find**: return object info
- **write**: send KNX bus group write telegram
- **response**: send KNX bus group response telegram
- **read**: send KNX bus group read telegram
- **update**: update local controller object value without KNX bus group write
- **alias**: group address or name, required
- **value**: new value to write, required for write/response/update, except for time and date

time datatypes:

- **day number**: (0 – 7), day of the week, optional
- **hour number**: (0 – 23)
- **minute number**: (0 – 59)
- **second number**: (0 – 59)

date datatypes:

- **day number** (1 – 31)
- **month number** (1 – 12)
- **year number** (1990 – 2089)

datatype: optional for write/response/update, data type is taken from the database if not specified: bool, bit2, bit4, char, uint8, int8, uint16, int16, float16, time, date, uint32, int32, float32, access string.

Examples:

Write boolean value to **1/1/2** you can use **true** or **false**, as well as **1** or **0**:

https://remote:remote@192.168.0.10/scada-remote?m=json&grp=write&1/1/2=true

Write value of **50** to **1/1/1**:

https://remote:remote@192.168.0.10/scada-remote?m=json&grp=write&1/1/1=50

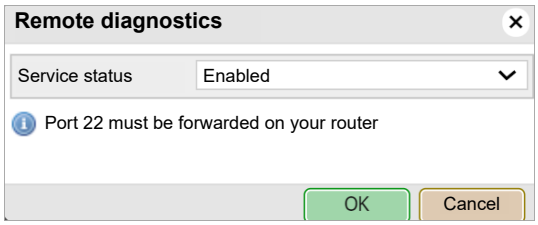
Explicit datatype setting to scale, send **50** to **1/1/1**:

https://remote:remote@192.168.0.10/scada-remote?m=json&grp=write&1/1/1=50&datatype=scale

Remote Diagnostics

You have the option to enable or disable remote diagnostics. To do so, follow these steps:

1. Port forwarding: Ensure that port 22 is forwarded on your router.
2. Enable **Service status**.
3. Click **OK**.

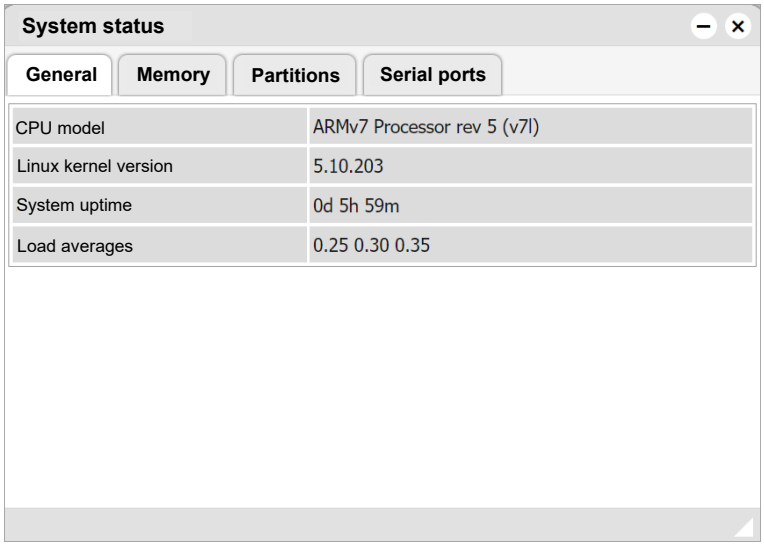


Status

System Status

The system information is displayed in the following tabs:

1. **General**: Provides details about the hardware and system, as provided by the kernel.
2. **Memory usage**: Shows the current memory utilization by the system.
3. **Partitions**: Lists the available partitions in the system.
4. **Serial ports**: Displays the list of available serial ports.

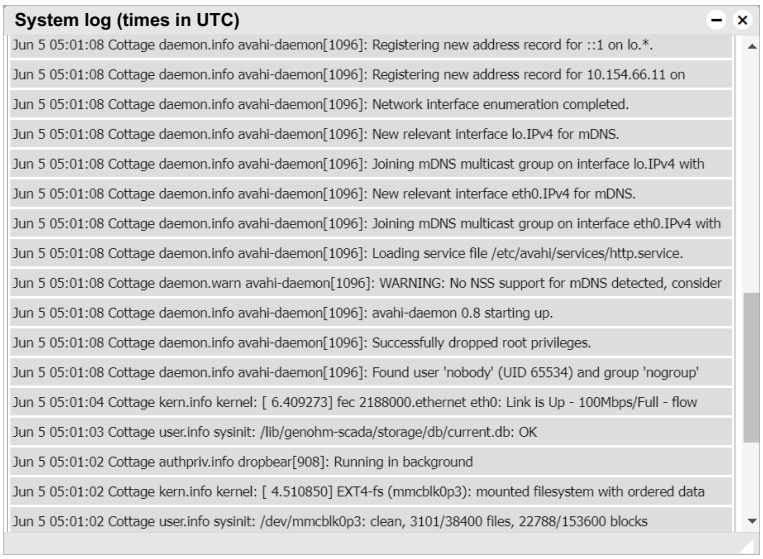


Network Utilities

1. **Ping**: This computer network tool is used to test whether a specific host is reachable across an IP network.
2. **Trace route**: The computer network diagnostic tool displays the route (path) and measures transit delays of packets across an IP network.

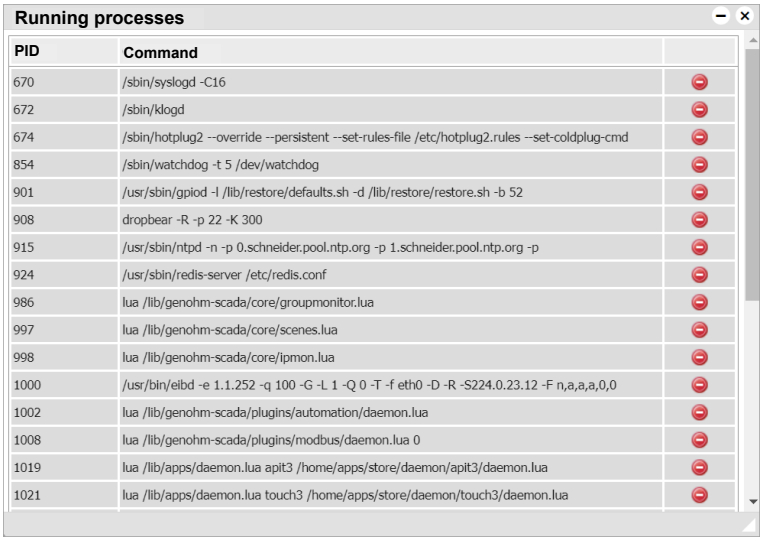
System Log

Log entries are automatically created and maintained by the controller for all system events.



Running Processes

To view the list of running processes, click **Running processes**.



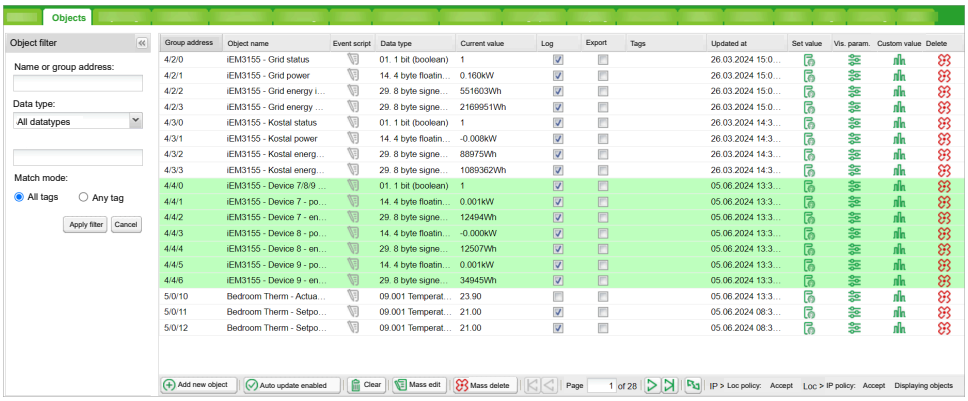
Objects

The list of KNX network objects appears under **Configurator** → **Objects**.

Objects are categorized as follows:

1. Captured by sniffing the bus:
   - If enabled in **Utilities**, objects are captured by sniffing telegrams from unknown group addresses.
2. Manually added:
   - Objects added manually.

3. Imported from ESF File:
    - Objects imported via the ESF file in **Utilities**.



Objects are sorted based on the following parameters:

- **Group address**
- **Object name**
- **Event script**
- **Data type**
- **Current value**
- **Log and export settings**
- **Tags**
- **Last update timestamp**
- **Set value**
- **Visualization parameters**
- **Custom values**

Additionally, objects are visually distinguished by background color:

- **Yellow:** Object discovered by a bus sniffer.
- **Green:** Object value has been updated.

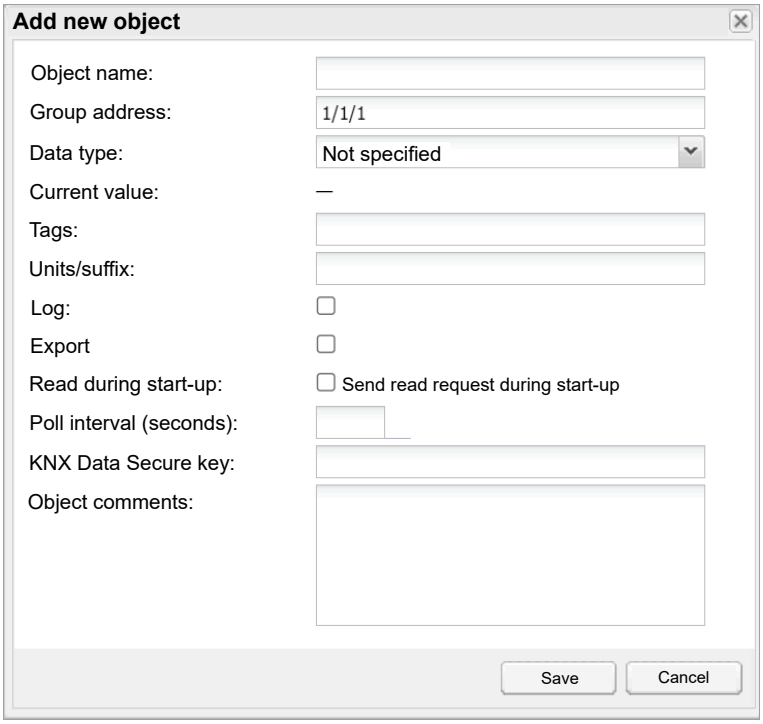
Add New Object

When adding a new object, you have the option to create either a standard KNX object or a virtual object.

Virtual objects:

- Virtual objects are marked with V icon.
- Their range starts from 32/1/1, and they cannot be sent to the KNX TP bus.
- Filtering is disabled for virtual objects.
- Virtual objects are useful for visualization purposes or communication with third parties (e.g., BACnet).

To create a new object, click the **Add new object** button located at the bottom left of the **Objects** tab and fill in the **Add new object** dialog fields:



1. Object properties:
   - **Object name**: Specify the name of the object.
   - **Group address**: Set the group address for your object.
   - **Data type**: Define the KNX data type for the object. This must be set once the controller sniffs the new object for it to function correctly.
   - **Current value**: Indicates the actual value of the object.
   - **Tags**: Assign the object to a tag, which can be used later in writing scripts.
   - **Unit/Suffix**: Add a unit or suffix to the object's value. Units that cannot be created from the keyboard can be pasted from an external editor.
   - **Log**: Enable logging for this object (logs appear in the **Object logs** tab).
   - **Export**: Make the object visible for remote XML requests.
   - **Read during start-up**: The object's actual value will be updated during the controller startup. The KNX object must have the read flag set.
   - **Poll interval (seconds)**: Perform automatic object reads at the selected time interval.
   - **KNX Data secure key**: If you want encrypted group address communication, enter the encryption key (32 hexadecimal characters). Enabling the **Import security key** option during KNXPROJ file import (see Import KNXPROJ File, page 25) will automatically fill in the key. Leave this field blank if you don't need encrypted communication for the group address. Objects with an encryption key have an *S* icon next to their number in the object list.
   - **Object comments**: Provide further descriptions for the object. These comments can also be used for filtering.
2. Configuration:
   - To adjust settings for existing or new objects, do the following:
     Click the specific object list entry → edit your object in the **Edit object** dialog → click **OK**.

Event Script

Clicking  in the object list opens the script editor in the next tab of your browser, allowing you to create an event-based script. The script will run each time a

telegram is sent to the selected group. If the script is attached to a group, the icon turns green .

Set Object Value

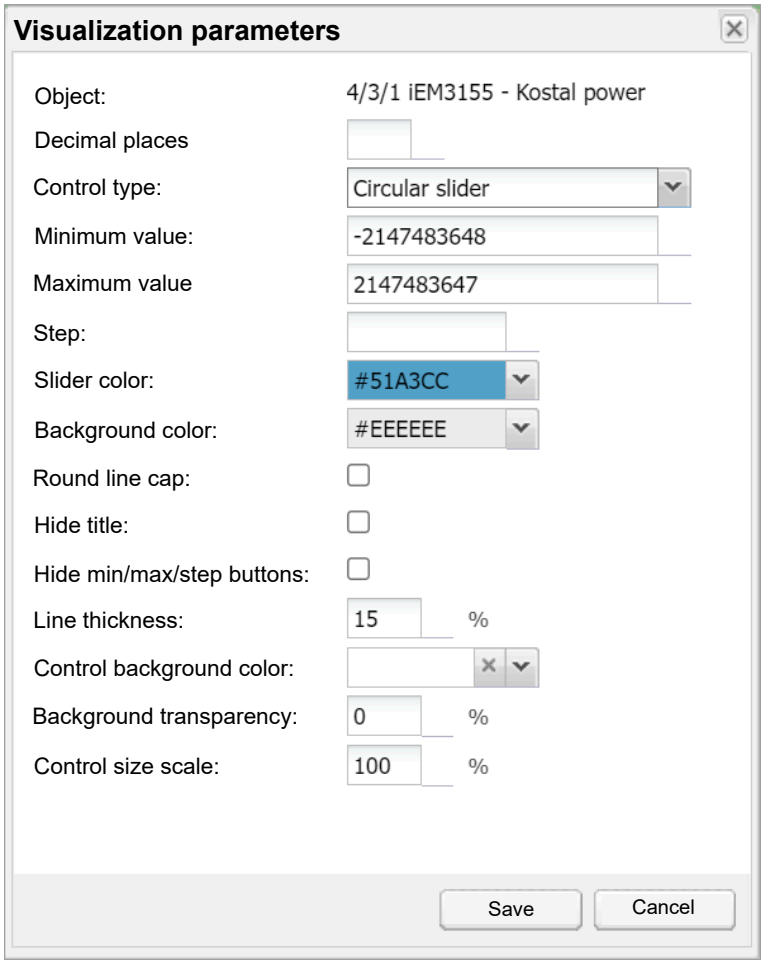
To change the object value, follow these steps:

1. Click corresponding to your object in the object list.
2. In the **New value** field of the **Set object value** dialog, type the desired value for your object.
3. Click **Save** to confirm the change.

Visualization Parameters

To configure the visualization parameters for a specific object, follow these steps:

1. Click corresponding to your object.
2. Adjust the visualization parameters as needed.
3. Click **Save**.

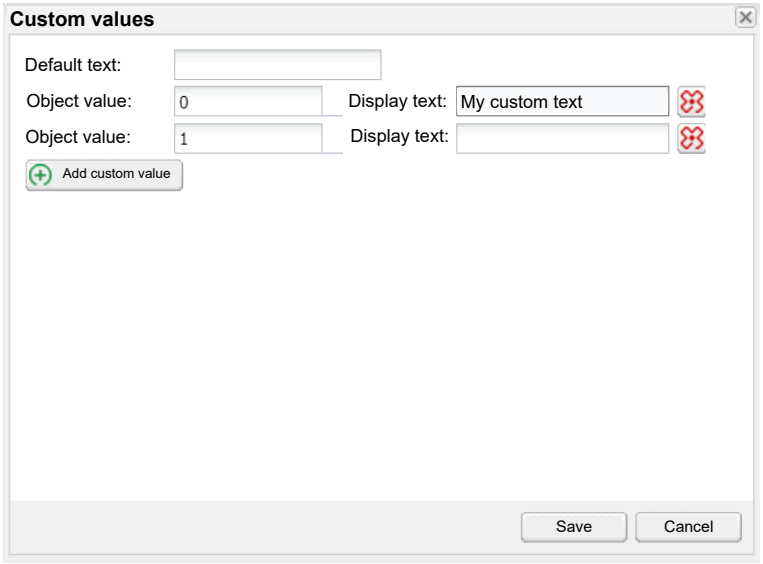


Custom Values

To add custom text to object values, follow these steps:

1. Click in the object list.
2. In the **Custom value** dialog, click **Add custom value**, type the object value, and define the custom text. Then click **Save**.
3. To delete an object value, click corresponding to your value in the **Custom value** dialog.

The **Default text** will be displayed if the value is not defined.



NOTE: Custom text values can only be set for boolean or integer values. You can filter objects based on the **Custom values** (Object Filter, page 55).

Object Control Bar

The object control bar at the bottom of the **Objects** tab includes the following buttons and information:

1. **Add new object**: Manually add new objects to the list.
2. **Auto update enabled**: Specifies whether the object list is updated automatically or not.
3. **Clear**: Clears the list of filtered group addresses.
4. **Mass edit**: Allows mass editing of selected filter objects. See example in Object Filter, page 55.
5. **Mass delete**: Enables mass deletion of selected objects. See example in .
6. **Next/Previous page**: Navigate to the next or previous page.
7. **Refresh**: Refreshes the object list.
8. **IP > Loc policy**: Displays the selected filtering policy.

Object Filter

On the left side of the object list, you can filter objects based on the following values:

1. **Name or group address**: Filter by name or group address. You can use * to replace digits in the address for range filtering.
2. **Data type**: Filter by the data type of objects.
3. **Tags**: Filter by tags. You can choose between matching all tags or any tag.

Click the **Apply filter** button for the filter to take effect.

Mass edit

Objects filtered in the object filter can be mass-edited in the following ways:

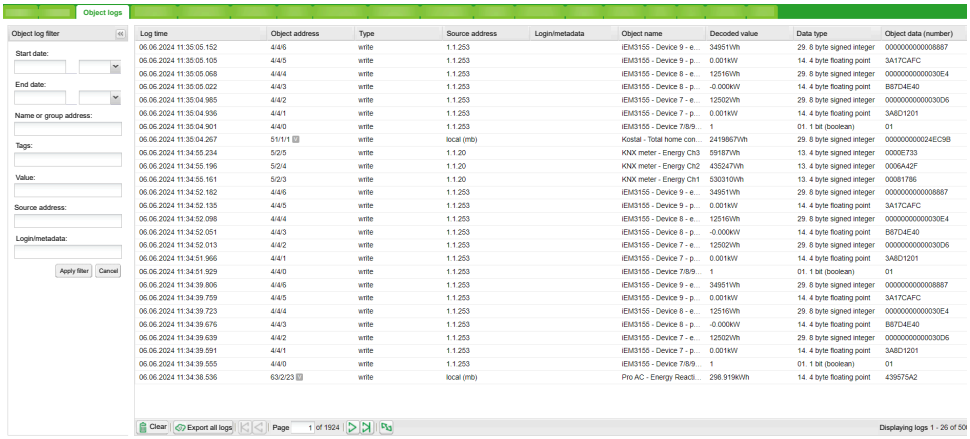
1. **Object properties**: Mass edit based on object properties as listed in the objects menu.
2. **Visualization parameters**: Mass edit based on visualization parameters (e. g., toggle, checkbox, slider).
3. **Custom values**: Mass edit based on custom values for boolean and integer data types.

Mass delete

Objects filtered in the object filter can be mass-deleted using the following options:

1. **Delete unnamed objects**: Delete all unnamed objects from the list.
2. **Delete objects from current filter**: Delete all objects selected by the current filter.
3. Wildcard search in objects: Filter on all objects with a search string (e.g., “**. G*_S*”).

Object Logs



In the **Object logs** tab, you can access historical telegrams for objects. Once logging is enabled for an object, all future data will be logged.

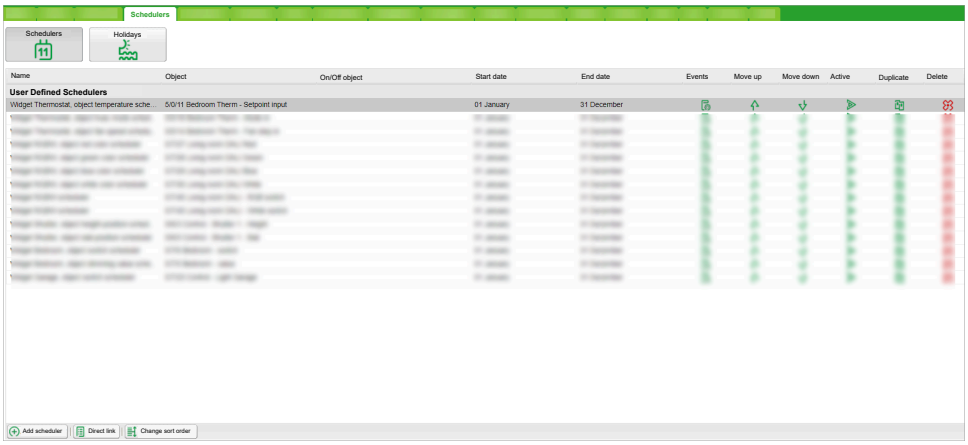
Filtering options are available when you need to find specific information within a period:

1. **Start date**: Specify the start date and time for log filtering.
2. **End date**: Specify the end date and time for log filtering.
3. **Name or group address**: Filter by a specific name or group address of the object.
4. **Tags**: Group objects with the same tags.
5. **Value**: Filter by a specific object value.
6. **Source address**: Filter by a specific source address.

You can clear all logs by clicking the **Clear** button.

To configure logging memory properties, navigate to **Utilities** → **General configuration**.

Schedulers



Schedulers enable end users to control KNX group address values based on the date or day of the week. Users can add events and enable/disable **schedulers** as needed.

Add Scheduler

To create a new **scheduler**, follow these steps:

1. Click the **Add Scheduler** button at the bottom left.
2. Fill in the **Scheduler** dialog with the following details:
   - **Object**: Specify the group address that the scheduler will control.
   - **Active**: Define whether the scheduler is active or not.
   - **Scheduler On/Off object**: Choose an object that can enable or disable the scheduler.
   - **Name**: Provide a name for the scheduler.
   - **Category**: Sort schedulers into created categories for better clarity.
   - **Start date**: Set the start date for the scheduler.
   - **End date**: Specify the end date for the scheduler.
3. Click **Save**.

Your scheduler will now appear in the **Schedulers** list, where you can edit or delete it as needed.

Direct Link

The **Direct link** icon at the bottom left opens a dialog to create a direct link for the scheduler visualization. This link can include the IP address of the host and specify whether to display holidays in the scheduler.

Scheduler → Events

Scheduler events can be added either in the administrator interface or by users in the special user mode Schedulers interface.

Follow these steps:

1. Click to open the **Events** list.

2. Click **Add Event** and fill in the **Event** dialog with the following details:
   - **Active**: Define whether the event is active or not.
   - **Name**: Provide a name for the event.
   - **Run at**: Specify whether the event will be triggered by a **specific time**, **sunrise**, or **sunset**.
   - **Start time offset**: Set an offset for sunrise/sunset (useful for locations in valleys or surrounded by hills).
   - **Start time**: Specify the time of event activation.
   - **Day of the week**: Choose the days on which the event will be active.
   - **Weekday in month**: Select the weekday (e.g., first Monday) for the event (which may fall in the second week of some months).
   - **Months**: Specify the months in which the event will be active.
   - **Year**: Optionally, set a specific year for the event (leave blank for recurring events).
   - **Holidays**: Define the behavior during holidays (options include **No effect**, **Do not run on holidays**, and **Run only on holidays**).
   - **Value**: Determine the value to send to the group address when the event is triggered.
3. Click **Save**.

   NOTE: It is recommended that administrators create all necessary schedulers, as users can only add events but not full schedulers.

Scheduler → Holidays

When an event is scheduled to run on a holiday (Scheduler → Events, page 57), the **Holidays** button for holiday entries will be activated in the **Schedulers** tab. To set up holidays, follow these steps:

1. Click the **Holidays** button.
2. Click **Add holiday**.
3. Fill in the **Holiday** dialog as follows:
   - **Name**: The name of the holiday.
   - **Holiday type**: Select either a specific date or a day of the week.
   - **Day**: The day on which the holiday will be active.
   - **Month**: Specify the month during which the holiday will be active.
   - **Year**: Leave this blank for recurring events.
   - **Duration (days)**: Set the duration of the holiday.
4. Click **Save**.

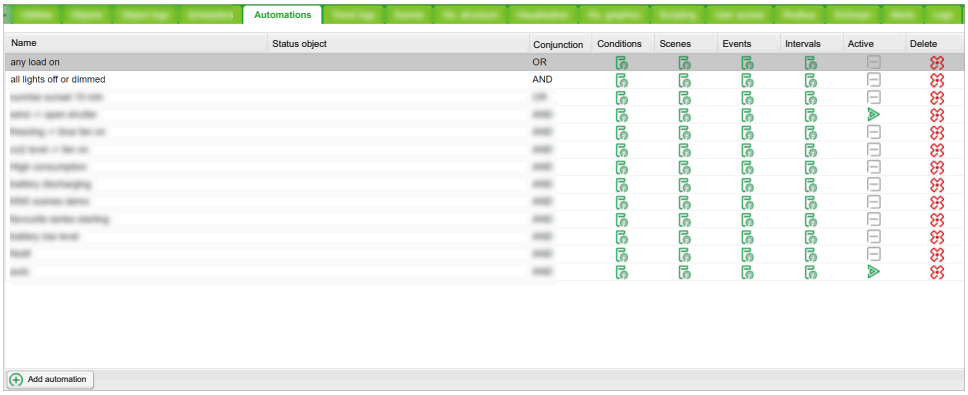
Your holiday will appear in the list of holidays.

NOTE: Using the **Scheduler** visualization on a smartphone is not recommended.

Change Sort Order

Clicking the **Change sort order** button opens a window that allows you to reorder **schedulers** using drag and drop.

Automations



Automations trigger devices automatically when specific conditions are met. Here are some examples:

- If it is sunrise, open my living room blinds.
- If motion is detected on weekends between 9:00 and 17:00, do not switch on the lights.

You can create and edit your automations in the **Automations** tab.

Create Automation

Add Automation

To add new automation, follow these steps:

1. Click the **Add automation** button at the bottom left of the **Automations** tab.
2. Fill in the **Automation** dialog as follows:
   - **Active**: Select whether you want to activate your automation immediately.
   - **Name**: Provide a descriptive name for your automation.
   - **Conjunction**: Choose the logical conjunction for automation conditions:
     - **AND**: All conditions must be fulfilled to trigger an automation.
     - **OR**: Any condition will trigger an automation.
   - **Status object**: Choose a status indication object.
   - **Status value true**: Specify the Status object value for the automation conditions evaluated as true.
   - **Status value false**: Specify the Status object value for the automation conditions evaluated as false.
3. Click **Save**.

Condition

In the next step, you will define the conditions for your automation.

For example: If the level of CO_2 exceeds 1000 ppm.

You can compare objects and/or values:

1. Click in the **Conditions** column of your automation
2. Click **Add condition**.
3. Fill in the **Condition** dialog and click **Save**.

Scenes

You can associate a scene with your automation. When the specified conditions are met, the scene will be triggered.

1. Click in the **Scenes** column of your automation
2. Click **Add scene**.
3. Select your scene and click **Save**.

Events

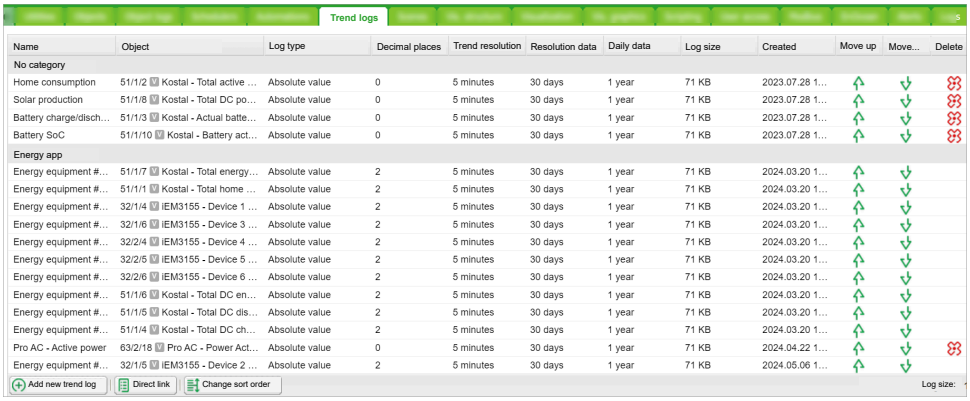
You can also link events to your automation.

Set up the period when your automation should be triggered.

Intervals

Specify the time intervals during which the automation is allowed to trigger.

Trend Logs



Trend logs, also known as data logging, allow you to store selected data and compare different time periods from the past. Access the trend graphs and export data from the **Start page** under **Trends**.

If the **Energy** plugin is installed in your controller, an **Energy** section will automatically be created in **Trend logs**. This is where trend logs for each **Energy** plugin equipment are stored. You cannot view the details of read-only trends, nor can you edit or delete them. They are there to reload data in case of an internet outage. However, if the internet outage lasts longer than 12 hours, the data will be lost and cannot be reloaded.

Trend logs are stored in the internal SD card memory.

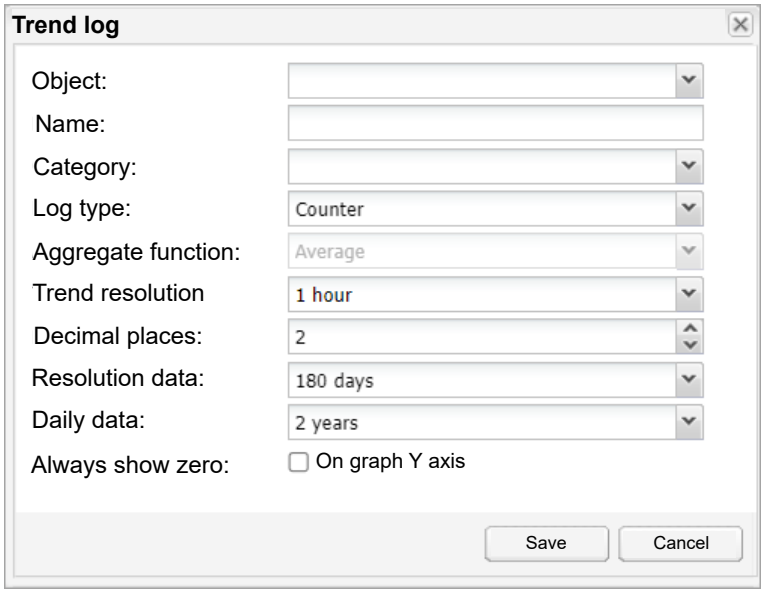
Avoid using trends visualization on smartphones.

Add New Trend Log

To create new trend log, follow these steps:

1. Navigate to the **Trend logs** tab in your system.

2. Click the **Add new trend log** button, located at the bottom left corner.
3. Fill in the **Trend log** details:



- **Object**: Choose an object from the list. This object will be the basis for creating the trends.
- **Name**: Provide a descriptive name for the trend log.
- **Category**: Sort trends into categories for better organization and clarity.
- **Log type**: Select the type of log.
  - **Counter**: Used for counting data.
  - **Counter with negative delta**: Counts data with alternately increasing/decreasing values (e.g., movement detection from a PIR sensor per hour).
  - **Absolute value**: Saves the actual readings.
- **Aggregate function**: Choose from stored data:
  - **Average**
  - **Minimum**
  - **Maximum**
  - **Last value**

  This value will be displayed on the graph.

  Each trend stores data for three periods:
  - Every minute (last hour): Data is added once a minute.
  - Custom interval (user-selectable): Data is added at the specified interval.
  - Once daily: Data is added once a day.
- **Trend resolution**: The average value of counted samples for a specific time interval will be shown on the trend graph. For example, if the interval is 1 hour, the trend step will be 1 hour with an average of 60 readings.
- **Decimal places**: If the object is of floating type, select the desired precision (e.g., 1.1111 = precision of 4).
- **Resolution data**: Short-term data storage (Max = 5 years).
- **Daily data**: Long-term data storage (Max = 10 years).
- **Always show zero**: Leave the option **On graph Y axis** unchecked if your values never reach zero. For example, consider CO_2 levels: If you start at the lowest real value, your trend resolution will appear better.

Direct Link

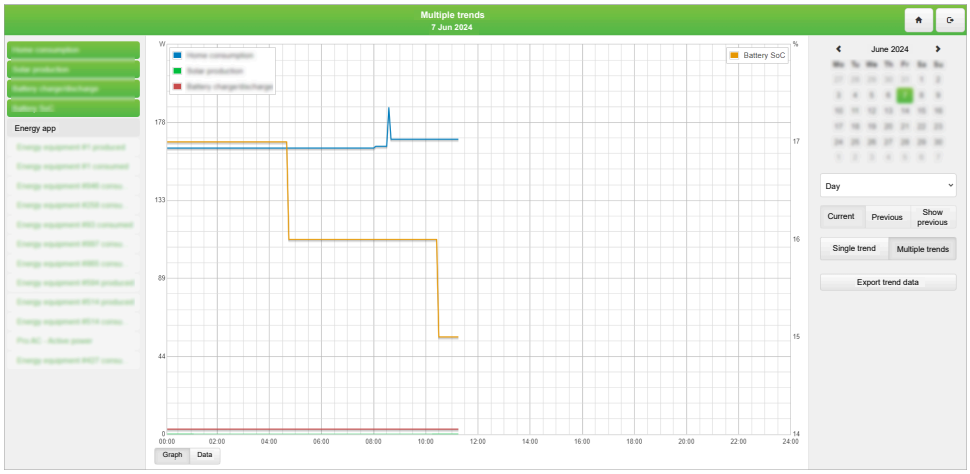
The **Direct link** button opens a dialog to create a direct link for the trend visualization. You can choose the view mode (Day, Week, Month, or Year). The link can include the IP address of the host and optionally display multiple trends.

Change Sort Order

Clicking the **Change sort order** button opens a window that allows you to reorder **trend logs** using drag and drop.

Trend Log Visualization

You can access trend log visualization by clicking **Trends** from the **Start page** of your controller. There, you can choose how to display your trends based on the time period (current or previous), whether to show single or multiple trends, and whether to display them as graphs or raw data.



Additionally, you have the option to export trends in CSV format for further analysis.

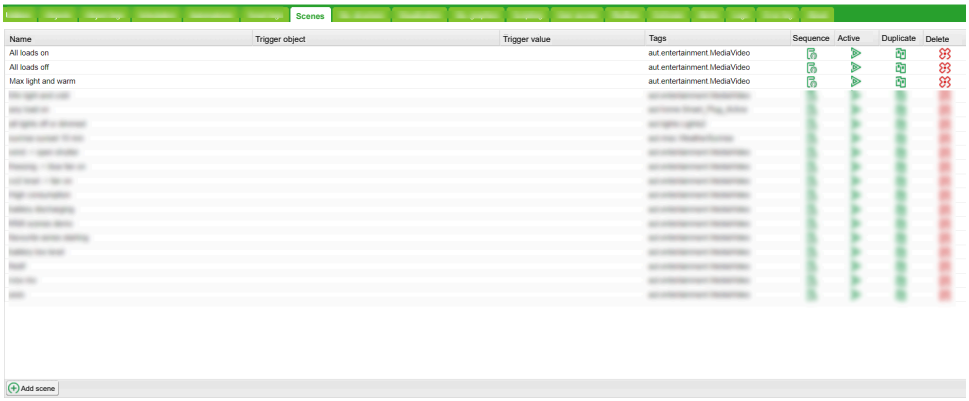
Multiple trends export: Exporting multiple trends is supported. However, keep in mind that the total number of trends is flexible and depends on the system memory capacity. Each trend reserves a portion of system memory based on its settings. If the memory becomes full, the system will prevent you from creating additional trends.

To optimize memory usage:

- Avoid storing data for unnecessarily long periods.
- Use an appropriate trend sampling rate.
- Regularly export your trend data.

Scenes

The scene module enables you to bypass the time-consuming process of setting scenes within ETS and create scenes directly in the controller within seconds.



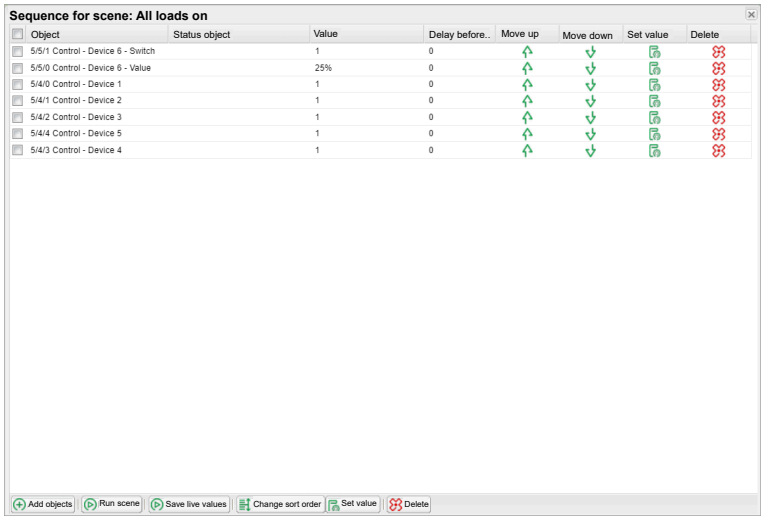
Adding New Scene

1. Click the **Add Scene** button to create a new scene.
2. Fill in the **Scene** dialog with the following details:
   - **Name**: Specify the name for your scene.
   - **Scene is active**: Toggle this option to enable or disable the scene.
   - **Trigger object**: Choose the object that activates the scene.
   - **Trigger value**: Set the value of the object that triggers the scene.
   - **Tags**: Add scene tags (only applicable if the scene is not compatible with object tags).
3. Click **Save**. Your new scene will now appear in the list of scenes, where you can edit, duplicate, or delete it.

Adding Sequences to Scenes

Adding a sequence to a Scene

1. Click  in the **Scenes** tab.

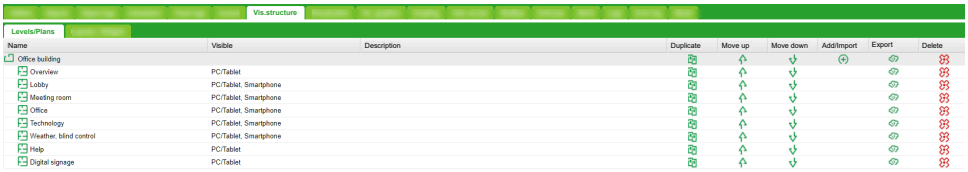


2. Click **Add objects** to add objects and status objects to create your sequence. You can take live values from either the status object or the main object. Additionally, you have the option to specify a delay before writing each sequence.

3. The other following actions are available:
   - **Run scene**: Instantly execute the scene.
   - **Save live values**: Save the actual values of objects within the scene.
   - **Change sort order**: Use drag-and-drop to reorder the sequence in scenes.
   - **Delete**: Remove a sequence.
4. To manually set the value of an object, click in the **Set value** column of the **Sequence for scene** dialog.
5. You can sort the order of objects within the scene using the **Move up** and **Move down** arrows, or delete them as needed.

Vis. Structure

The **Visualization structure** tab serves as the foundation for creating building levels and visualization plans. In addition, it facilitates the creation of layouts and widgets for plan visualization.

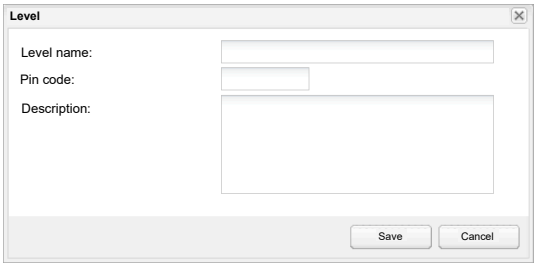


When starting a new project, only the **Layout** and **Widget** folders are initially visible. However, adding a new level allows you to define specific floor plans.

While **Layouts** and **Widgets** are valuable tools, they are not mandatory for basic visualizations. You can choose to define and implement them in other plans as needed.

Levels

To add a new level, click **Add new level** in the **Levels/Plans** tab.



The main level typically corresponds to the project name. Additional levels can be added later.

For importing a level, click **Import** → select your level file → click **Save**.

> **NOTE:** Plans and visualization structures can be imported from another project, with the option to retain or clear linked objects.

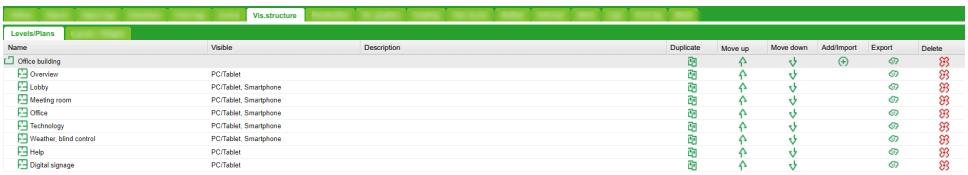
The **Change sort order** button opens a window for drag-and-drop ordering within the visualization structure.

Second level: The second level is utilized in buildings with multiple floors. To create an additional level, click button next to the main level. Provide a name and sort order for the new level.

Duplicate level: Each level can be duplicated along with sublevels and plans by clicking next to the level.

Plans

A plan can represent either a single room within a flat, combining various functions, or a specific function (such as lighting or heating) for the entire flat.



To add plans, follow these steps:

1. Click the button next to the desired level under which the plan should be added.
2. Select **Add plan**.
3. Fill in the **Plan** dialog with the following details:
   - **Name**: Provide a name for the plan.
   - **Plan size**: Choose the size of the plan (pre-defined sizes are available in the drop-down menu).
   - **Layout**: Specify the layout for this plan. Objects from the layout will be duplicated on this particular plan, including background color and the plan image. Ensure that the layout is defined before adding the plan.
   - **PC/Tablet visualization**: Set visibility for this plan in PC/Tablet visualization.
   - **Smartphone visualization**: Set visibility for this plan in Smartphone visualization.
   - **Pin code**: Optionally protect each plan with a PIN code (minimum 4 digits with brute force protection).
   - **Primary background image**: Select the primary background for the plan.
   - **Secondary background image**: Choose the secondary background for a parallax effect. Use a background previously added to **Vis. graphics** → **Images/Backgrounds**.
   - **Background color**: Pick the background color for the plan.
   - **Smartphone background color**: Choose the background color for smartphone visualization.
   - **Repeat background image**: Decide whether to show the image once or repeat it to fill the entire plan.
   - **Fixed primary background**: Use a static primary picture for Parallax projection.
4. Click **Save**.

Your new plan will appear in the **Levels/Plans** tab under the respective level. You can further edit it, duplicate it along with all its components, change its sort order, or delete it.

> **NOTE:** Remember that the content of the created plan should be defined under the **Visualization** tab. An empty plan (with no objects) will not be visible in the visualization.

Layouts

A layout serves as an advanced background for plans. You can place any object from the editor onto the layout, which can then be attached to one or multiple plans. While all objects from the layout will be visible on the plan, the objects on the plan will appear above those from the layout.

To add a layout, follow these steps:

1. Click in the **Layouts/Widgets** tab → click **Add** or use the **Add new layout** button at the bottom left of the tab.
2. Fill in the **Layout** dialog with the following details:
   - **Name**: Provide a name for the layout.
   - **Plan size**: Choose the size of the layout (pre-defined sizes are available in the drop-down menu).
   - **Primary background image**: Select the primary background for the layout.
   - **Secondary background image**: Choose the secondary background for a parallax effect. Use a background previously added to **Vis. graphics** → **Images/Backgrounds**.
   - **Background color**: Pick the background color for the layout.
   - **Smartphone background color**: Choose the background color for smartphone visualization.
   - **Repeat background image**: Decide whether to show the image once or repeat it to fill the entire layout.
   - **Fixed primary background**: Use a static primary picture for Parallax projection.
3. Click **Save**.

Your new layout will appear in the **Layouts/Widgets** tab. You can further edit it, duplicate it along with all its components, change its sort order, export or delete it.

Remember that the content of this layout should be defined under the **Visualization** tab.

Widgets

A widget is a small web page that can be attached to a button and pops up when activated.

To add a widget, follow these steps:

1. Click next to the **Widgets** folder → select **Add**.
2. Or click the **Add new widget** button.
3. Fill in the **Widget** dialog.

   NOTE: The content of the widget should be defined under the **Visualization** tab. Additionally, ensure that the widget size is smaller than the plan on which it is placed.

Your new widget will appear in the **Layouts/Widgets** tab. You can further edit it, duplicate it along with all its components, export or delete it.

An **empty widget** (with no objects) will not be visible in the visualization.

Object Visualization Priority

In the context of visualization, each object has a specific priority. Below, the objects are listed from highest to lowest:

1. Text label on plan

2. Object on plan
3. Plan link as text on plan
4. Plan link as icon on plan
5. Camera on plan
6. Graph on plan
7. Gauge on plan
8. Image on plan
9. Frame on plan
10. Text label on layout
11. Object on layout
12. Plan link as text on layout
13. Plan link as icon on layout
14. Camera on layout
15. Graph on layout
16. Gauge on layout
17. Image on layout
18. Frame on layout
19. Background of plan
20. Background of layout

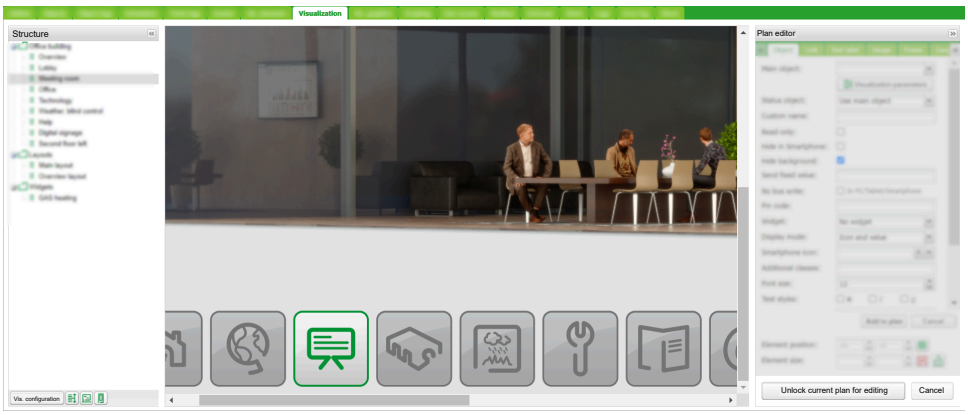
Keep in mind that the order of objects with the same priority is not explicitly defined and may vary between the editor and PC/tablet visualizations.

Visualization

This **Visualization** window is divided into three sections:

1. **Structure**: This section contains a navigation tree for levels, plans, and widgets that were created under the visualization structure tab.
2. Visualization map: The actual visualization field where you can add all visualization components.
3. **Plan editor**: In this section, you can configure all parameters of the component.

Additionally, both sidebars can be minimized by clicking  , which enhances visibility – especially on small displays.



Structure

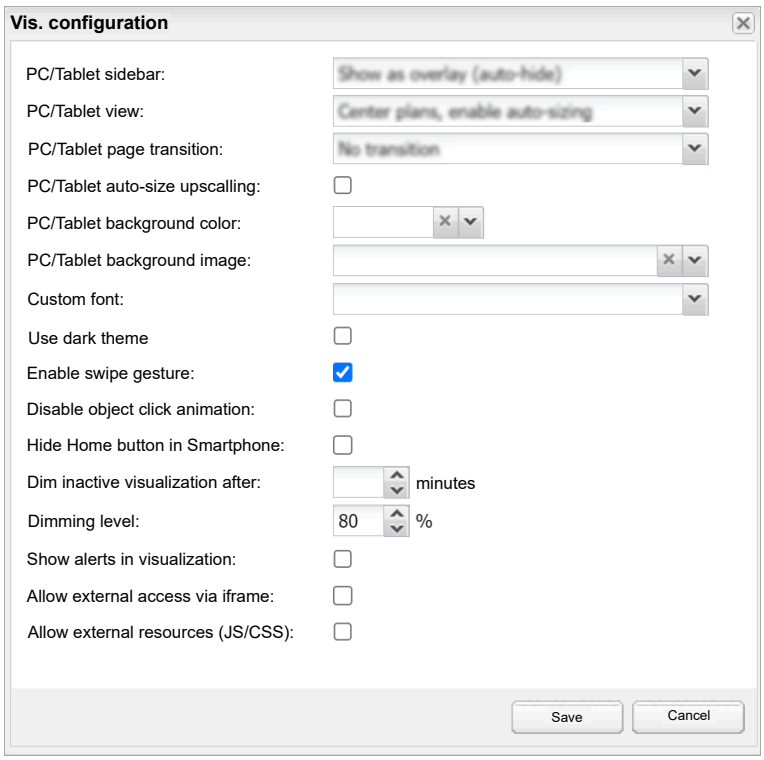
To navigate between plans, layouts, and structure widgets, use the navigation tree in the structured view.

In editing mode, the following additional parameters are available:

- Size of plans, layouts, or widgets
- Source picture/background color

Ensure that the plan size is correctly positioned against the background. Widget size should always be smaller than the plan it is placed on. Use component positioning to align objects.

Click the **Vis.configuration** button to adjust visualization parameters according to your needs.



To reorder objects in smartphone visualization, click .

Next to the **Reorder smartphone objects** icon, you will find two icons for quick access:

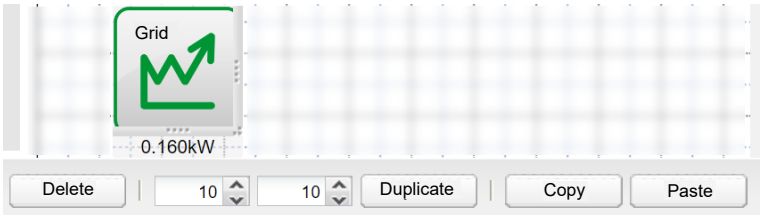
- Preview in PC/tablet

- Preview in smartphone

Visualization Map

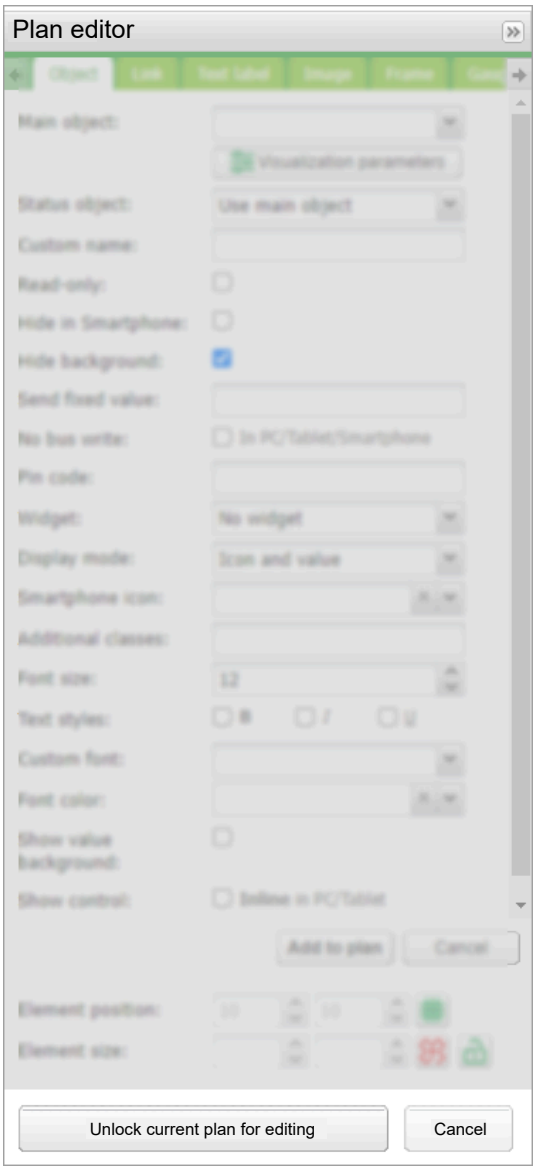
- When adding a new object, it will be placed in the top-left corner of the plan, following the predefined vertical and horizontal spacing from the object menu.
- The selected object can be resized by dragging the strip on the bottom or right side. You can also delete or duplicate it (the duplicated object will maintain the predefined spacing).
- To enable the snap-to-grid feature, click next to the element position in the **Plan editor**.

- The **Copy** button allows you to copy existing visualization objects from one plan to another.
- The **Paste** button becomes available when a visualization object has been selected for copying.



Plan Editor

The **Plan editor** is situated on the right side of the visualization map. To enter editing mode, click **Unlock current plan for editing**.



Object

The **Object** tab allows you to configure control and monitoring settings for various objects. Different data types have distinct parameters:

1. **Main object**:
   - Displays existing group addresses on the KNX/EIB bus available for configuration in the **Objects** tab.
   - To speed up selection, consider typing group addresses directly.
2. Visualization parameters:
   - Adjust object visualization parameters
     - **Control type**: Choose the appropriate control type.
     - **Minimum/Maximum value**: Set the range for the parameter.
     - **Step**: Define the increment value.
     - Colors and size: Customize visual aspects.
   - After clicking the **Visualization parameters** button, choose one of the following options in the **Select which parameters to edit** dialog:
     - **Global (per object) parameters**: These settings apply to all visualization elements with the same object.
     - **Local (per-element) parameters**: Customize settings for specific visualization elements.
     - **Override global parameters with local**: Modify the object based on local settings.
     - **Clear local parameters**: Reset local customizations.
3. **Status object**:
   - Lists status objects on the KNX/EIB bus.
   - Control objects can also serve as status indicators.
4. **Custom name**:
   - Assigns a name to the object.
   - Important for Smartphone Visualization; if left blank, the group address name is used.
5. **Read-only**:
   - Makes the object read-only, without write (control) permission.
6. **Hide in Smartphone**:
   - Prevents the object from appearing in smartphone visualization.
7. **Hide background**:
   - Hides the icon background.
8. **Send fixed value**:
   - Allows sending a specific value to the bus when the object is pressed.
9. **No bus write**:
   - Prevents writing the value to the KNX bus.
   - Useful for triggering scripts with busload limitations.
10. **Pin code**:
    - Adds PIN protection to the object.
    - Requires entering a PIN when changing the value.
    - Minimum PIN length: 4 digits with brute force protection.
11. **Widget**:
    - Attaches a widget to a button (requires prior creation).
    - Widget testing available only in PC/tablet visualization.
12. **Display mode**:
    - Determines how the object is displayed.
13. **Smartphone icon**:
    - Default icon for smartphones (can differ from PC/tablet icon).

14. **Additional classes**:
    - Creates custom CSS classes to modify graphical objects.
15. **Font size**:
    - Defines display text style for values.
16. **Text styles**:
    - Options for bold, italic, and underscore.
17. **Custom font**:
    - Choose from installed fonts.
18. **Show value background**:
    - Enhances readability by showing value backgrounds.
19. **Show control**:
    - If enabled, control button graphics change from symbols to switches (visible in PC/tablet visualization).
20. **Additional icons** (for value-type objects):
    - Specify additional icons while setting parameters.

You can adjust these parameters to customize your object behavior.

After defining the parameters, click **Add to plan** to create a new object.

Cancel button sets the object parameters to default settings.

Objects can be moved within the plan, duplicated, and edited.

Element position:

- You can add elements manually or by dragging and dropping objects to set their X and Y axis positions on the plan.

Element size:

- Adjust the element size manually or by dragging a vertical-horizontal strip of the object.
- The minimum value for size is 5 × 5 pixels.
- To reset an object size to the default, click  .
- Lock the aspect ratio of the object by clicking  .

When ready, click **Save and reload plan** to activate the objects.

Link

To enhance visualization convenience, plan links have been integrated. These special icons on the map serve as links to other plans.

Follow these steps to set up a link within the **Link** tab:

1. **Link to**:
    - Select the plan link you want to associate with this icon.
2. **Custom name**:
    - Type a name for the link.
3. **Hide in Smartphone**:
    - Choose whether to display this plan link on smartphones.
4. **Hide background**:
    - Toggle the icon background visibility.
5. **Display mode** [Icon; Value]:
    - Decide how the plan link should appear.

6. **Icon**:
   - Select the icon to display in the visualization. If only text is chosen, text parameters apply.
7. **Active state icon**:
   - If an icon is selected, the active plan icon becomes available.
8. **Additional classes**:
   - Create custom classes for modifying specific groups of graphical objects using CSS.

After defining these parameters, click **Add to plan** to create a new link.
The **Cancel** button resets link parameters to default settings.

Links can be moved within the plan, duplicated, and edited.

Element position:

- Manually add elements or drag and drop elements to set their X and Y axis positions on the plan.

Element size:

- Adjust the element size manually or by dragging a vertical-horizontal strip of the link.
- The minimum size value is 5 × 5 pixels.
- To reset an element size to the default, click  .
- Lock the aspect ratio of the link by clicking  .

When you are ready, click **Save and reload plan** to activate the links.

Text Label

Text labels can be added and moved across the visualization map. Configure the parameters for your text label in the **Text label** tab as follows:

1. **Text**:
   - Enter the label text.
2. **Font size**:
   - Specify the font size for the label.
3. **Text style**:
   - Choose the style for the text (bold, italic, underscored).
4. **Custom font**:
   - Provide the font name (if applicable).
5. **Font color**:
   - Set the label font color.
6. **Additional classes**:
   - Creates custom CSS classes to modify graphical objects.

Once the label parameters are defined, click the **Add to plan** button, and the newly created label will appear on the map.

You can move the label to your desired location.

Remember to adjust the element size and position (refer to Object, page 69).

Finally, click **Save and reload plan** to activate the label.

Image

The **Image** tab allows you to add images to the visualization map either from local storage or the internet. External images are particularly useful for dynamic weathercast visuals. To configure your image, follow these steps:

1. **Image source**:
   - Choose either **Local** or **Remote** as the image source.
   - If using a remote image, provide the URL.
2. **Select image**:
   - Pick an image that you have previously added to the **Vis. graphics** → **Images/Backgrounds** section.
3. **Image size**:
   - Specify the desired dimensions for your image.
4. **External link**:
   - Define an external link URL that users can access by clicking the image (e.g., http://www.se.com).
5. **Refresh interval**:
   - Set the interval for refreshing the image when it is sourced externally.
6. **Additional classes**:
   - Creates custom CSS classes to modify graphical objects.

Once you have configured the image parameters, click the **Add to plan** button. A new object will appear on the map, which you can move to your desired location. If necessary, resize the image by grabbing its edges.

Remember to adjust the element size and position (refer to Object, page 69).

Finally, click **Save and reload plan** to apply your changes.

Frame

The **Frame** tab allows you to display either internal or external webpages within the visualization. You can integrate **Schedulers** and **Trend logs** into the frame. To configure the **Frame** parameters, follow these steps:

1. **Source**:
   - Choose from **Scheduler**, **Trend log**, or an external **URL**.
2. **Frame size**:
   - Specify the desired width and height for the frame.
3. **Custom name**:
   - Provide a title for the frame.
4. **Refresh interval (seconds)**:
   - Set the rate at which the frame content refreshes (maximum = 3,600 seconds).
5. **Persistent**:
   - By default, frames load when the plan is visible and are removed when hidden (for performance reasons with many frames).
   - Persistent frames load during initialization and remain throughout (useful for apps like alerts).
6. **Hide in Smartphone**:
   - When checked, the frame won’t be available in Smartphone visualization.

After defining the frame parameters, click the **Add to plan** button. A newly created object will appear on the map, which you can move to your desired location. Resize the frame by holding its edges.

Remember to adjust the element size and position (refer to Object, page 69).

Finally, click **Save and reload plan** to activate the frame.

NOTE: Some web pages may have JavaScript preventing frame usage. In such cases, the webpage will open in full screen rather than within the frame. For **Schedulers** or **Trends**, it is recommended to stretch the frame to maximum width (minimum width: 1024). The frame is visible only under PC/ Tablet visualization. Ensure that **Schedulers** or **Trends** are not accessible from Smartphone visualization by adjusting settings in **Vis. structure** under a dedicated plan.

Gauge

The **Gauge** tab enables dynamic visualization and allows you to change the object value displayed in the gauge. Follow these steps to set up the gauge parameters:

1. **Data object**:
   - Specify the KNX group address associated with the gauge.
2. **Gauge size**:
   - Define the desired dimensions for the gauge.
3. **Custom name**:
   - Provide a custom name for the gauge object.
4. **Read-only**:
   - Toggle this option to make the gauge read-only (if needed).
5. **Additional classes**:
   - Create an additional class for custom CSS modifications targeting specific graphical objects.

Once you have defined the gauge parameters, click the **Add to plan** button. A newly created object will appear on the map, which you can move to your desired location.

Adjust the element size and position (refer to Object, page 69).

Click **Save and reload plan** to activate the gauge.

Camera

The controller supports third-party IP web camera integration into its visualization.

Only cameras supporting HTTP MJPEG streaming in the web browser can be visualized.

Configure the camera parameters as follows:

1. **Local source URL**:
   - Use the local source address of the video stream available on the same sub-network.
2. **Remote source URL**:
   - Specify the remote source address of the video stream.
   - The remote camera stream is displayed when the client IP is from a different sub-network than the server.
3. **Window size**:
   - Define the width and height for displaying the video graph.
4. **Custom name**:
   - Assign a name to the camera object.
5. **Auto open window**:
   - Enable this option to automatically open the video window when the plan is accessed.

6. **Additional classes**:
   - Create an additional class for custom CSS modifications targeting specific graphical objects.
7. **Hide background**:
   - Choose whether to hide the icon background.

If your IP camera requires a username and password, format the URL accordingly:

http://USER:PASSWORD@IP

Once you have defined the camera parameters, click the **Add to plan** button. A newly created object will appear on the map, which you can move to your desired location.

Adjust the element size and position (refer to Object, page 69).

Click **Save and reload plan** to activate the camera.

Streaming issues:

Keep in mind that the controller acts as a redirecting stream from the camera to the browser. If the stream doesn't work, it is likely a web browser issue, not related to the controller.

For external camera availability, ensure that the camera's IP is port-forwarded through the router. When adding an external camera, use the correct IP with the port (e.g., IP:port). Using the local IP will not make the camera available externally.

If you encounter issues with direct video streaming, contact the technical support of the camera manufacturer.

Graph

The **Graph** tab allows you to integrate real-time graphs into a visualization system for monitoring scale-type objects current and historical values. Follow these steps to set up the graph parameters:

1. **Data object**:
   - Specify the group address of the object.
   - Ensure that the object has the Log option activated for the graph to function in visualization.
2. **Custom name**:
   - Assign a name to the object.
3. **Icon**:
   - Choose an icon to launch the graph.
4. **Window size**:
   - Define the width and height for displaying the graph.
5. **Number of points**:
   - Determine the maximum number of data points to display in the graph (up to 200 points).
6. **Auto-follow value**:
   - Use this option for objects that never reach zero value (e.g., CO_2 levels). It improves graph resolution.
7. **Auto open window**:
   - The graph window automatically opens when accessed.
8. **Hide background**:
   - Choose whether to hide the icon background.

9. **Additional classes**:
    - Create an additional class for custom CSS modifications targeting specific graphical objects.

Once you have defined the graph parameters, click the **Add to plan** button, and a newly created object will appear. You can move this object to your desired location.

Adjust the element size and position (refer to Object, page 69).

Click **Save and reload plan** to activate the objects (with a delay for obtaining relevant data).

IMPORTANT: Ensure that the object providing data is set as logged in its properties (see Add New Object, page 52).

Launching Visualization on iPhone

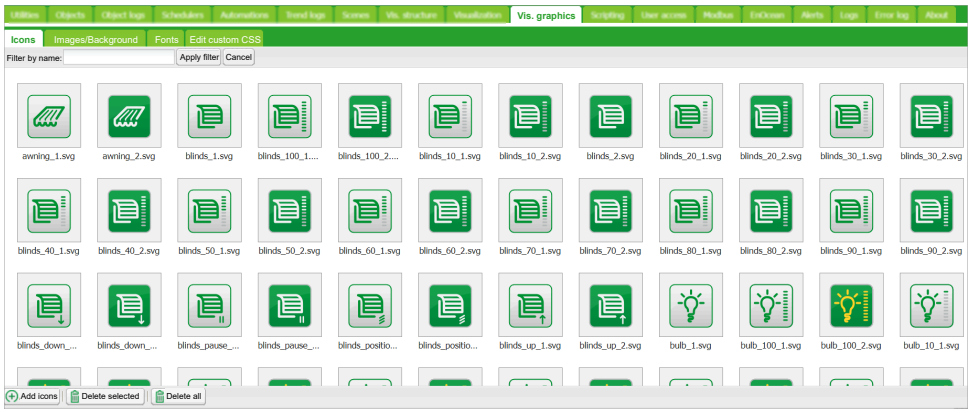
1. Ensure that your iPhone is wirelessly connected to the controller via a separate access point (such as a wireless router).
2. Open your iPhone web browser (Safari).
3. Enter the controller IP address (default: 192.168.0.10) in the browser.
4. Click Smartphone on the **Start page** of your controller.
5. To create a shortcut for easy access, save the application link to your iPhone Home screen: Tap in the Safari browser and tap **Add to Home Screen**.

Launching PC Visualization

1. Ensure that your PC or tablet device can access the controller.
2. Open a web browser and enter the controller IP address (default: 192.168.0.10).
3. Click PC/Tablet on the **Start page**.
4. Select the desired plan.
5. To maximize the map visibility, you can minimize the sidebar by clicking .

Vis. Graphics

This tab is divided into four sections:



1. **Icons**: This section contains all the object icons.

2. **Images/Backgrounds**: Here, you can find locally stored pictures.
3. **Fonts**: All available fonts are listed in this tab.
4. **Edit custom CSS**: Use this option to create or modify custom cascading style sheets (CSS).

Filter by name: You can filter icons or images/backgrounds based on their names.

Adding icons:

1. Click the **Add icon** button to create a new entry.
2. The system accepts icons of any size.
3. Supported formats include JPEG, GIF, PNG, and SVG.
4. You can also upload a ZIP archive containing multiple graphics. Each item within the archive must not exceed 2 MB, and the total archive size cannot exceed 32 MB.
5. **Name (optional)**: Specify a name for the icon. It will appear in the list when adding a new object. The name can contain letters, numbers, underscores, and hyphens.
6. **File**: Provide the location of the icon file.
7. Click **Save**.

Adding images/fonts: You can follow the same process as adding icons to include new images or fonts.

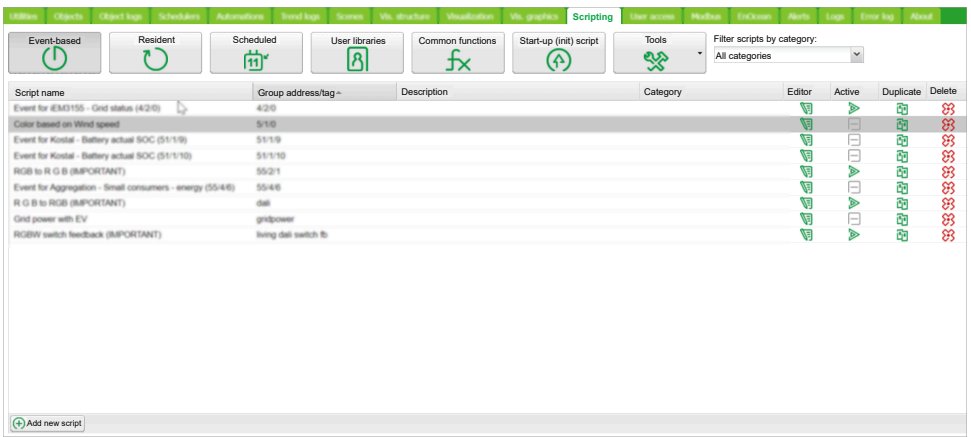
You can also delete icons/images with **Delete selected** or **Delete all** buttons.

Customizing CSS:

- To change the CSS style, upload a new CSS file: Click **Edit custom CSS** → **Upload custom CSS** → provide the location of the CSS file → click **Save**.
- CSS files define the appearance of control buttons, smartphone visualization, scheduler, and trend elements.
- For detailed instructions on modifying the CSS file, please contact your local front office for additional documentation.
- Remember to clear your browser cache after uploading a new CSS file.

Scripting

The **Scripting** tab allows you to add and manage various scripts, depending on their type. These scripts are implemented using the Lua programming language.



Within the **Scripting** tab, you'll find four actions you can perform for each script:

1. **Editor**: Enter the scripting editor to write specific code for a particular program.
2. **Active**: Toggle the script's activation status (green for active, grey for deactivated).
3. **Duplicate**: Create a duplicate of the script along with its source code.
4. **Delete**: When you click this icon, a confirmation prompt appears to accept the deletion.

Data format:

- In most cases, data is stored and transferred between the controller components using hex-encoded strings (2 bytes per 1 byte of data).

Programming in Lua:

- For more information on programming in Lua and to explore code samples, visit www.se.com.

Event-Based Script

Event-based scripts execute when a group event occurs on the bus. They are particularly useful for real-time responses. To create an event-based script, follow these steps:

1. Click **Event-based** button in the **Scripting** tab → click the **Add new script** button.
2. Set the parameters in the **Event-based script** dialog as follows:
   - **Script name**: Provide a descriptive name for the script.
   - **Group address/tag**:
     - Manually enter group addresses (using digits from 0 to 15) or select from the drop-down list.
     - Group address format: Use digits and "/" as a separator (e.g., 1/1/1).
     - If tags are attached to group addresses, the script can run based on those tags.
   - **Active**: Set the script's activation status (green for active, red for disabled).
   - **Execute on group read**: Choose whether to run the script with every group read.
   - **Execution mode**:
     - **Normal**: The event script runs each time it's triggered.
     - **First instance only**: Does not start a new script (PID) if it's already running.
     - **Last instance only**: Stops all running scripts and PIDs, executing only the last one triggered. Commonly used when saving the current PID into storage.
   - **Category**: Assign the script to a new or existing category. This helps organize scripts in the **Print script listings** page under **Tools**.
   - **Description**: Add a brief description of the script.

If the script runs only on a read request, consider using the following example:

```
if event.type == 'groupread' then
-- Your script logic here
end
```

Resident Script

A resident script is a script that runs directly on the controller itself. To set up a resident script, follow these steps:

1. Click the **Resident** button in the **Scripting** tab.
2. Click the **Add new script** button.
3. Configure the parameters in the **Resident script** dialog as follows:
   - **Script name**: Specify a descriptive name for the script.
   - **Sleep interval (seconds)**: Set the interval after which the script will execute.
   - **Active**: Choose whether the script is active (green circle) or disabled (red circle).
   - **Category**: Assign a new or existing category name to group similar scripts (this won't affect the script's action but helps with organization).
   - **Description**: Provide a brief description of the script.

Scheduled Script

A scheduled script is a script that runs based on predefined schedules or time events. To set up a scheduled script, follow these steps:

1. Click the **Scheduled** button in the **Scripting** tab.
2. Click the **Add new script** button.
3. Configure the parameters in the **Scheduled script** dialog as follows:
   - **Script name**: Specify a descriptive name for the script.
   - **Minute/Hour/Day of the month**: Scheduled scripting uses standard cron format for date/time parameters.

*	Execute script every minute, hour, or day.
*/N	Execute script every N minutes/hours/days. N is an integer. A script is executed when current value divided by N gives 0 in modulo. For example, a script with hour parameter set to */8 will be executed when the hour value is 0, 8, and 16.
N	Execute script when minute/hour/day matches N.
N-K	Execute script when minute/hour/day is between N-K range (inclusive).
N,K	You can specify multiple N and N-K type parameters separated by commas. For example, a script with the minute parameter set to 15,50–52 will execute when the minute value is 15, 50, 51, and 52.

   - **Month of the year/Day of the week**: Select from the dropdown menu.
   - **Active**: Choose whether the script is active (green circle) or disabled (red circle).
   - **Category**: Assign a new or existing category name to group similar scripts (this won't affect the script's action but helps with organization).
   - **Description**: Provide a brief description of the script.

User Libraries

User libraries contain user-defined functions that can be called from other scripts.

Keep source:

- There is an option to keep the source code available for user libraries. When disabled, the code is compiled into binary form and cannot be further edited. If this option is enabled, the source code remains visible in the editor.

The **Auto load library** option loads the selected script when the controller starts.

Including the library in scripts

To use functions defined in a user library, include them at the beginning of your script. For example, if you have a user library named ‘test’, include it as follows:

```
require('user.test')
```

User libraries backup and restoration

User libraries can be backed up and restored or added from an archive. Keep in mind that importing a new library will replace the existing one.

Common Functions

Common functions comprise a library of globally used functions. These functions can be called from any script at any time, without requiring special inclusions with user libraries. By default, **Common functions** include features such as sunrise/sunset calculations and email functionality.

Start-Up (Init) Script

The initialization script (init script) is used to set up specific system or bus values when the system starts. This script runs each time after the system has restarted, whether due to power-up, software reboot, or via a hardware reset push button.

Tools

The **Tools** menu includes the following options:

1. **Backup scripts**: Back up all scripts in a *.gz file.

   NOTE: Script backup does not include user libraries; those must be backed up separately.
2. **Restore scripts**: Restore a script from an archive (*.gz) file with two possibilities:
   - Remove existing scripts and import from backup.
   - Append, keeping existing scripts.
3. **Print script listings**: Shows all scripts with codes in a list format, sorted by categories.
4. **Edit custom JavaScript**: Insert JavaScript code for script control.
5. **Show logs window**: All log data is listed here; it is a duplicated window from **Configuration** → **Logs**. It allows debugging a script while simultaneously checking the logged data.

Example: Sample code for 1-Byte Object 1/0/0 controlling navigation between pages

```
$(function(){
  /* Create an event listener on 1/0/0 to jump to the page
with the object value */
  const addr = Scada.encodeGroupAddress('1/0/0');
  objectStore.addListener(addr, function(obj, type) {
     /* To avoid execution on the opening page */
     if (type === 'init') {
       return;
     }
     /* Jump to the page with the object value */
     if (currentPlanId !== obj.value) {
       showPlan(obj.value);
```

```
        /* Write the object back to 0 */
        setObjectValue({ address: '1/0/0', rawdatatype: 5 },
0, 'text');
      }
    });
 });
```

Script Editor

When you add a script, an icon appears in the **Editor** column, allowing you to open the script in the script editor. Here are some key features:

1. **Helpers**: Save time by using built-in code snippets. Click the appropriate snippet, and it will automatically add code to the editor field.
2. Keyboard shortcuts:
   - **Ctrl + F**: Find syntax in your code (text will be highlighted in yellow).
   - **Ctrl + G**: After finding text via Ctrl + F, use Ctrl + G to select the next syntax in the script.
   - **Shift + Ctrl + G**: Select the previous syntax.
   - **Shift + Ctrl + F**: Replace syntax in a script with another one (choose one by one if needed).
   - **Shift + Ctrl + R**: Replace all syntaxes in a script at once.
   - **Ctrl + Space**: Auto-detect code and suggest commands. Write the first letter of a command, then select the correct one from the list.
3. Main features in the script editor:
   - **Helpers**: Predefined code snippets (e.g., if-then statements).
     - **Loops and iterators**: Includes Array, Repeat…Until, and more.
     - **Math**: Random value, Ceiling, Absolute value, Round, etc.
     - **Objects/KNX bus**: Get object value, Group read, Group write, Update interval, etc.
     - **Storage**: Retrieve and save data.
     - **Script control**: Manage other scripts (enable/disable).
     - **Alerts and logs**: Create alerts, log variables, and formatted alerts.
     - **Time functions**: Delay script execution.
     - **Miscellaneous**: Sunrise/sunset, etc.
     - **Serial**: Communication through internal controller I/O ports.
     - **Modbus**: Create RTU/TCP connections, read/write registers.
   - **Group addresses**: Existing group addresses on the KNX bus.
   - **Objects by name**: Choose objects by name.
   - **Tags**: Select objects by tag.
   - **Data types**: Choose objects by data type.
   - **Scripts**: Lists of event-based, resident, scheduled, and startup (init) scripts.

User Access

The user access menu enables the creation and management of user accounts.

Here are some key points:

1. User accounts: You can create and manage user accounts through this menu. It allows you to define roles, permissions, and access levels for each user.

2. Best practice: While designing your user access system, consider the following best practices:
   - **Defining user roles**: Clearly define roles and responsibilities for users. This forms the foundation of a robust security framework.
   - **Strong authentication**: Implement password policies.
   - **Least privilege principle**: Grant users the minimum necessary permissions to perform their tasks.
   - **Regular access reviews**: Periodically review users' access rights to ensure they align with their roles.
3. User count recommendation: For optimal performance, it is advisable to keep the number of users below 20.

Add New User

To add a new user, follow these steps:

1. Open the **User access** tab.
2. Click the **Add new user** button located at the bottom left corner.
3. Update the user access settings as needed.

Here are the details for each field in the **General** section of the **User** dialog:

- **Name**: Enter the name of the account.
- **Login**: Use a login name with a length between 2 and 20 characters. Accepted characters include "-", "_", lowercase letters (a-z), and digits (0-9).
- **Active**: Activate or deactivate the user account.
- **Password**: Set a password with a length at least 8 characters. Password must contain at least one uppercase letter, lowercase letter, and a digit..

Next, configure access levels for the user:

- **Visualization/Schedulers/Trends access**:
  - Choose from **None**, **Partial**, or **Full** access. If you select partial access, the **Visualization**, **Schedulers**, and **Trend logs** tabs become active, allowing you to specify parameters.
- **Touch access**:
  - Select either **Partial** or **Full** access. If you select partial access, the **Touch** tab becomes active, allowing you to specify which rooms are visible in the **Touch** visualization. Check the rooms the user is allowed to access..
- **Homepage**: Set the page the user sees after logging in (e.g., default **Start page** or a specific application).
- **Applications access**: If you choose partial access in the **General** tab, the **Applications** tab becomes active. Select the applications the user can access from the **Start page**.
- **Advanced**: Provide a list of group addresses available to users.

User Access Settings

To configure user access settings, follow these steps:

1. Click the **User access settings** button in the **User access** tab.

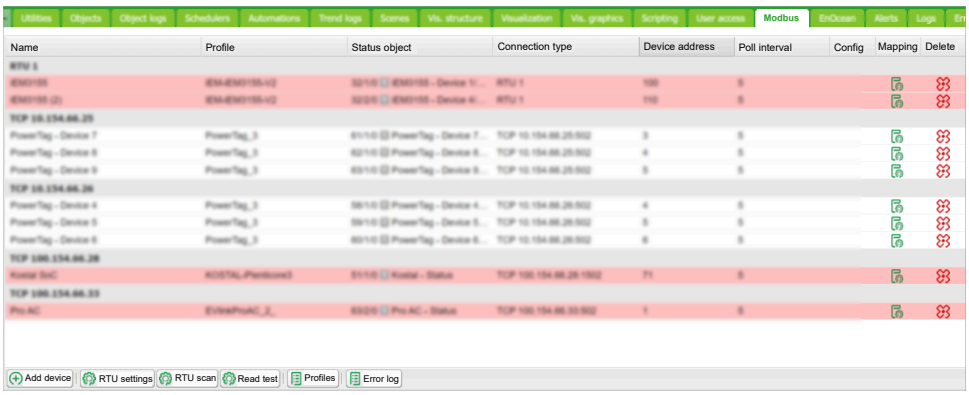
2. Adjust the parameters in the **User access settings** dialog as needed:
   - **Disable password for Visualization**:
     - Disabling active password protection allows direct visualization access. Consider using a PIN for security.
   - **Enable password for Applications**:
     - Require a password when entering any application on the main page.
   - **Enable password for user directory**:
     - The **User directory** (located at **/data/apps/store/user**) is a folder within the controller. You can upload files to this directory via FTP (using the username “apps”). These uploaded files become accessible via an HTTP endpoint: **http://{CONTROLLER_IP}/user/{FILE NAME}**. Files in the User directory can be of various types, including HTML, TXT, LP (Lua) files.
     - When accessed via a browser, these files are interpreted as follows: HTML and TXT files are displayed in text form. However, *.lp (Lua) files are executed, and their standard output is returned to the browser.
     - The **Enable password for User directory** parameter allows you to require a password for accessing *.lp files via the HTTP endpoint. Files with different extensions remain accessible without a password.
   - **Visualization PIN code**:
     - If active password protection is disabled, protect visualizations with a 6-digit PIN code.
   - **Remember login and password**:
     - Store user credentials for convenience.
   - **User cookie expiration days**:
     - Set the duration after which web browser cookies will be deleted.

Access Logs

Click the **Access logs** button in the **User access** tab to view login history.

Unsuccessful logins are marked in red, and FTP and applications access are also logged.

Modbus



The Modbus open standard provides a comprehensive analysis of consumption across various areas of your building. By connecting up to 31 Modbus extension devices within a single Modbus line, you can monitor different types of meters. These include:

- Schneider Electric energy meters

- Schneider Electric power meters
- Schneider Electric Smart Interface Modules (SIM10M module)
- Schneider Electric Smartlink
- Other Schneider Electric Modbus devices (e.g., SE8000, Modicon PLCs)
- Non-Schneider Electric Modbus TCP/RTU devices (offering greater flexibility)

With the information provided by your controller, you can visualize energy or media consumption and implement control strategies within the KNX/IP network to reduce consumption.

Modbus communication details:

- Modbus RTU is supported over the RS485 interface.
- Modbus TCP is supported over the Ethernet port.
- Configure Modbus communication settings using the Modbus tab in the **Configurator** of your controller.
- Easily map Modbus registers using predefined Modbus profiles.

Script-based Modbus control:

- Modbus Master can be controlled directly from scripts. Typically, a resident script reads Modbus values at specific intervals and writes them into KNX objects or visualizations.
- Add your script in the **Script editor**, utilizing predefined code blocks from the **Helpers**.

IMPORTANT: Avoid using Modbus settings via profiles alongside Modbus controlled from scripts. Interference between these settings can lead to communication errors. We recommend prioritizing Modbus device profiles over configuration by scripting.

Application Example

Requirements:

1. Measure and visualize the energy used for lighting in an office building.
2. Monitor gas and water consumption within the building.
3. Ensure operational safety of IT equipment by monitoring network quality.

Solution:

1. Install an iEM3150 meter to measure energy consumption by the lights.
2. Use an iEM3255 meter to assess power mains quality.
3. Employ a SIM10M module with pulse meters to measure gas and water consumption.
4. Connect all devices via Modbus for seamless communication.

Modbus RTU Interface

The Modbus RTU interface offers the following features:

- Supported over RS-485 physical interface.
- Can function as either a Modbus/RTU Master or a Modbus/RTU Extension.
- Supported Function Codes: #01, #02, #03, #04, #05, #06, #07, #0F, #10.
- Maximum of 32 devices on the bus (1 master and 31 extensions).
- RS-485 interface is not isolated.

Main RS-485 characteristics

- Mode of operation: Differential.
- Voltage at any bus terminal: -7 V to +12 V.
- Receiver Input Sensitivity: +/-200 mV.
- Sink/Source current: 60 mA.
- Built-in asymmetrical protection against transient voltages (ESD, EFT, and lightning).
- Non-isolated RS-485 interface.

Connection Types:	• Point-to-point connections. • Point-to-multipoint connections.
Trunk Cable Type:	Shielded cable with 1 twisted pair and at least a third conductor.
Maximum Bus Length:	1,000 m (3,280 ft) at 19,200 bit/s with the Telemecanique TSX CSA cable.
Maximum Devices (without repeater):	32 devices (1 master and 31 extensions).
Maximum Tap Link Length:	• 20 m (65 ft) for one tap link. • Total of 40 m (131 ft) for all tap links on the bus.

Common ground wires

To maintain voltage within the allowed range (-7 V to +12 V) between drivers and receivers:

- An additional third wire (in 2-wire systems) serves as a common circuit.
- Connect this wire directly to protective ground at one point for the entire bus.
- Choose the master device or its tap as the grounding point.

NOTE:

- No terminal for cable shield.
- In harsh environments, consider placing an additional shield clamp near the controller to drain EMC disturbances.
- Earthed connectors from USB, RS232, LAN, and Modbus should be interconnected cautiously to avoid earth leakage currents affecting controller operation.

Maximum devices without repeater

An RS-485 network can handle up to 32 devices. If you need more, integrate a repeater.

Biasing the network

When there is no data activity on the Modbus bus – meaning all nodes are in receive mode and no active driver is available – the state of the line becomes uncertain. In such cases, external noise or interference can affect the line. To prevent receivers from adopting improper states, it is essential to bias the line. This involves maintaining a constant state by using an external pair of resistors connected to the RS-485 balanced pair.

RC termination

- Use RC termination to minimize the loop current and the line reflections.
- Choose 1 nF (10 V minimum) serial capacitors and 120 Ω (0.25 W) resistors as line termination.
- Integrate these components at both ends of your Modbus SL communication line.

R termination only

If R = 150 Ohm termination is preferred (not RC):

- Connect external polarization resistors (450 – 650 Ohm) at the master tap.

Modbus interface isolation

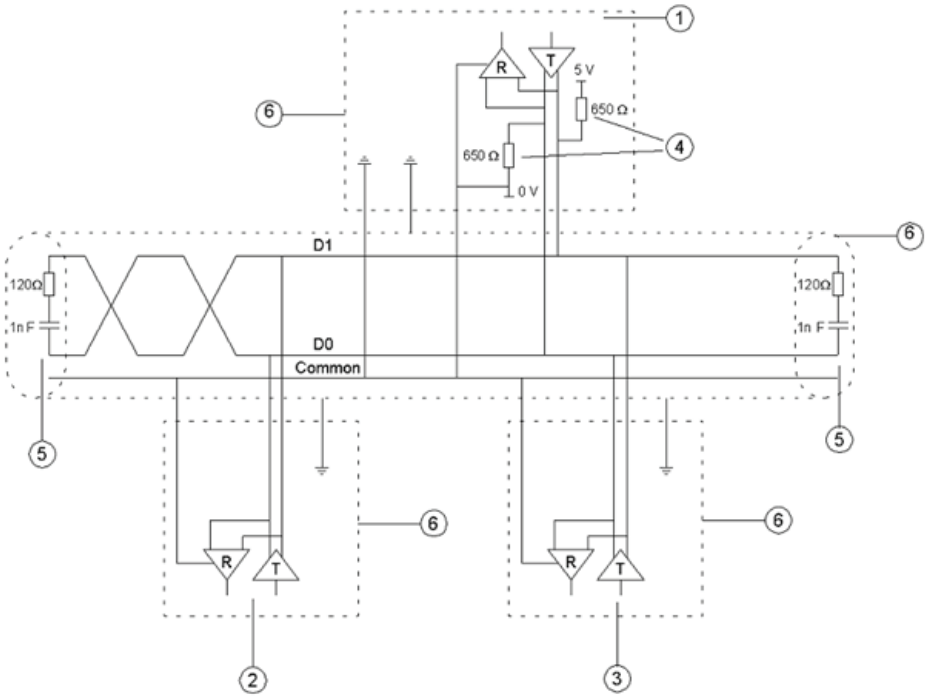
The controller Modbus interface is not isolated. Consider using optic-coupler separation for increased robustness and reduced noise in the Modbus network.

Termination, polarization, and separation using Schneider Electric devices

Schneider Electric provides devices for RS-485 termination, polarization, and separation. You can refer to the following product numbers:

1. TWD XCA ISO
2. TWD XCA T3J

These devices are designed to ensure the recommended RS-485 connection scheme (see the picture below). For more detailed information about TWD XCA ISO and TWD XCA T3J, please consult the product documentation on the Schneider Electric website.



Legend:

1. Master
2. Extension 1
3. Extension n
4. Polarization resistors
5. Line termination
6. Shield

Maximum cable length without termination

When a cable is not terminated, you need to adjust the transmission rate based on the cable length. Here are the relevant details:

Transmission rate:	9600	19,200	57,600	115,200
Bit time (µs):	104.1	52.08	17.36	8.68

Bit time/4 (µs):	26.0	13.0	4.3	2.2
Max. cable length without termination (meters):	859	430	143	72

Modbus TCP Interface

- Supported over Ethernet interfaces (10Mb, 100Mb)
- Can function as a Modbus/TCP-IP Client or Modbus/TCP-IP Server
- Supports DHCP
- Maximum opened TCP connections: 100
- Supported Function Codes: #01, #02, #03, #04, #05, #06, #07, #0F, #10

Grounding-isolation

The metal cover of the RJ45 socket is connected to the device ground.

Modbus Settings in the Controller Using Device Profiles

All Modbus communication settings for the controller are accessible through the **Modbus** tab. Device profiles play a crucial role in mapping Modbus addresses (registers) to KNX group objects. If you need to read or write a Modbus register, simply configure the mapping rules, allowing you to access the register via KNX group objects.

Here is a step-by-step guide for Modbus communication settings:

1. **Modbus RTU configuration:**
   - Specify the details for Modbus RTU communication, such as baud rate and parity.
   - This step applies if you're using Modbus RTU.
2. **Device profile upload:**
   - Ensure that a device profile is uploaded in the controller.
   - Preinstalled profiles for Schneider-Electric devices are available, and you can also upload custom Modbus profiles.
3. **Add the device:**
   - Add your Modbus device to the device list.
4. **Register mapping:**
   - Configure the mapping between Modbus registers and KNX group objects.
5. **Automatic discovery:**
   - To find Modbus devices connected to the controller via Modbus RTU, use the scan function.
   - Access it here: **Configurator → Modbus → RTU scan**.
   - Devices marked as **RTU scan = No** do not support automatic discovery.

Modbus device	RTU scan	Modbus device	RTU scan	Modbus device	RTU scan
Compact_NSX-Compact_NSX_E	No	PM-PM820	No	EM-EM6400	-
Masterpact_NT_NW-Masterpact_A	No	PM-PM850	No	EM-EM6433	-
Masterpact_NT_NW-Masterpact_H	No	PM-PM870	No	EM-EM6434	-
Masterpact_NT_NW-Masterpact_P	No	PM-PM9C	No	EM-EM6436	-

Modbus device	RTU scan	Modbus device	RTU scan	Modbus device	RTU scan
PM-PM1200	No	SIM10M	No	EM-EM6436 and EM-EM6436dual	-
PM-PM210	No	Smartlink-RTU	Yes	EM-EM6438	-
PM-PM3250	Yes	Smartlink-TCP	No	ION-ION6200	-
PM-PM3255	Yes	SE8300	No	ION-ION7300	-
PM-PM5110	No	SE8600	No	ION-ION7330	-
PM-PM5111	No	SER8300	No	ION-ION7350	-
PM-PM5310	No	TC303	No	ION-ION7550	-
PM-PM5330	No	iEM-iEM2150	Yes	ION-ION7650	-
PM-PM5350	No	iEM-iEM2155	Yes	Masterpact_NT_NW-Masterpact_E	-
iEM-iEM3150	Yes	Vigilohm IM20	Yes	PM-PM500	-
iEM-iEM3155	Yes	Vigilohm IM400	Yes	PM-PM5111	-
iEM-iEM3250	Yes	PowerTag	No	PM-PM5320	-
iEM-iEM3255	Yes	CM-CM3250	-	PM-PM5331	-
iEM-iEM3350	Yes	CM-CM3350	-	PM-PM5340	-
iEM-iEM3355	Yes	CM-CM4000	-	PM-PM5341	-
PM-PM710	No	Compact_NSX-Compact_NSX_A	-	PM-PM5560	-
PM-PM750	No	EM-EM3550	-	PM-PM5561	-
PM-PM810	No	EM-EM3555	-	PM-PM5563	-

Adding a New Modbus Device

To add a new Modbus device to your controller configuration, follow these steps:

1. Click the **Add device** button in the **Modbus** tab.

2. Configure the parameters in the **Modbus device** dialog as follows:
   - **Connection type**:
     - Choose either **RTU (RS-485)** or **TCP/IP** as the connection type for your Modbus device.
   - **Name**:
     - Define a name for the Modbus device.
   - **Status object**:
     - Monitor the connection status with the Modbus device using the **Status object**.
     - When communication is disrupted, the status object is set to 0.
   - **Write to bus**:
     - Enable sending data to the KNX TP bus.
     - If disabled, telegrams to KNX TP are not sent when the Modbus value changes. Telegrams are still sent to KNX IP.
   - **Profile**:
     - Select the Modbus profile for your device.
     - You can use a preinstalled profile or upload a custom device profile to the controller.
     - Refer to the procedure for creating a new profile definition (Creating a New Modbus Device Profile, page 91).
   - **Device address**:
     - Set the extension address of your Modbus device.
   - **Poll interval (seconds)**:
     - Specify how often values are polled from the Modbus extension device.
     - A value of 5 means new values are read every 5 seconds.
   - **Timeout (seconds)**:
     - If there is an error in the connection, the device will wait for the specified time and then send an error message to the **Error log**.
   - **IP** (Modbus TCP):
     - If you are using Modbus TCP, set the IP address of the Modbus device.
     - If the Modbus device is connected via a Modbus gateway, set the gateway IP address here.
   - **Port** (Modbus TCP):
     - Set the port for Modbus TCP communication.
     - The default value according to the Modbus standard is 502.
   - **Persistent connection**:
     - When enabled, the connection remains open.
     - When disabled, the controller reads all registers and disconnects.

     IMPORTANT: Some **Modbus TCP/IP devices** may not handle frequent (re)connections well (e.g., panel servers, inverters, Modbus meters). Consider using the **Persistent connection** option if your polling interval is less than a minute.

RTU Settings

To communicate with Modbus extensions connected via Modbus RTU (serial), follow these steps:

1. Click the **RTU settings** button in the **Modbus** tab.

2. Configure the parameters in the **RTU settings** tab as follows:
   - **RTU (serial) enabled**:
     - Enable Modbus RTU communication.

       NOTE: Only **RTU1** can be set for Modbus communication.
   - **Port**:
     - Set the name of the serial port.
     - The default setting is **/dev/RS485**.
   - **Parity**:
     - Specify the parity or stop bits.
   - **Duplex**:
     - Choose either **Half-duplex** or **Full-duplex**.
     - The default value is **Half-duplex**.
   - **Reset to defaults**:
     - Click this button to reset all RTU settings parameters to their default values.

RTU Scan

The **RTU scan** feature automatically detects and adds Modbus devices connected to Modbus RTU.

To initiate the scan, click the **RTU scan** button in the **Modbus** tab.

Only devices with valid profiles will be added. This feature does not function with custom-made profiles.

For a list of devices supporting automatic discovery, refer to Profiles, page 90.

Read Test

This feature allows for a quick test of RTU communication. To initiate the RTU read test, follow these steps:

1. Click the **Read test** button in the **Modbus** tab.
2. Set up the parameters in the **Read test** dialog as follows:
   - **Device address**: Specify the address of the Modbus device.
   - **Function**: Choose between **Coil**, **Discrete input**, **Holding register**, and **Input register** functions.
   - **Address**: Enter the register address.
   - **Data type**: Select the appropriate message data point type.
   - **Read swap**: Adjust the order of reading if needed.
   - **Read length**: Define the length of the read message.

Profiles

All uploaded Modbus profiles are displayed in a table that appears when you click the **Profiles** button in the **Modbus** tab. Here's what you can do with these profiles:

1. Delete profiles:
   - Click  associated with each profile to delete it.

2. Download profiles:
   - Click to save a profile as a .json file.
   - You can then use this downloaded profile for further customization.
3. Modify profiles:
   - If you need to read/write registers not available in the preinstalled profile, consider modifying it according to your specific requirements.
   - Upload the modified profile back to the controller the **Add profile** button at the bottom left of the **Profiles** table.

Error Logs

Any errors related to Modbus are displayed in the **Error log** window under the **Modbus** tab.

Modbus Mapping

After adding the Modbus device as described in the RTU Scan, page 90, the next step is to set up the register mapping. This involves creating bindings between Modbus registers and KNX group objects in the controller.

Follow these steps:

1. Navigate to the **Modbus** tab in the **Configurator**.
2. Each device in the list has a green mapping icon on the right side . Click it to open the **Object mapping** settings.
3. The **Object mapping** table represents each Modbus register (defined in the device profile).
4. Double-click a selected line or click the **Map selected items to objects** button to open the **Mapping** dialog.
5. Configure the following settings:
   - **Link to object**: Select the controller object where the value read from the Modbus register will be saved. You can choose an existing object from the dropdown menu or create a new one directly from this dialog.
   - **Write to bus**: Enable sending data to the KNX TP bus. If disabled, telegrams to KNX TP are not sent when the Modbus value changes. Telegrams are still sent to KNX IP. Properly configure the filtering table if you don't want to share values through KNX IP.
   - **Value send delta**: Set the value of the delta. If the change in value read from the Modbus register exceeds this delta, the value is sent to the KNX bus. For KNX TP, ensure that the **Write to bus** option is enabled.
   - **Unit/Suffix**: Optionally set the unit of the Modbus value. This setting applies to the selected group object in the controller.
   - **Tags**: Optionally select a tag to apply to the group object in the controller.
   - **Comments**: Optionally select a comment to apply to the group object in the controller.

NOTE: A Modbus device is highlighted in red after three unsuccessful polls.

Creating a New Modbus Device Profile

If your Modbus device profile is not included in the preinstalled profiles in the controller, you can define your own custom profile.

Follow these steps:

1. File format: Modbus device profiles are distributed as *.json files. You can create and edit these files using a common text editor (such as Notepad or Notepad++). Remember to save the file with the extension *.json.

2. Profile structure example:

```
{
   "manufacturer": "Schneider Electric",
   "description": "Example device",
   "mapping": [
     {
           "name": "Output 1",
           "bus_datatype": "bool",
           "type": "coil",
           "address": 0,
           "writable": 1
     },
     {
           "name": "Input 1",
           "bus_datatype": "float16",
           "type": "inputregister",
           "address": 0,
           "value_multiplier": 0.001,
           "units": "V"
     }
   ]
}
```

- The `"mapping"` table in the *.json file contains information for each Modbus register, coil, input, or output.
- Adjust the mapping settings according to your specific device:

Parameter	Description	Type	Required
Name	Object name, e.g. Output 2	String	Yes
bus_datatype	KNX object data type, key **from dt** table, e.g. float32	String/Number	Yes
type	Modbus register type, possible values: coil, discreteinput, register, inputregister.	String	Yes
address	Register address (0-based)	Number	Yes
writable	Set to true to enable writing to register if type is either coil or register.	Boolean	No
write_only	Set to true to disable reading coil or register value when *"writable"* is enabled.	Boolean	No
datatype	Modbus value data type. If set, conversion will be done automatically. Possible values: bool, uint16, int16, float16, uint32, int32, float32, uint64, int64, quad10k, s10k	String	No
value_delta	New value is sent when the difference between previously sent value and current value is larger than delta. Defaults to 0 (send after each read).	Number	No
value_base	Add specified number to the resulting value.	Number	No
value_multiplier	Multiply resulting value by the specified number,value = value_base + value * value_multiplier.	Number	No
value_bitmask	Bit mask to apply, shifting is done automatically based on least significant 1 found in the mask.	Number	No
value_nan	Array of 16-bit integers. If specified and read operation returns the same array no further processing of value is done.	Array	No
value_conv	Apply one of built-in conversion functions	String (Int)	No
value_custom	Name of a built-in enumeration or a list of key -> value mapping, resulting value will be 0 if key is not found.	String/Object	No
internal	Not visible to user when set to true, should be used for scale registers.	Boolean	No
units	KNX object units/suffix	String	No
address_scale	Address of register containing value scale, value = value * 10 ^ scale	Number	No
read_count	Number of register to read at once (for devices that only support reading of a specific block of registers)	Number	No
read_swap	Swap register order during conversion (endianness)	Boolean	No

Parameter	Description	Type	Required
read_offset	Position of first register of data from the block of registers (0-based).	Number	No
timeout	Specify device timeout in seconds. If the extension device does not reply within specified time, it is considered as timeout error. Default values: 0.5 s for Modbus RTU, 3 s for Modbus TCP	Number	No
write_multiple	This parameter set the multiple writing function (function 15 or 16 is used instead of function 5 or 6). If "*Type"* is set to *"register"* and *"Write_multiple"* is set to *"true"*, Modbus function 16 is used for writing to the register. If "*Type"* is set to *"coil"* and *"Write_multiple"* is set to *"true"*, Modbus function 15 is used for writing to the coil. Default value is *"false"*, which means that Modbus function 5 or 6 (depending on register type) is used for writing.	String	No

3. Upload to the controller:
   - Once you have created your *.json file with all the necessary profile information, upload it to the controller via **Configurator** → **Modbus** → **Profiles** → **Add profile**.
4. Best practice:
   - When creating a new device profile, consider using an existing profile as an example or template. You can download existing profiles from the controller to understand the structure and syntax.

For detailed instructions on custom device profile creation, refer to the application note available at Schneider Electric website.

Modbus Settings in the Controller Using Scripts

When working with Modbus in the controller, it is essential to understand the available function codes. These codes determine the type of memory (such as holding registers, input coils, etc.) to access and the corresponding action (read or write). All the functions described below can be used both for Modbus TCP and Modbus RTU.

1. Function Codes:
   - Function codes define specific operations:
     - Read: Retrieve data from extension devices.
     - Write: Update coils or registers.
   - Common function codes include:
     - Read Discrete Inputs (Function Code 2)

Name	"Read discrete input"
Command	value = mb:readdiscreteinputs(address)
Arguments	[address]: address of the input
Returned values	1: ON, 0: OFF
Exception codes	01 or 02 or 03 or 04.
Name	"Read discrete inputs"
Command	value = mb:readdiscreteinputs(start,count)
Arguments	[address]: address of first input to read [count]: number of inputs to read (max 2000)
Returned values	1: ON, 0: OFF
Exception codes:	01 or 02 or 03 or 04
Example	bool1, bool2= mb: readdiscreteinputs(10,2) Value read from discrete input address 11 is returned into variable bool1. Value read from discrete input address 12 is returned into variable bool2.

     - Read Coils (Function Code 1)

Name	"Read single coil"
Command	coil = mb:readcoils(address)
Command	coil= mb:readcoils(start, count)
Arguments	[address]: address of the coils
Returned values	1: ON, 0: OFF
Exception codes	01 or 02 or 03 or 04
Name	"Read Multiple coil"
Arguments	[start]: address of first coil to read [count]: number of coils to read (max 2000)
Returned values	1: ON, 0: OFF
Exception codes	01 or 02 or 03 or 04
Example	coil1,coil2,coil3= mb:readcoils(1000, 3) Value read from coil address 1000 is returned into variable coil1. Value read from coil address 1001 is returned into variable coil2. Value read from coil address 1002 is returned into variable coil3.

     - Read Input Registers (Function Code 4)

Name	"Read input registers"
Command	value = mb:readinputregisters(address,count)
Arguments	[address]: address of first input register to read [count]: number of input registers to read (max 125)

Returned values	2byte values
Exception codes	01 or 02 or 03 or 04.
Example	value1, value2, value3, value4 = mb:readinputregisters(1015,4) Value read from input register address 1015 is returned into variable value1. Value read from input register address 1016 is returned into variable value2. Value read from input register address 1017 is returned into variable value3. Value read from input register address 1018 is returned into variable value4.

- Read Holding Registers (Function Code 3)

Name	"Read registers"
Command	value = mb:readregisters(address,count)
Arguments	[address]: address of first register to read [count]: number of registers to read (max 125)
Returned values	2byte values
Exception codes	01 or 02 or 03 or 04.
Example	int1, int2= mb: readregisters(1100,3) Value read from register address 1100 is returned into variable int1. Value read from register address 1101 is returned into variable int2. Value read from register address 1102 is returned into variable int3.

- Write Single Coil (Function Code 5)

Name	"Write single bit"
Command	value = mb:writebits (1000, value) [starting address, value "true" or "false"/"0"]

- Write Single Register (Function Code 6)

Name	"Write single register"
Command	value = mb:writeregisters(1000, 123) [address, value]

- Write Multiple Coils (Function code 0F)

Name	"Write multiple bits"
Command	value = mb:writebits(1000, true, false, true, ...) [address, bit value1, bit value2,..{max 1968 bits}]

- Write Multiple Registers (Function code 10)

Name	"Write multiple registers"
Command	value = mb:writeregisters(1000, 123, 321, 222, ...) [address, value1, value2, ..{max 123 registers}]
Exception codes	mb:readcoils(start, count) mb:readdiscreteinputs(start, count) mb:readregisters(start, count) mb:readinputregisters(start, count)These commands read one or more registers/coils from the start address and return all values in case of success. In case of error, three variables are sent back: Nil Exception code description Exception code

Modbus exception codes: (see here).

Modbus RTU Configuration Commands

To configure Modbus RTU communication in the controller, follow these steps:

1. Create a Modbus RTU object:
   ```
   require('luamodbus')
   mb = luamodbus.rtu()
   ```
2. Open Modbus RTU connection:
   - Set the communication parameters (baud rate, parity, data bits, stop bits, and duplex mode).
   - Example (19200 baud rate, even parity, 8 data bits, 1 stop bit, half duplex):
     ```
     mb:open('/dev/RS485', 19200, 'E', 8, 1, 'H')
     mb:connect()
     ```
3. Terminal name:
   - The terminal name is `/dev/RS485`.
4. Supported baud rates:
   - Choose from the following baud rates:
     - 300 bit/s
     - 600 bit/s
     - 1200 bit/s
     - 2400 bit/s
     - 4800 bit/s
     - 9600 bit/s
     - 19200 bit/s
     - 38400 bit/s
     - 57600 bit/s
     - 115200 bit/s
     - 230400 bit/s
5. Parity:
   - Set the parity mode:
     - “N” (None)
     - “E” (Even)
     - “O” (Odd)

6. Data bits and stop bits:
   - Data bits: Choose from 5, 6, 7, or 8.
   - Stop bits: Choose either 1 or 2.
   - Duplex mode:
     - “H” (Half duplex)
     - “F” (Full duplex, not supported in RS-485)
7. Baud rate considerations:
   - The baud rate affects communication distance:
     - 9600 bit/sec: Max distance for 1-15 Modbus RTU devices is 1200 meters.
     - 19200 bit/sec: Max distance is 900 meters (typical with Belden 3105A cables).

Baudrate setting	Maximum communication distance for 1 to 15 Modbus RTU devices (Typical with Belden 3105A cables)
9600 bit/sec	1200 m
19200 bit/sec	900 m

8. Parity explanation:
   - Parity checks for successful transmission.
   - Odd parity: Odd number of 1s.
   - Even parity: Even number of 1s.
   - Both gateway and meter must match (odd, even, or none).
9. Delay between frames:
   - Some devices need extra time after a response.
   - Use the delay command (e.g., `os.sleep(1.5)`).
10. Set extension address:
    - Set slave address (e.g., `mb:setslave(123)`).
11. Read registers:
    - Read from address 1000 and store the value:
      ```
      value = mb:readregisters(1000)
      ```
12. Close Modbus connection:
    ```
    mb:close()
    ```
13. Timeout intervals:
    - Adjust timeout intervals as needed:
      - Byte timeout:
        ```
        mb:getbytetimeout()
        mb:setbytetimeout(timeout)
        ```
      - Response timeout:
        ```
        mb:getresponsetimeout()
        mb:setresponsetimeout(timeout)
        ```
      - Receive timeout (extension mode only):
        ```
        mb:getreceivetimeout()
        mb:setreceivetimeout(timeout)
        ```

14. Example initialization:
    - Initialize Modbus on first script execution:

```
if not mb then
     require('luamodbus')
     mb = luamodbus.rtu()
     mb:open('/dev/RS485', 38400, 'E', 8, 1, 'H')
     mb:connect()
 end
 mb:setslave(30)
 mb:flush()
```

Modbus TCP Configuration Commands

1. Create Modbus TCP object:

```
local luamodbus = require('luamodbus')
local mb = luamodbus.tcp()
```

2. Open Modbus TCP connection:

```
mb:open('192.168.1.2', 1234)
mb:connect()
```

The remaining commands for configuring the Modbus TCP connection are the same as for Modbus RTU (Modbus RTU Configuration Commands, page 97).

Modbus Master Functions

Functions listed below can be used for Modbus RTU master or Modbus TCP client.

`mb:setslave(slaveid)`

sets extension id to read/write data from/to

`mb:readcoils(start, count) [01]`

start – address of first coil to read

count – number of coils to read

`mb:readdiscreteinputs(start, count) [02]`

start – address of first discrete input to read

count – number of discrete inputs to read

`mb:readregisters(start, count) [03]`

start – address of first holding register to read

count – number of holding registers to read

`mb:readinputregisters(start, count) [04]`

start – address of input register to read

count – number of input registers to read

returns all values on success and nil, error description on error

`mb:writebits(start, v1, [v2, [v3, ...]]) [05]`

writes values to coils from start address

```
mb:writeregisters(start, v1, [v2, [v3, ...]]) [06]
```

writes values to registers/coils from the start address

single write will be used when only one value is supplied, multiple write otherwise

returns all of values written on success and nil, error description on error

```
mb:reportslaveid()
```

reads extension internal data

returns values on success

returns nil, error description on error

Modbus extension functions

Receive data from master
```
mb:receive()
```

receives data from master with 1-minute timeout

returns data as a binary string on success

returns nil, error description on error

Set modbus mapping of extension device
```
mb:setmapping(coils, inputs, holding_regs, input_regs)
```

creates memory mapping for the registers with size specified for each type

Handle extension
```
mb:handleslave()
```

waits for an incoming indication from master and sends a reply when necessary

Get functions
```
mb:getcoils(start, count)
mb:getdiscreteinputs(start, count)
mb:getinputregisters(start, count)
mb:getregisters(start, count)
```

gets one or many register/coil/input values from mapping from the start address

returns all values on success

returns nil, error description on error, exception code if applicable

Set functions
```
mb:setcoils(start, v1, [v2, [v3, ...]])
mb:setdiscreteinputs(start, v1, [v2, [v3, ...]])
mb:setinputregisters(start, v1, [v2, [v3, ...]])
mb:setregisters(start, v1, [v2, [v3, ...]])
```

sets value to register/coil mapping from the start address

returns true on success

returns nil, error description on error, exception code if applicable

Callback functions
```
mb:setwritecoilcb(fn)
mb:setwriteregistercb(fn)
```

sets a callback function for coil/register write event

callback should accept two parameters - coil/register address and value (boolean or number)

for multiple writes callback is executed for each coil/register separately

use nil to remove a callback.

EnOcean

EnOcean is an energy-harvesting wireless technology that enables communication between devices such as push buttons, thermostats, and PIR sensors within the controller ecosystem. By integrating EnOcean, the controller expands the possibilities of wired KNX installations, offering easy implementation and configuration.

To enable EnOcean technology in the controller, you will need to connect a USB EnOcean gateway (EnOcean USB Gateway, page 101).

EnOcean USB Gateway

- Compatibility:
  The controller EnOcean functions have been thoroughly tested with the EnOcean USB Gateway LSS10020040. Keep in mind that this specific product reference may not be available in all countries. However, any USB EnOcean gateway based on the EnOcean organization USB 300 (OEM) product can be used by third parties.
- Frequency considerations:
  EnOcean operates on different frequencies depending on the geographical region:
  - 868 MHz: Europe
  - 902 MHz: USA/Canada
  - 928 MHz: Japan
  - 2.4 GHz: worldwide usage
- The EnOcean USB gateway is a compact USB stick that connects various devices (PCs, consumer devices, DSL boxes) to EnOcean-based radio products.
- Equipped with a TCM 310 transceiver gateway module, it provides bidirectional EnOcean radio communication and a bidirectional serial interface via USB.
- Radio messages are transmitted and received through an externally connected USB host.
- Installation notes:
  - You can use a single EnOcean gateway connected to the USB port on the top of the controller case.
  - If needed, the gateway can be extended using an extension cable (up to a maximum of 5 meters).

EnOcean → Interfaces

To connect a USB EnOcean gateway to your controller, follow these steps:

1. Navigate to **Configurator** → **EnOcean** → **Interfaces** in the controller interface.
2. Connect your USB EnOcean gateway to the USB port on the controller.
3. Click **Rescan** in the lower-left corner.
4. Once the device is detected, it will appear in the list of interfaces.

EnOcean → KNX Mapping

EnOcean devices transmit telegrams periodically. When an EnOcean gateway receives a telegram, the corresponding device appears in the **Configurator** → **EnOcean** → **EnOcean >> KNX** section.

Most EnOcean devices include a dedicated button that allows immediate telegram transmission without waiting for periodic intervals.

To map a specific EnOcean device to KNX, follow these steps:

1. Click the corresponding row for the device in the list.
2. Specify the EnOcean **Profile** for the device.
3. Optionally, assign a device name.
4. Click **Save**.
5. All supported device profiles are listed in the Supported EnOcean Profiles, page 103 section.

Once the device profile is specified, you can proceed with mapping to KNX objects:

1. Click on the desired line in the device list to open the **Device mapping** dialog.
2. Each data object from the EnOcean device can be linked to a KNX object in the controller.
3. Select the appropriate controller object from the drop-down menu or create a new object directly from the dialog using the icon.
4. If the **Write to bus** parameter is enabled, the value is sent to the KNX TP bus.

When the EnOcean gateway receives a telegram from a specific device, the corresponding row is highlighted in green, and the respective KNX group address is updated with the new value from EnOcean.

KNX → EnOcean Mapping

In the **Configurator** section under **EnOcean** → **EnOcean >> KNX**, you can enable the capability to control EnOcean devices (such as actuators and dimmers) from your KNX installation via the controller.

Here's how it works:

1. Device simulation:
   - The controller simulates the behavior of specific EnOcean devices.
   - For example, an EnOcean switch actuator can be controlled by an EnOcean rocker switch.
   - To achieve this, the controller emulates the function of the rocker switch and manages the switch actuator.
2. Configuration steps:
   - Start by defining the device to be simulated by the controller.
   - Click the **Add new device** button in the lower-left corner.
   - In the **Device** dialog, specify a unique **Address**, **Device name**, and **Profile**. The profile represents the function of the device simulated by the controller.
   - Click **Save**.

3. Pairing with EnOcean devices:
   - Once the device is added, pair it with a specific device in the EnOcean network.
   - Set the EnOcean device to learning mode and then press the **Teach-in** button in the controller configuration.
4. Mapping to KNX Addresses:
   - The device created in the controller can now be mapped to specific KNX addresses.
   - Click on the desired line in the device list to open the **Device mapping** dialog.
   - Select the appropriate controller object from the drop-down menu or create a new object directly from the dialog using the icon.
   - Make sure to tick the **Send telegram** option. Otherwise, the EnOcean telegram will not be transmitted.
   - Click **Save**.
   - When the KNX object value changes, a telegram is sent to the device paired with the controller virtual device.

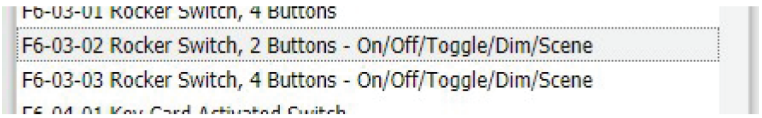
Supported EnOcean Profiles

ID	Profile Name	ID	Profile Name
00-00-00	RAW 4-bytes	A5-04-01	Temperature & Humidity Sensor (0C..40C, 0%..100%)
00-00-01	RAW 4-bytes, split	A5-04-02	Temperature & Humidity Sensor (-20C..60C, 0%..100%, Battery)
F6-01-01	Rocker Switch, 1 Rocker	A5-06-01	Light Sensor (300lx..60000lx)
F6-01-02	Rocker Switch, 1 Rocker (inverted)	A5-06-02	Light Sensor (0lx..1024lx)
F6-01-03	Rocker Switch, 1 Rocker (separate)	A5-06-03	Light Sensor (0lx..100lx, 300lx..30000lx)
F6-02-01	Rocker Switch, 2 Rocker	A5-07-01	Occupancy Sensor
F6-02-02	Rocker Switch, 2 Rocker (inverted)	A5-08-01	Light Sensor 0lx to 510lx, Temperature 0°C to +51° C and Occupancy
F6-03-01	Rocker Switch, 4 Buttons	A5-08-02	Light Sensor 0lx to 1020lx, Temperature 0°C to +51° C and Occupancy
F6-04-01	Key Card Activated Switch	A5-08-03	Light Sensor 0lx to 1530lx, Temperature -30°C to +50°C and Occupancy
F6-10-00	Window Handle	A5-09-04	Humidity, CO_2, Temperature Sensor
D5-00-01	Single Input Contact	A5-10-01	Temperature Sensor; Set Point, Fan Speed and Occupancy Control
A5-02-01	Temperature Sensor (-40C..0C)	A5-10-02	Temperature Sensor; Set Point, Fan Speed and Day/Night Control
A5-02-02	Temperature Sensor (-30C..10C)	A5-10-03	Temperature Sensor; Set Point Control
A5-02-03	Temperature Sensor (-20C..20C)	A5-10-04	Temperature Sensor; Set Point and Fan Speed Control
A5-02-04	Temperature Sensor (-10C..30C)	A5-10-05	Temperature Sensor; Set Point and Occupancy Control
A5-02-05	Temperature Sensor (0C..40C)	A5-10-06	Temperature Sensor; Set Point and Day/Night Control
A5-02-06	Temperature Sensor (10C..50C)	A5-10-07	Temperature Sensor; Set Point and Fan Speed Control
A5-02-07	Temperature Sensor (20C..60C)	A5-10-08	Temperature Sensor; Fan Speed Control
A5-02-08	Temperature Sensor (30C..70C)	A5-10-09	Temperature Sensor; Fan Speed and Day/Night Control

ID	Profile Name	ID	Profile Name
A5-02-09	Temperature Sensor (40C..80C)	A5-10-0A	Temperature Sensor; Set Point and Single Input Contact
A5-02-0A	Temperature Sensor (50C..90C)	A5-10-0B	Temperature Sensor and Single Input Contact
A5-02-0B	Temperature Sensor (60C..100C)	A5-10-0C	Temperature Sensor and Occupancy Control
A5-02-10	Temperature Sensor (-60C..20C)	A5-10-0D	Temperature Sensor and Day/Night Control
A5-02-11	Temperature Sensor (-50C..30C)	A5-10-10	Temperature and Humidity Sensor; Set Point and Occupancy Control
A5-02-12	Temperature Sensor (-40C..40C)	A5-10-11	Temperature and Humidity Sensor; Set Point and Day/Night Control
A5-02-13	Temperature Sensor (-30C..50C)	A5-10-12	Temperature and Humidity Sensor; Set Point Control
A5-02-14	Temperature Sensor (-20C..60C)	A5-10-13	Temperature and Humidity Sensor; Occupancy Control
A5-02-15	Temperature Sensor (-10C..70C)	A5-10-14	Temperature and Humidity Sensor; Day/Night Control
A5-02-16	Temperature Sensor (0C..80C)	A5-20-10	Generic HVAC interface
A5-02-17	Temperature Sensor (10C..90C)	A5-30-01	Single Input Contact, Battery Monitor
A5-02-18	Temperature Sensor (20C..100C)	A5-30-02	Single Input Contact
A5-02-19	Temperature Sensor (30C..110C)	A5-38-08-02	Dimmer
A5-02-1A	Rocker Switch, 2 Buttons – On/Off/ Toggle/Dim/Scene	F6-03-02	Light and blind control 2 buttons
A5-02-1B	Rocker Switch, 4 Buttons – On/Off/ Toggle/Dim/Scene	F6-03-03	Light and blind control 4 buttons

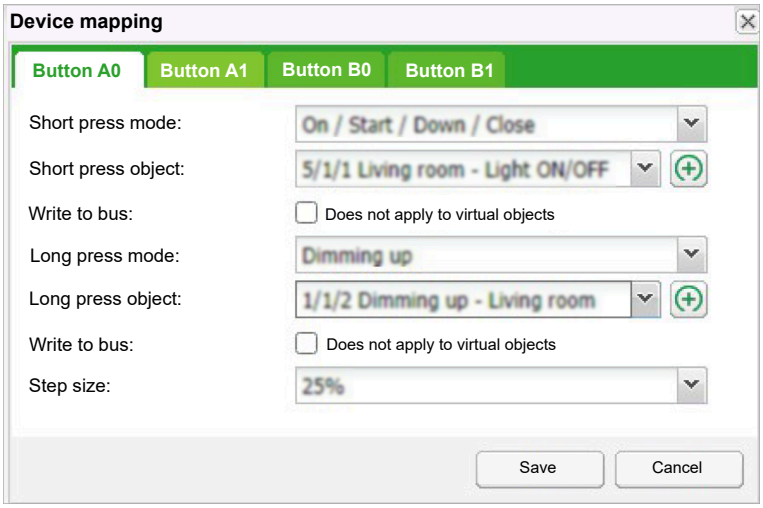
Blinds and Dimming with EnOcean

The controller now includes a new EnOcean feature implemented since firmware version 2.5.0. This feature enables the controller to handle dimming and blind functions while recognizing the long and short presses of EnOcean pushbuttons. Specifically, it applies to EnOcean profiles F06-03-02 and F06-03-03.



Here's how it works:

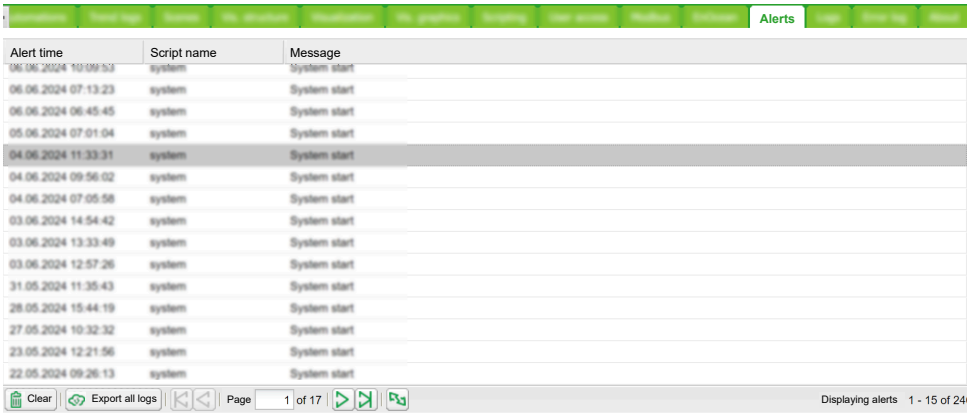
1. Mapping EnOcean Pushbuttons:
   - When mapping EnOcean pushbuttons, you will open a settings dialog that allows you to configure short and long press settings.



2. Short press modes:
   - Choose from the following short press modes:
     - **On/Start/Down/Close**
     - **Off/Stop/Up/Open**
     - **Toggle**
     - **Start/Stop**
     - **Stop/Start**
     - **Scene**
   - **Short press object**: Additionally, link a KNX object to the short press action.
3. Long press modes:
   - Long press modes include:
     - **No Action (disabled)**
     - **On/Start/Down/Close**
     - **Off/Stop/Up/Open**
     - **Toggle**
     - **Dimming up**
     - **Dimming down**
   - **Long press object**: Link a KNX object to the long press action.

Alerts

The **Alerts** tab displays a list of alert messages defined using the **alert** function within scripts. These messages are stored in the main database.



You have the option to export alerts to a CSV file.

Example

The alert has the following syntax:
`alert(message, [var1, [var2, [var3]]])`

It stores the alert message along with the current system time in the main database.

```
temperature = 25.3
if temperature > 24 then
-- resulting message: "Temperature level is too high: 25.3"
alert("Temperature level is too high: %.1f", temperature)
end
```

Logs

Logs are useful for debugging scripted code. The appearance of log messages is defined by the log function. You can also export logs to a CSV file.

Example

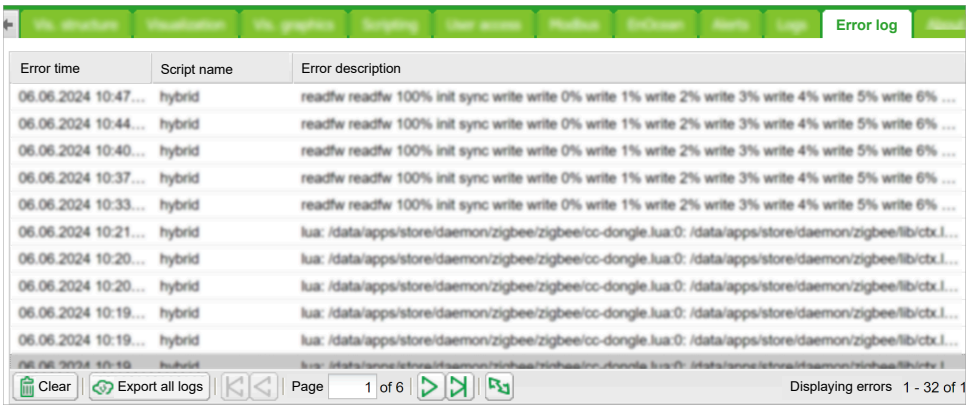
`log(var1, [var2, [var3, ...]])`

Converts variables to a human-readable form and stores them as follows:

```
-- log function accepts Lua nil, boolean, number and table (up to 5
nested levels) type variables
a ={ key1 ='value1', key2 =2}
b ='test'
c =123.45
-- logs all passed variables
log(a, b, c)
```

Error Log

Error messages are displayed in the **Error log** tab.



You have the option to export the error messages to a CSV file.

Additionally, clicking a specific error message allows you to view its details in a separate dialog.

HomeKit Plugin

The **HomeKit** plugin for Wiser for KNX/spaceLYnk controller is designed to control KNX devices in the installation through Apple´s **Home** application, available in Apple devices like iPhone, iPad, iWatch, Mac, Apple TV, etc. (hereafter referred to as the app/mobile application).

System Requirements

To ensure optimal performance and compatibility with HomeKit plugin, please make sure you meet the following system requirements:

Hardware	Wiser for KNX or spaceLYnk controller	Ensure you have a hardware version greater than 2.
Firmware	Latest firmware	Make sure you have the latest firmware installed. The latest firmware is available for download at se.com.
Software	Plugin	Ensure the KNX IoT 3rd Party API plugin is installed and enabled (for Local). Enabling automatic updates is recommended to ensure you always have the latest version of the plugin installed.

Limitations of the HomeKit Plugin:

- Control of slats in blinds (not supported).
- Stop commands for shutters/blinds.
- Heating works only in the room temperature controller.
- KNX scenes cannot be called, but new scenes can be created in the **Home** app.
- Separate settings from the **Touch** plugin.
- Terminology/naming of group objects is different from KNX conventions.
- In larger or more complex KNX installations, the app may behave unreliably.

Home App

The **Home** app can be operated in two ways:

1. **Without a home hub**: You can control the KNX installation locally from your mobile device using the **HomeKit** plugin.
2. **With a home hub (e.g., Apple TV)**: With a home hub, you can access the installation remotely via the internet through Apple Cloud. Additionally, you can set up automations/scenes and share your home with other users.

Main features of the Apple Home app

The **Home** app is a centralized hub to control and manage your smart home devices. Here are the key features:

1. **Device control**: You can manage a variety of smart devices such as lights, thermostats, cameras, and locks directly from the app. Toggle devices on or off and adjust settings to suit your preferences.
2. **Scenes**: Create customized scenes that allow you to control multiple devices simultaneously.
3. **Automations**: You can set up automations to trigger actions based on specific conditions such as time of day, location, or device activity.
4. **Secure remote access**: Access and control your devices remotely through an Apple TV or HomePod acting as a hub. This ensures you can monitor and manage your home even when you're away.
5. **User sharing**: Share control of your devices with family members and friends, allowing them to access and manage devices from their own Apple devices.
6. **Room and zone management**: Organize your smart home devices by assigning them to specific rooms and zones within the app.

For more information about the Apple **Home** app, visit https://www.apple.com/uk/home-app/.

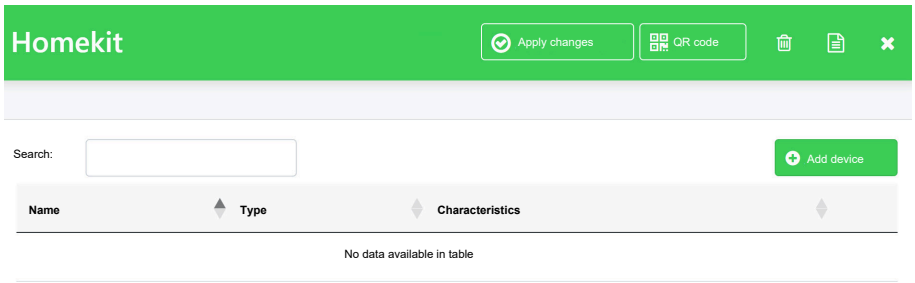
You can find the user guide for the Apple **Home** app at https://support.apple.com/.

HomeKit Plugin Installation

1. Log in to the controller through your web browser.
2. Click on  in the top right corner to open the **Marketplace**.
3. Click on the green bar in the middle to display the applications available for installation under **Install new apps and widgets**.
4. From the list of available applications, select the **HomeKit** plugin and click  on the right to start the installation.
5. Return to the **Start page** of your controller in the web browser. The installed **HomeKit** plugin will appear among other applications:

Homekit

6. Click on the **HomeKit** icon to open the plugin:



NOTE: We recommend enabling the plugin's automatic update to ensure you always have the latest version.

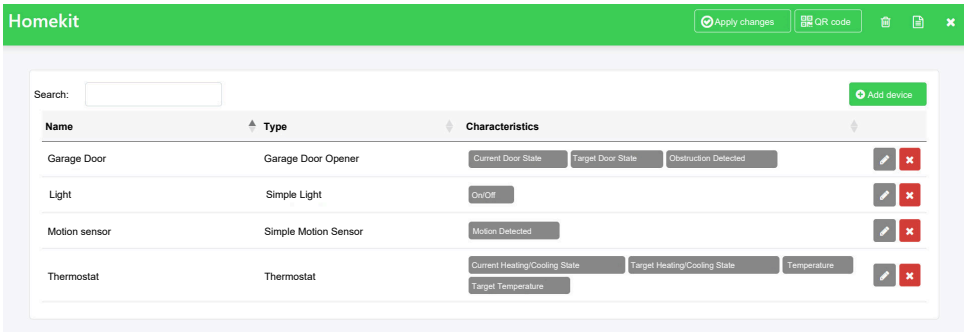
Adding Devices into HomeKit Plugin

To control your devices through the **Home** app, you need to create new widgets within the **HomeKit** plugin. This allows you to control your devices directly from the app on your Apple device.

Follow these steps:

1. Click on the **Add device** button in the top right corner.
2. Name your device: Give it a unique identifier.
3. Select the type of device: Available objects for the device will appear. Objects are divided into three groups (see more here: Types of Devices Available in the HomeKit Plugin, page 111):
   - **Mandatory** (marked with a star): These need to be mapped for the device to be added.
   - **Required**: Necessary for the proper functioning of the device.
   - **Optional**: These can be mapped at your discretion.

After mapping the objects, click on **Save**, and your device will appear in the **HomeKit** plugin device list.



For each device, all mandatory objects are displayed in the **Characteristics** column.

Finally, click on the **Apply changes** button to save the updates.

NOTE:

- The **Apply changes** button will turn red when there are new changes that need confirmation, signaling that you have changes pending to be saved.
- If you want your changes to be saved, click on **Apply changes**.
- This button ensures that the changes you make will be reflected in the mobile application.

After adding devices to the **Home** app, you can assign them to your preferred room structure. This structure can match or differ from your controller's visualization.

Adding Accessory to the Home App via QR Code

After mapping the devices in the **HomeKit** plugin, transfer them to the **Home** app using a bridge created by the mobile app. This bridge represents the controller. To complete the transfer, use the QR code generated by the **HomeKit** plugin.

Follow these steps:

1. Make sure that your mobile device and **HomeKit** plugin are connected to the internet on the same network.

2. In the **HomeKit** plugin, on the main screen, click on the **QR code** button at the top right. A **Setup code** dialog will open with a generated QR code for pairing.
3. Open the mobile application on your mobile device, and tap **Add Accessory**.
4. Point your mobile device at the QR code in the **HomeKit** plugin.
5. The mobile app will load a bridge from your **HomeKit** plugin. You can then name your devices using the wizard and assign them to individual rooms in the mobile app.

NOTE: The Apple **Home** app supports adding multiple bridges, allowing you to integrate other Apple **HomeKit**-compatible systems and devices, such as cameras and sensors.
Additionally, the Apple **Home** app enables you to create automations and scenes that combine all the devices integrated into the **Home** app.
For more information, please refer to the Apple **HomeKit** user guide.

Editing and Deleting Devices in HomeKit Plugin

You can later edit/delete the devices mapped in the **HomeKit** plugin:

1. Editing devices:
   - Click  on the right of your device's row.
   - Make the intended changes.
   - Click on **Save** and then **Apply changes** to save the updates.

2. Deleting devices:
   - Click  on the right of your device's row.
   - In the **Delete device** dialog, click on **Delete**.
   - Remember to click on **Apply changes**.

Types of Devices Available in the HomeKit Plugin

This list includes all types of devices available in the **HomeKit** plugin, along with their available characteristics (objects) for mapping and DPT. It details the naming and structure of the **HomeKit** plugin based on Apple's **Home** app specification. To simplify the creation of widgets, we have added information on the equivalent naming for the group objects currently used in the **Touch** widget configuration, where available.

Simple Controls

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
SIMPLE LIGHT	On/Off Control * (bool) On/Off Status (bool)		Light switch	Switch object Status feedback object
SIMPLE TEMPERATURE SENSOR	Temperature Status * (float16)		Co_2/humidity/temp sensor	Temperature value

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
SIMPLE HUMIDITY SENSOR	Humidity Status * (scale)		Co_2/humidity/temp sensor	Humidity value
SIMPLE LIGHT SENSOR	Light Status * (float16)		Weather station Motion sensor	Brightness value
SIMPLE LEAK SENSOR	Leak Detected Status * (bool)		Water leak alarm	Alarm object, alert
SIMPLE MOTION SENSOR	Motion Detected Status * (bool)		Motion sensor	Detection
SIMPLE OCCUPANCY SENSOR	Occupancy Detected Status * (bool)		Motion sensor	Detection
SIMPLE SMOKE SENSOR	Smoke Detected Status * (bool)		Fire alarm	Alarm object, alert
SIMPLE OUTLET	On/Off Control * (bool) On/Off Status (bool)		Socket switch	Switch object Status feedback object
SIMPLE FAN	Active Control * (bool) Active Status (bool)		Fan switch	Switch object Status feedback object
SIMPLE SWITCH	On/Off Control * (bool) On/Off Status (bool)		General switch	Switch object Status feedback object
SIMPLE AIR QUALITY SENSOR	Air Quality Status * (uint8)	0 = Unknown, 1 = Excellent, 2 = Good, 3 = Fair, 4 = Inferior, 5 = Poor	N/A	
SIMPLE CARBON MONOXIDE SENSOR	Carbon Monoxide Detected Status * (bool)		Air quality sensor	Carbon monoxide
SIMPLE CONTACT SENSOR	Contact Sensor State Status * (bool)		Window/Door sensor, Any switch	Switch object

Other Controls

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
LIGHT ON/OFF BRIGHTNESS	On/Off Control * (bool) On/Off Status (bool) Brightness Control * (scale) Brightness Status (scale)		Dimmer rotary Dimmer horizontal Dimmer vertical	Switch Object Status feedback object Value object Status feedback value object
LIGHT BRIGHTNESS	Brightness Control * (scale) Brightness Status (scale)	DPT of the general object must be a scale. If the object's DPT is an unsigned integer, it is not recognizable.	Dimmer rotary Dimmer horizontal Dimmer vertical	Value object Status feedback value object
LIGHT RGB	RGB Control * (rgb) RGB Status (rgb)		RGB	RGB object RGB status object
LIGHT COLOR TEMPERATURE	On/Off Control * (bool) On/Off Status (bool) Brightness Control * (scale) Brightness Status (scale) Color Temperature Control * (uint16)		Dimmer rotary Dimmer horizontal Dimmer vertical	Switch Object Status feedback object Value object Status feedback value object

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
	Color Temperature Status (uint16)			
THERMOSTAT	Current Heating / Cooling State Status * (uint8) Target Heating / Cooling State Control * (uint8) Target Heating / Cooling State Status (uint8) Temperature Status * (float16) Target Temperature Control * (float16) Target Temperature Status (float16)	0 = Off, 1 = Heat, 2 = Cool 0 = Off, 1 = Heat, 2 = Cool, 3 = Auto 0 = Off, 1 = Heat, 2 = Cool, 3 = Auto	Thermostat	Current Temperature Current setpoint temperature input Current setpoint temperature output
Optional	Humidity Status (scale) Target Relative Humidity Control (scale) Target Relative Humidity Status (scale) Cooling Threshold Temperature Control (float16) Cooling Threshold Temperature Status (float16) Heating Threshold Temperature Control (float16) Heating Threshold Temperature Status (float16) Temperature Display Units Control (bool) Temperature Display Units Status (bool)	0 = Celsius, 1 = Fahrenheit 0 = Celsius, 1 = Fahrenheit		
LOCK MECHANISM	Lock Target State Control * (bool) Lock Target State Status (bool) Lock Current State Status * (uint8)	0 = Unsecured, 1 = Secured 0 = Unsecured, 1 = Secured 0 = Unsecured, 1 = Secured, 2 = Jammed, 3 = Unknown	N/A	
DOOR	Current Position Status * (scale) Target Position Control * (scale) Target Position Status (scale)		Shutter rotary Shutter horizontal Shutter vertical	Status feedback for height Height position (Status feedback for height)
Optional	Position State Status (uint8) Invert value checkbox	0 = going to minimum, 1 = going to maximum, 2 = stopped Option to invert the value		
WINDOW	Current Position Status * (scale) Target Position Control * (scale)		Shutter rotary Shutter horizontal Shutter vertical	Status feedback for height Height position

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
	Target Position Status (scale)			(Status feedback for height)
Optional	Position State Status (uint8) Invert value checkbox	0 = going to minimum, 1 = going to maximum, 2 = stopped Option to invert the value		
WINDOW COVERING	Current Position Status * (scale) Target Position Control * (scale) Target Position Status (scale)		Shutter rotary Shutter horizontal Shutter vertical	Status feedback for height Height position (Status feedback for height)
Optional	Position state status Invert value checkbox	0 = going to minimum, 1 = going to maximum, 2 = stopped Option to invert the value		
AIR PURIFIER	Active Control * (bool) Active Status (bool) Current Air Purifier State Status * (uint8) Target Air Purifier State Control * (bool) Target Air Purifier State Status (bool)	0 = Inactive, 1 = Idle, 2 = Purifying Air 0 = Manual, 1 = Auto 0 = Manual, 1 = Auto	N/A	
Optional	Lock Physical Controls Control (bool) Lock Physical Controls Status (bool) Rotation Speed Control (scale) Rotation Speed Status (scale) Swing Mode Control (bool) Swing Mode Status (bool)			
HEATER COOLER	Active Control * (bool) Active Status (bool) Current Heater / Cooler State Status * (uint8) Target Heater / Cooler State Control * (uint8) Target Heater / Cooler State Status (uint8) Temperature Status * (float16)	0 = Inactive, 1 = Idle, 2 = Heating, 3 = Cooling 0 = Auto, 1 = Heat, 2 = Cool 0 = Auto, 1 = Heat, 2 = Cool	Thermostat	Switch on/off thermostat Switch on/off thermostat status Current Temperature
Optional	Lock Physical Controls Control (bool) Lock Physical Controls Status (bool) Rotation Speed Control (scale) Rotation Speed Status (scale)	0 = Celsius, 1 = Fahrenheit 0 = Celsius, 1 = Fahrenheit		

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
	Swing Mode Control (bool) Swing Mode Status (bool) Cooling Threshold Temperature Control (float16) Cooling Threshold Temperature Status (float16) Heating Threshold Temperature Control (float16) Heating Threshold Temperature Status (float16) Temperature Display Units Control (bool) Temperature Display Units Status (bool)			
HUMIDIFIER/ DEHUMIDIFIER	Active Control * (bool) Active Status (bool) Current Humidifier / Dehumidifier State Status * (uint8) Target Humidifier / Dehumidifier State Control * (uint8) Target Humidifier / Dehumidifier State Status (uint8) Humidity Status * (scale)	0 = Inactive, 1 = Idle, 2 = Heating, 3 = Humidifying, 4 = Dehumidifying 0 = Auto, 1 = Humidifier, 2 = Dehumidifier 0 = Auto, 1 = Humidifier, 2 = Dehumidifier	N/A	
Optional	Lock Physical Controls Control (bool) Lock Physical Controls Status (bool) Rotation Speed Control (scale) Rotation Speed Status (scale) Swing Mode Control (bool) Swing Mode Status (bool) Relative Humidity Dehumidifier Threshold Control (scale) Relative Humidity Dehumidifier Threshold Status (scale) Relative Humidity Humidifier Threshold Control (scale) Relative Humidity Humidifier Threshold Status (scale) Water Level Status (scale)			

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
VALVE	Active Control * (bool) Active Status (bool) In Use Status * (bool)		Thermostat	Switch on/off thermostat Switch on/off thermostat status
Optional	Valve Type Status (uint8)	0 = Water faucet, 1 = Irrigation, 2 = Shower head		
AIR QUALITY SENSOR	Air Quality Status * (uint8)	0 = Unknown, 1 = Excellent, 2 = Good, 3 = Fair, 4 = Inferior, 5 = Poor	Air quality Particle sensor	
Optional	Nitrogen Dioxide Density Status (float16) Ozone Density Status (float16) PM10 Density Status (float16) PM2.5 Density Status (float16) Sulphur Dioxide Density Status (float16) VOC Density Status (float16) Status Active Status (bool) Status Fault Status (bool) Status Low Battery Status (bool) Status Tampered Status (bool)			NO_2 (Air quality) Ozone (Air quality) Part.10 (Particle sensor) Part.2.5 (Particle sensor) TVOC (Air quality)
SECURITY SYSTEM	Security System Current State Status * (uint8) Security System Target State Control * (uint8) Security System Target State Status (uint8)	0 = Stay Arm, 1 = Away Arm, 2 = Night Arm, 3 = Disarmed, 4 = Alarm Triggered 0 = Stay Arm, 1 = Away Arm, 2 = Night Arm, 3 = Disarm 0 = Stay Arm, 1 = Away Arm, 2 = Night Arm, 3 = Disarm	N/A	
Optional	Security System Alarm Type Status (bool) Status Fault Status (bool) Status Tampered Status (bool)			
GARAGE DOOR OPENER	Current Door State Status * (uint8) Target Door State Control * (bool) Target Door State Status (bool) Obstruction Detected Status * (bool)	0 = Open, 1 = Closed, 2 = Opening, 3 = Closing, 4 = Stopped 0 = Open, 1 = Closed 0 = Open, 1 = Closed	N/A	

Device type	Required characteristics (* Mandatory)	Parameters	Equivalent widget (Touch)	Group objects (Touch)
Optional	Lock Current State Status (uint8) Lock Target State Control (bool) Lock Target State Status (bool)	0 = Unsecured, 1 = Secured, 2 = Jammed, 3 = Unknown 0 = Unsecured, 1 = Secured 0 = Unsecured, 1 = Secured		

Web Server Software NGINX

NGINX, the web server software used for the controller, offers improved performance with minimal memory usage. Find more information at https://www.nginx.com.

If you are using pre-1.2 versions of firmware, remember to clear your browser cache after a firmware upgrade. Additionally, be aware that **Nginx** uses different links to pages compared to older versions:

- Main page link (old versions):
  - Example: http://your_IP_address/cgi-bin/scada-vis/index.cgi
- Main page link (**NGINX**):
  - Example: http://your_IP_address/scada-vis

RS-232 Serial Line

The RS-232 serial interface communication standard remains one of the most widely used connections for serial data transmission due to its simplicity and reliability. Despite the emergence of new, higher-speed standards, RS-232 continues to offer robust performance.

Key Points:

- **Widespread use:** RS-232 is still found on some computers and many interfaces.
- **Applications:** It serves various purposes, from data acquisition to providing a serial data communication facility in general computing environments.
- **Cost-effective:** Products adhering to the RS-232 standard are readily available and affordable.
- **Distance limit:** The interface is designed to operate over distances of up to 15 meters, following a one master/one extension rule.

Application Examples:

1. **Connection to simple devices:** RS-232 is commonly used to connect to straightforward devices or other bus subsystems.
2. **Audio/video and IR system integration:** It plays a role in integrating audio/video systems and infrared (IR) devices.

Configuration Commands

- **Open connection:**
```
require('serial')
port = serial.open('/dev/RS232', {baudrate = 9600})
```
- **Write to Port:**
```
port:write('test data')
```
- **Blocking Read:**
```
-- Script will block until 10 characters are read
data = port:read(10)
```
- **Timeout Read**
```
-- Script will wait for 10 characters for 20 seconds
data = port:read(10, 20)
```
- **Close serial port**
```
port:close()
```

RS-485 serial line

Controlled similarly using the same configuration commands. The only difference lies in the **serial.open** command:

```
port = serial.open('/dev/RS485', {baudrate = 9600})
```

For more detailed information about RS-232 communication, please refer to the respective application note available at www.se.com Application Notes, page 122).

USB

- Devices complying with the USB specification can consume a total of 500 mA from the bus. This allows devices with a power requirement of up to 2.5 W to be supplied via the bus.
- Integrated voltage supply (5 VDC) is available for devices connected via the 4-pole USB cable.
- USB cables must not exceed a maximum length of 4.5 meters according to the specification.
- Data transmission is possible in both directions.
- The USB interface is commonly used for extending memory capacity by connecting USB flash drives to computers or other devices.

Configuration Commands

1. `io.readfile (file)`
   - Reads the entire file at once.
   - Returns the file contents as a string on success or *nil* on error.
2. `io.writefile (file, data)`
   - Writes the given data to a file.
   - Data can be either a value convertible to a string or a table of such values.
   - When data is a table, each table item is terminated by a new line character.
   - Returns a Boolean indicating the write result (whether the file can be opened for writing) or *nil* if the file cannot be accessed.

USB flash drive:

- Supports FAT, FAT32, and NTFS file systems.

Send and receive SMS messages via USB GSM adapter:

- Use USB data modem.
- Plug the modem into the USB port of the controller; it will start operating immediately.
- Specific functions should be added to the user script library, including PIN code settings and a telephone number whitelist for sending and receiving SMS messages.

Command syntax for SMS:

- To change an object value using SMS or read an object value via SMS request, follow the format below:
  - Write to bus:
    - SMS command format:
      `W ALIAS VALUE`
  - Read from bus:
    - SMS command format:
      `R ALIAS`
    - On read request, the script will reply with an SMS message containing the current value of the selected object.

ALIAS can be:

- Group address (e.g., 1/1/1)
- Name (e.g., Obj1). If the name contains spaces, escape it using double quotes (e.g., *Room temperature*).

Object data type and name:

- Set in the **Configurator** → **Objects** tab.
- Without proper configuration, the script will not be able to read from or write to the object.

NOTE:

- Only ASCII symbols are accepted in the message.
- For more details on sending SMS, refer to the application note document AN011 (Application Notes, page 122).

Application Notes

The application notes provide detailed descriptions on various topics related to controller settings, emphasizing effective implementation. You can access them on the Schneider Electric website.

Wiser for KNX

- AN001_Wiser for KNX integration using BACnet
- AN003_Modbus power meters and Wiser for KNX
- AN005_PIR trend and logs with Wiser for KNX
- AN006_Advance techniques in visualization (Wiser for KNX)
- AN010_RS232 control with Wiser for KNX
- AN011_Email, SMS and FTP in Wiser for KNX
- AN015_Wiser for KNX visualization shortcut on the desktop
- AN016_Wiser for KNX as a Modbus extension
- AN017_Addressable limits of the Wiser for KNX controller
- AN018_Advanced project tutorial
- AN019_Exiway Power Control connected to SBO via Wiser for KNX
- AN023_Redundant operation mode of two Wisers for KNX
- AN025_Advanced graphic's tutorial
- AN027_Creation of Modbus profile
- AN028_Wiser for KNX upgrade procedure
- AN029_Umotion client touch panels 10 and 15
- AN032_Lighting solution for comfortable environment using Wiser for KNX
- AN033_DMX control with Wiser for KNX
- AN034_Lighting solution for comfortable environment using Wiser for KNX II
- AN037_Modbus integration of room controllers SE8000 series
- AN041_CoolMasterNet
- AN042_Optimalization of visualization in complex projects
- AN044_Power_dissipation of DIN rail KNX
- AN046_Programming in Lua with Wiser for KNX
- AN047_ Corridor Linking using LUA scripts with Wiser for KNX
- AN048_ Monthly Email Backup File of Wiser for KNX
- AN049_ Exiway Smart Control and Wiser for KNX
- AN2_001_ How to flash microSD card in Wiser for KNX
- AN2_003_ Connectivity Ecosystem
- AN2_005_ Power Dissipation of KNX devices
- AN2_100_ RGBW-Dimming with KNX Multitouch Pro
- AN2_101_ Authorize actions on KNX with RFID-Tags
- AN2_102_ Revox Integration
- AN2_103_ How to fix issue with Bootloader in Wiser for KNX / Wiser for KNX
- AN2_104_Alexa Voice control integration
- AN2_105_Touch 3
- AN106_Wiser for KNX – How to change DNS settings for remote connectivitycontroller's DNS
- AN107_Wiser for KNX, SpaceLYnk- System Hardening Guideline

- AN108_Wiser for KNX – Google Assistant integration
- AN109_Wiser for KNX – KNX IoT 3rd Party API Guidelines

spaceLYnk

- AN001_spaceLYnk integration using BACnet
- AN003_Modbus power meters and spaceLYnk
- AN005_PIR trend and logs with spaceLYnk
- AN006_Advance techniques in visualization (spaceLYnk)
- AN010_RS232 control with spaceLYnk
- AN011_Email, SMS and FTP in spaceLYnk
- AN015_spaceLYnk visualization shortcut on the desktop
- AN016_spaceLYnk as a Modbus extension
- AN017_Addressable limits of the spaceLYnk controller
- AN018_Advanced project tutorial
- AN019_Exiway Power Control connected to SBO via spaceLYnk
- AN023_Redundant operation mode of two spaceLYnks
- AN025_Advanced graphic's tutorial
- AN027_Creation of Modbus profile
- AN028_spaceLYnk upgrade procedure
- AN029_Umotion client touch panels 10 and 15
- AN032_Lighting solution for comfortable environment using spaceLYnk
- AN033_DMX control with spaceLYnk
- AN034_Lighting solution for comfortable environment using spaceLYnk II
- AN037_Modbus integration of room controllers SE8000 series
- AN041_CoolMasterNet
- AN042_Optimalization of visualization in complex projects
- AN044_Power_dissipation of DIN rail KNX
- AN046_Programming in Lua with spaceLYnk
- AN047_ Corridor Linking using LUA scripts with spaceLYnk
- AN048_ Monthly Email Backup File of spaceLYnk
- AN049_ Exiway Smart Control and spaceLYnk
- AN2_001_ How to flash microSD card in spaceLYnk
- AN2_003_ Connectivity Ecosystem
- AN2_005_ Power Dissipation of KNX devices
- AN2_100_ RGBW-Dimming with KNX Multitouch Pro
- AN2_101_ Authorize actions on KNX with RFID-Tags
- AN2_102_ Revox Integration
- AN2_103_ How to fix issue with Bootloader in spaceLYnk / Wiser for KNX
- AN2_104_Alexa Voice control integration
- AN2_105_Touch 3
- AN2_107_ Wiser for KNX, SpaceLYnk- System Hardening Guideline
- AN109_Wiser for KNX – KNX IoT 3rd Party API Guidelines

Open Source Licence Information

This Schneider Electric product is provided with certain open source software components (OSS) developed by third parties. The OSS is licensed in accordance with the OSS license terms set below.

The following open-source software components are available under the terms of the GNU General Public License (https://www.gnu.org/licenses/gpl-3.0.html).

Linux Kernel	Copyright © 1991-1992 Linus Torvalds	https://spdx.org/licenses/Linux-syscall-note.html
LEDE Build System	Copyright © 2016 LEDE Project	
Base-files	Copyright © 2007-2016 OpenWrt.org Copyright © 2010 Vertical Communications	
Eibd	Copyright © 2005-2011 Martin Koegler	
Luarrd	Copyright © 1997-2015 Tobias Oetiker	https://oss.oetiker.ch/rrdtool/prog/rrdlua.en.html
Opkg	Copyright © 2011 Mikhail Gusarov	
Rrdtool	Copyright © 1997-2015 Tobias Oetiker	https://oss.oetiker.ch/rrdtool/license.en.html
Pcsk	Copyright © 2004 Norbert Buchmuller	
Bacnet	Copyright © 2012 Steve Karg	https://github.com/stargieg/bacnet-stack/blob/master/demo/object/bacfile.h
vsftpd	Copyright © Chris Evans	https://github.com/dagwieers/vsftpd/blob/master/LICENSE

The following open-source software components are available under the terms of the GNU Lesser General Public License (https://www.gnu.org/licenses/lgpl-3.0.en.html).

Libmodbus	Copyright © 2008-2014 Stephane Raimbault	https://libmodbus.org/
px5g	Copyright © 2009 Steven Barth Copyright © 2014 Felix Fietkau	
UCI	Copyright © 2008-2014 OpenWrt.org Copyright © 2016 LEDE Project	https://openwrt.org/docs/guide-user/base-system/uci
GNU Lesser General Public License	Copyright © 1991, 1999 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110, USA	https://www.gnu.org/licenses/lgpl-3.0.html#license-text

Printed in:
Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison - France
+ 33 (0) 1 41 29 70 00

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

2410_LSS100100/LSS100200_SW_03.0