SpaceLogic IHC Interface

LSS100500

User Guide

This document describes the SpaceLogic IHC Interface solution (features and user interface), which consists of a DIN device and the plugin.
Release date 22/04/2026

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

Table of Contents 3
Safety Information 5
About the Document 7
Introduction 9
Workflow 11
IHC Plugin Installation 12
Pairing the IHC Interface with the Controller 13
- Operating elements, terminals 15
- Device and Factory Reset 16

Import Existing IHC Project 18
Device Mapping 20
- Understanding Standard/Virtual KNX Group Addresses in the Controller 20
  - Standard KNX Group Addresses 20
  - Virtual KNX Group Addresses 21
  - Creating KNX Objects in the Controller 22
- Input Devices 23
- Output Devices 25
- In-Out View 27
- RS485 28

About 30
Use Cases 34
- LED Dimmer Configuration 34
  - Importing LED Dimmer Configuration 34
  - Manual Configuration 34
  - Checking LED Dimmer Functionality 36
  - Unlink/Clear Command 37
  - Advanced Channel Scan 37
- LED Dimmer Toggle Button Functionality 38
  - Configuring Binary Input as Toggle Button 38
  - Connecting Toggle Button to LED Dimmer 39
  - Multiple Toggle Buttons 40
  - Feedback with LED Diode 41
- Scene Control with LED Dimmers 42
  - Configuring Scene Control for LED Dimmers 42
  - Scene Recall Functionality 43
  - Scene Store Functionality 43
  - Configuring a Push Button for Scene Recall/Store 44
  - Feedback from Scene Recall/Store 45
- Short/Long Press Functionality of a Push Button 46
  - Controlling Two Lights with a Single Button 46
  - Controlling One Light and Recalling a Scene with a Single Button 47
  - Creating “Follow” Functionality with Short/Long Press 50
- Staircase Timer Setup for LED Dimmers 51
- PIR Sensor Based Light Control 53

PIR Sensor Input Configuration....54
PIR Timer Configuration....55
Toggle Button Configuration....56
Scenes for the Light Level Control....56
Automation Configuration....57
Testing the PIR Sensor Functionality....60
NTP Client/Server....61

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 documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this 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 follow 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.

⚠ 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.

The addition of this 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 follow 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.

⚠ 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.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Document

Document Scope

This document provides a comprehensive user guide for the **IHC Plugin** designed for the LSS100100 Wiser for KNX/homeLYnk controller (controller). It outlines the installation, configuration, and usage of the plugin, which facilitates communication between the LSS100500 **SpaceLogic IHC Interface** and the controller.

The guide is intended for system integrators, installers, and advanced users who wish to integrate an existing IHC system with a KNX-based smart home or building automation system.

It also covers the use of the LSS100400 **Hybrid module** for Zigbee device integration and the process of importing existing IHC projects into the plugin.

Validity Note

This user guide is applicable to the **IHC Plugin** and the LSS100500 **SpaceLogic IHC Interface**. It provides detailed instructions on the installation, configuration, and usage of these components. Ensure you are referring to the latest version of this document to access the most up-to-date information and features. For any updates or additional support, please visit the official Schneider Electric website.

General Cybersecurity Information

In recent years, the growing number of networked machines and production plants has seen a corresponding increase in the potential for cyber threats, such as unauthorized access, data breaches, and operational disruptions. You must, therefore, consider all possible cybersecurity measures to help protect assets and systems against such threats.

To help keep your Schneider Electric products secure and protected, it is in your best interest to implement the cybersecurity best practices as described in the Cybersecurity Best Practices document.

Schneider Electric provides additional information and assistance:

- Subscribe to the Schneider Electric security newsletter.
- Visit the Cybersecurity Support Portal web page to:
  - Find Security Notifications.
  - Report vulnerabilities and incidents.
- Visit the Schneider Electric Cybersecurity and Data Protection Posture web page to:
  - Access the cybersecurity posture.
  - Learn more about cybersecurity in the cybersecurity academy.
  - Explore the cybersecurity services from Schneider Electric.

Product Related Cybersecurity Information

Available Languages of the Document

The document is available in these languages:

- English

Related Documents

Title of documentation	Reference number
SpaceLogic KNX Hybrid Module User Guide	LSS100400_SW_EN
SpaceLogic KNX LSS100100 Wiser for KNX/ LSS100200 spaceLYnk User Guide	LSS100100_LSS100200_SW
Wiser KNX Mobile Application User Guide	LSS100100_W4K_App_EN

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Trademarks

QR Code is a registered trademark of DENSO WAVE INCORPORATED in Japan and other countries.

SpaceLogic™ and Wiser™ are trademarks and the property of Schneider Electric SE, its subsidiaries and affiliated companies.

IHC®, IHC Control®, IHCRemote®, OPUS®, and FUGA® are trademarks and the property of Schneider Electric SE, its subsidiaries and affiliated companies.

Zigbee® is a registered trademark of the Connectivity Standards Alliance.

Other brands and registered trademarks are the property of the relevant owner.

Introduction

The **SpaceLogic IHC Interface** is a DIN rail-mounted device designed to work in tandem with the Wiser for KNX/homeLYnk controller (referred to as the controller). It is not a standalone unit. The **IHC Interface** allows the controller to receive signals from input devices connected to IHC Input modules and to control output devices connected to IHC Output modules. Together, the controller and the **IHC Interface** effectively replace an **IHC Controller** version 2 or 3, enabling continued use of the existing IHC wiring and connected devices.

To establish communication between the **IHC Interface** and the controller, users must install the **IHC Plugin**, a downloadable application available from the W4K Marketplace. This plugin pairs the **IHC Interface** with the controller and enables the mapping of IHC addresses to KNX group addresses and vice versa, which is essential for integrating IHC inputs and outputs into the KNX system.

Internally, the controller operates using KNX group addresses, and this mapping allows seamless interaction with devices connected via the IHC modules. Once connected and configured, the system can be reprogrammed to restore most of its original functions. Additionally, it can be expanded with KNX devices or Wiser/ Zigbee devices using the **SpaceLogic KNX Hybrid module**.

With the **IHC Plugin**, your current IHC installation can be managed through the controller. Integration with Zigbee devices is supported via the **KNX Hybrid module**, and the entire system can be controlled using the Wiser KNX mobile app.

Limitations and Considerations

- Functions previously controlled wirelessly via **IHC Remote** application cannot be re-established directly and must be rebuilt and integrated via KNX and the controller.
- **Room temperature/heating control** can be configured via KNX and the controller.
- The **IHC Remote** application is not supported and must be replaced by the Wiser KNX mobile application.
- The **IHC Visual** and **IHC FirmwareLoader** applications are not supported.

However, you can import device settings from your existing IHC **Visual 3** project into the plugin to facilitate management and migration of your IHC installation.

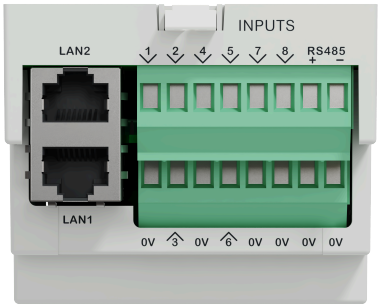
Prerequisites

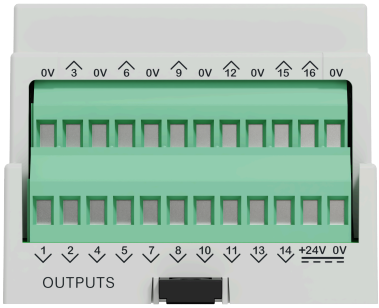
Before installing and using the **IHC Plugin**, ensure the following prerequisites are met:

Hardware requirements	• Wiser for KNX/homeLYnk controller with access to the web interface. Minimum hardware version: 3.2 Minimum firmware version: 3.1 • **SpaceLogic IHC Interface** connected to the controller via ethernet cable. • Properly wired IHC system, including input/output modules. • Optional: **IHC LED dimmer** • Optional: **KNX Hybrid module** for Wiser Zigbee device integration (via RS232). See the **KNX Hybrid module** user guide available here.
Software & tools	• Access to the **Marketplace** via the controller's web interface. • **eConfigure KNX** or **controller web Interface** for firmware upgrades and configuration. • Web browser for accessing the controller and plugin interface. • To import your existing IHC configuration, have your IHC **Visual 3** project file (`.vis`) ready. • To configure the **SpaceLogic IHC Interface** via a web browser, it is necessary to use an HTTPS connection. Using HTTP may result in limited or improper functionality.

User permissions	Administrator access to the controller to install applications and perform pairing.
Network & connectivity	• Stable local network connection between the controller and the **SpaceLogic IHC Interface**. • The controller and the **SpaceLogic IHC Interface** must be connected to the same LAN network. • Internet access (recommended) for downloading the plugin and enabling automatic updates. • There is no difference in which LAN port (1 or 2) is used for connecting the controller and the router; the ports are interchangeable.
System preparation	• Ensure that the IHC system wiring has been correctly transferred to the **SpaceLogic IHC Interface**. • The **SpaceLogic IHC Interface** must be powered and accessible before starting the pairing process.

Key Features of the IHC Interface





Interface and connectivity	• The connection between the controller and the **SpaceLogic IHC Interface** is established through an Ethernet cable. • 8 IHC input datalines (total of 128 wired inputs) • 16 IHC output datalines (total of 128 wired outputs) • IHC wired devices are connected via existing IHC Input and Output modules, which in turn connect to the input and output datalines on the **IHC Interface**. • Supported functions include: ◦ switching ◦ dimming ◦ blind control
Firmware upgrades	• Firmware upgrades for the **SpaceLogic IHC Interface** can be performed using the **IHC plugin**.
Device integration	• The **SpaceLogic IHC Interface** serves as a bridge between IHC devices and those supported by the controller, such as KNX and Wiser Zigbee devices, through the **KNX Hybrid module** (RS232). See the **KNX Hybrid module** user guide available here.
Bi-directional communication	The system supports bi-directional communication across protocols. For instance: • Control IHC loads from a KNX device (e.g., a KNX Multitouch display). • Send temperature sensor values from IHC to the SpaceLogic KNX valve drive controller. • Control Wiser Zigbee devices (e.g., a ZigBee lamp outlet) from KNX and/or IHC devices. See the **KNX Hybrid module** user guide available here.
Resetting and commissioning	• The **SpaceLogic IHC Interface** is equipped with an external button for resetting and commissioning. Pairing and commissioning with the controller can be initiated by pressing and holding the reset button for at least 1 second and no more than 5 seconds. • To reset the **SpaceLogic IHC Interface**, press and hold the reset button for at least 5 seconds and no more than 10 seconds. • Ensure the device is powered and accessible before starting the pairing process.
Interface communication	• The controller can send and receive messages to and from the **SpaceLogic IHC Interface**, which communicates with the IHC Input and Output modules. A table or list view of connected IHC devices is available in the **IHC plugin**.
Programming interface	• **IHC Plugin** • **eConfigure KNX**

Workflow

This overview shows the sequence of the entire process of commissioning the **SpaceLogic IHC Interface** up to integration with **IHC Plugin**.

1. Connect your **SpaceLogic IHC Interface** to the controller.
2. Login to controller.
3. Download **IHC Plugin**.
4. Open **IHC Plugin**.
5. Pair your **SpaceLogic IHC Interface** with the controller.
6. Import existing IHC project into **IHC Plugin**.
7. Fine-tune the settings in the **Device mapping** tab of the **IHC Plugin**.

IHC Plugin Installation

To begin using the **IHC Plugin**, you must first install it on your controller. The plugin is available through the controller's Marketplace.

Follow the steps below to complete the installation process:

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 **IHC 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 **IHC Plugin** will appear among other applications:
6. Click on the **IHC Plugin** icon to open the plugin:

NOTE: It is recommended to enable the plugin's automatic update to ensure you always have the latest version.

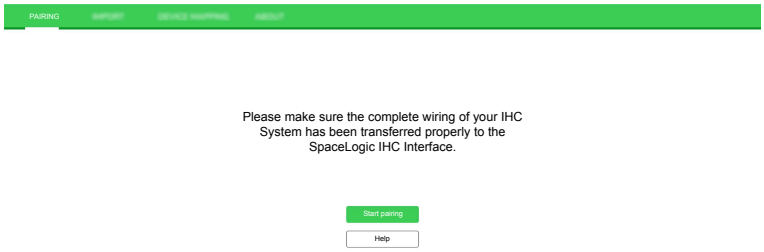
Pairing the IHC Interface with the Controller

To use the **IHC Plugin** for communication between the controller and the **IHC Interface**, you must first pair the **IHC Interface** with the controller.

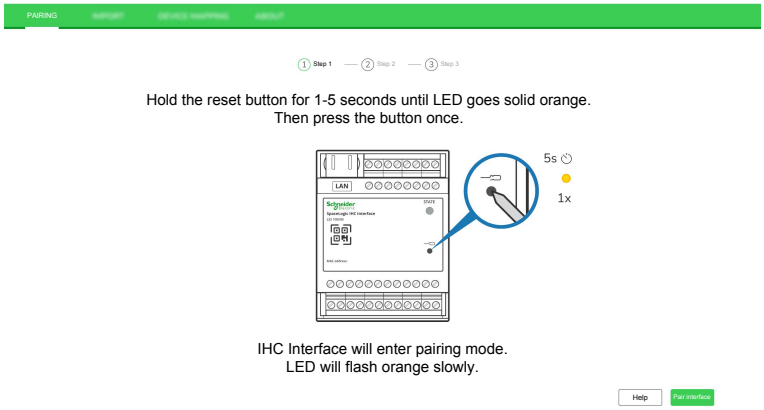
Please make sure the complete wiring of your IHC system has been transferred properly to the **SpaceLogic IHC Interface**.

Follow these steps:

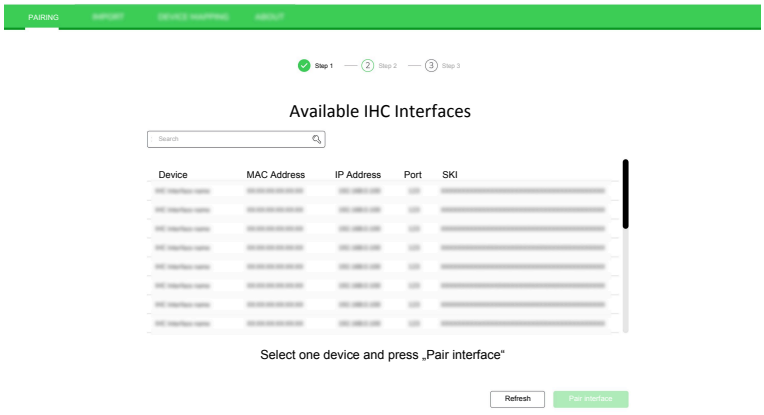
1. Open the **IHC Plugin** and in the **PAIRING** tab, click on the **Start pairing** button to access the pairing wizard.



2. Press and hold the reset button on the **IHC Interface** for 1 to 5 seconds until the LED turns solid orange. Then, press the button once.

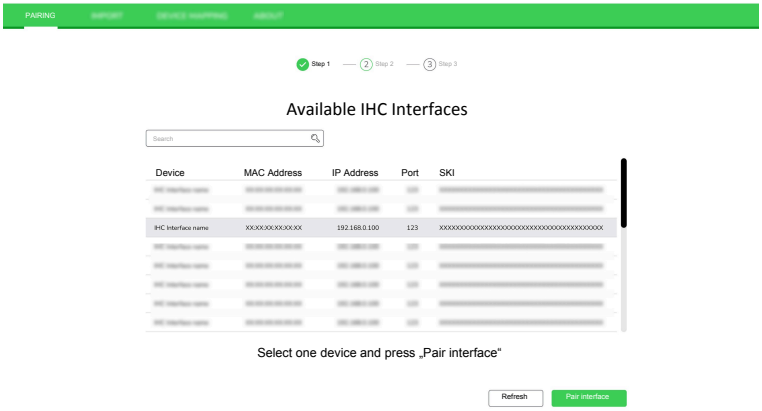


3. Click **Next**.
4. On the wizard's next screen, the available **IHC Interfaces** for pairing will appear.

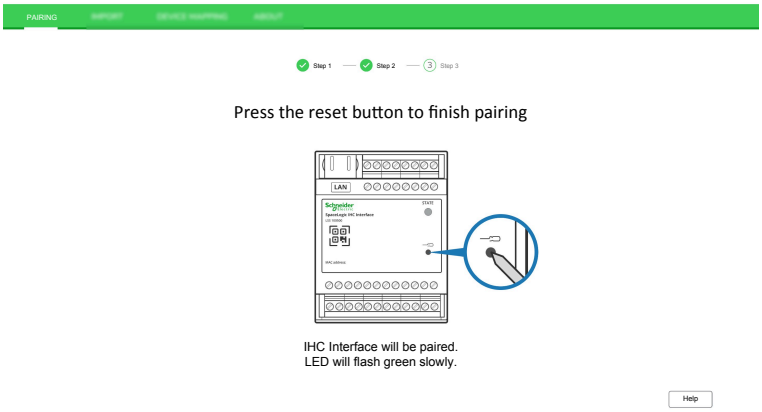


NOTE: If no interfaces appear, click the **Refresh** button.

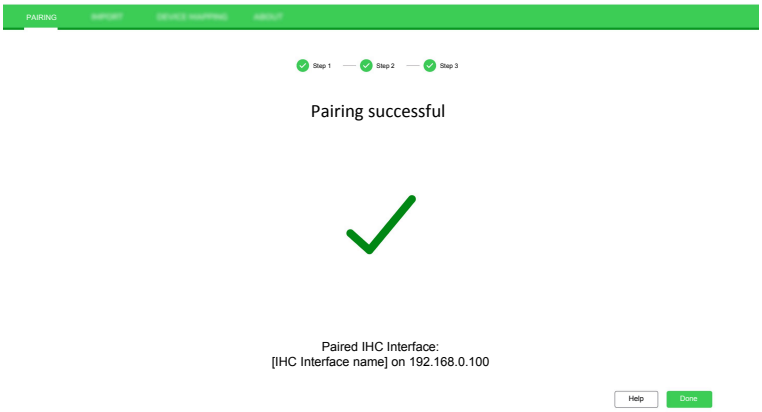
5. Select the interface you want to pair and click the **Pair device** button. The status LED should start blinking orange.



6. Press the reset button on your **IHC Interface** to complete the pairing process. After successful pairing, the **IHC Interface**'s LED will blink three times in green, then the **IHC Interface** will reset itself, causing the LED to flash red a few times.
It will then flash alternately green and red until it successfully connects to the controller.
Once connected, the LED will begin to flash slowly in green, indicating normal operation.

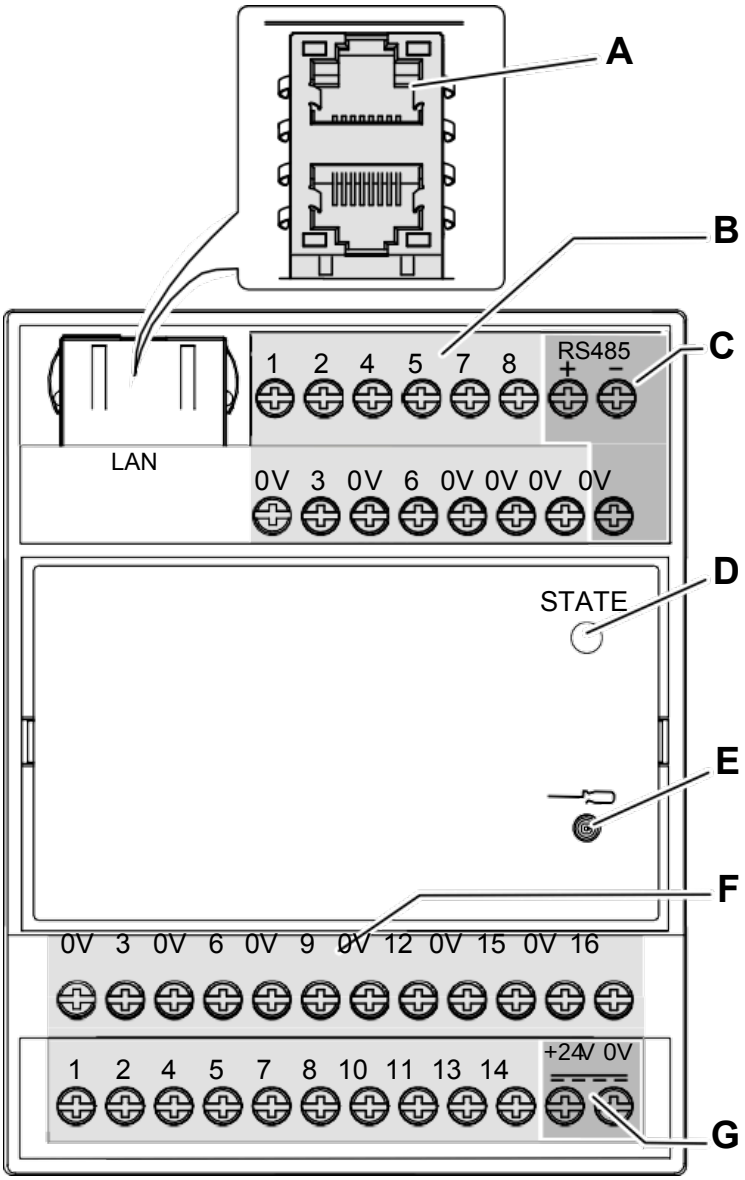


7. Once the message **Pairing successful** appears on the screen, the pairing has been completed successfully. Click the **Done** button.



NOTE: If the pairing fails, try repeating the entire process. For assistance, click the **Help** button at the screen's bottom right corner.

Operating elements, terminals



A 2x RJ45

B Inputs 1 ...8

C RS-485 +/-/0 V, SELV

D LED for status indication

E Reset button: for device reset, factory reset, initialization of pairing mode

F Outputs 1 ... 16

G Power Supply DC 24 V

LED for Status Indication

The **STATE LED (D)** indicates the following conditions:

Operating		Meaning
● ● ●...	Green, flashing	Normal operation

Power		Meaning
	Red, flashing	No link available
	Red/Orange, flashing	Waiting for DHCP
	Red/Green, flashing	Connecting to W4K controller

Pairing		Meaning
	Orange	You have held down the reset button (E) for 1 to 5 seconds until the LED lights up orange.
	Orange, flashing	After pressing the reset button (E) again, the LED starts flashing orange.
	Orange, flashing quickly	After selecting the module you want to pair, the LED flashes orange quickly.
	Green, flashing 3 times	Pairing was successful.
	Red, flashing 3 times	Pairing was not successful or a timeout occurred during the pairing process.

Reset		Meaning
	Red	Device reset is in progress. You have held down the reset button (E) for 5 to 10 seconds until the LED lights up red.
	Red, flashing	Factory reset in progress. You have held the reset button (E) for at Red, flashing least 10 seconds and the again for 10 seconds. After releasing the reset button (E), the LED flashes red.

Device and Factory Reset

If your **IHC Interface** does not respond as expected, you can perform a reset. There are two types of reset: device reset and factory reset.

Device Reset

A device reset reboots the **IHC Interface** without erasing configuration or commissioning data. This is useful for resolving temporary issues while keeping all settings intact.

To perform a device reset:

1. Press and hold the **reset button (E)** for **5 to 10 seconds** until the **STATE LED** lights up **red**.
2. Release the button. The device will restart and resume normal operation.

Factory Reset

A factory reset restores the **IHC Interface** to its original factory settings. All configuration data, commissioning information, and logs are erased. After a factory reset, you must reconfigure and pair the device again.

To perform a factory reset:

1. Press and hold the **reset button (E)** for **at least 10 seconds**, then press and hold it again for **10 seconds**.
2. Release the button. The **STATE LED** will flash **red**, indicating that the factory reset is in progress.

Important Notes

- Use **device reset** for troubleshooting without losing configuration.
- Use **factory reset** only when reconfiguration is required or if pairing cannot be restored.
- After a factory reset, default network settings (DHCP enabled) are applied.

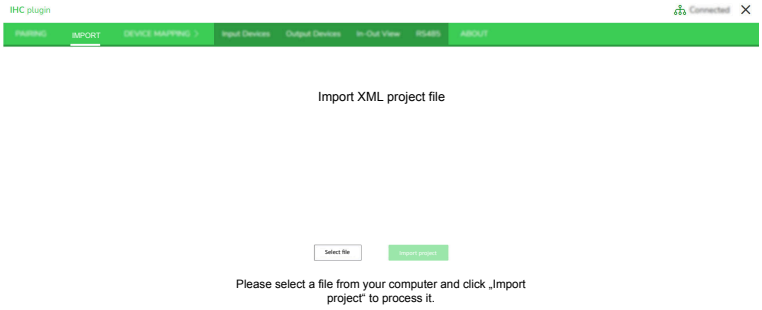
Import Existing IHC Project

If you already have a project for your IHC installation, you can import device settings into the **IHC Plugin** instead of starting from scratch.

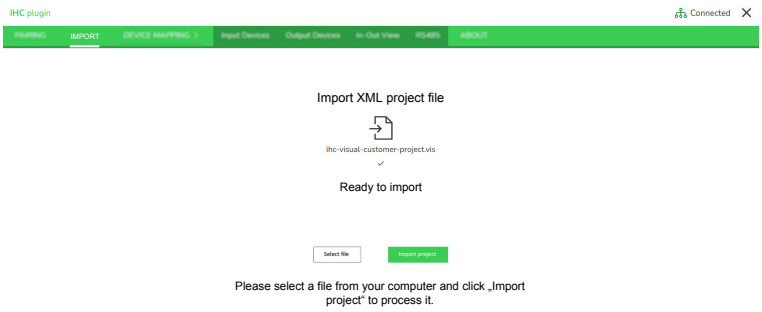
Please note that importing project data will delete current device mappings in the **Input devices**, **Output devices**, **In-Out View** and **RS485**. This action cannot be undone.

To import your project, follow these steps:

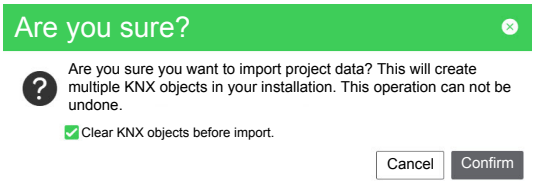
1. In the **IHC Plugin**, click on the **Import** tab.



2. A wizard will start to guide you through the entire process.
3. In the next step, click the **Select file** button, choose the file you want to import from your computer, and click **Open**.
4. The selected file will begin uploading. Once the upload is complete, click the **Import project** button.



5. If you previously imported a project, you can choose what will happen to previously created objects. You can keep them or delete them (**Clear KNX objects before export**).



6. If you are sure you want to import the project, click **Confirm**.
7. Once the project is successfully imported, you can fine-tune the settings in the **Device mapping** tab.

During the import, multiple KNX objects (with virtual KNX group addresses) will be created. These objects will be tagged with the label “`import`" and can be removed if needed.

To delete them:

1. Open the Start page in the controller’s web interface (you must be logged in).
2. Open the **Configurator**.
3. Navigate to the **Objects** screen.
4. In the **Tags** field, type "import" and click **Apply filter**.
5. Click **Mass delete**.
6. Confirm by selecting **Delete objects from current filter**.

Limitations of Import

While the import feature is helpful for configuring IHC inputs, outputs, and LED dimmers, and it automatically creates KNX group objects mapped to these elements, there are some important limitations to be aware of:

- **Function blocks (logic)** from the original IHC project **cannot be transferred**.
- **Buttons** are imported as **plain inputs**, which means you will need to manually configure their behavior after the import.

Device Mapping

The **DEVICE MAPPING** tab in the **IHC Plugin** interface provides a centralized view for managing the connection between your IHC devices and the controller. This section is essential for assigning and organizing how input and output signals from the IHC system are interpreted and controlled within the KNX environment.

The tab is divided into three sub-sections:

- **Input devices**: For mapping IHC input device addresses to corresponding KNX group addresses for devices such as push buttons, sensors, and binary inputs.
- **Output devices**: For mapping of KNX group addresses to corresponding IHC output device addresses for devices like lights, fans, or other actuators.
- **In-Out view**For visualizing and editing the logical connections between input and output devices in a single, integrated interface.
- **RS485** (Dimmers): For configuring dimming devices connected via the RS485 interface, such as 2-channel LED dimmers.

Each sub-tab allows you to view and edit connected devices, assign names, configure port mappings, and ensure proper communication between the IHC system and the controller. This structured approach simplifies commissioning and ensures a seamless integration of IHC devices into your smart home or building automation setup.

Understanding Standard/Virtual KNX Group Addresses in the Controller

The controller uses two types of KNX group addresses to manage communication:

- Standard KNX group addresses
- Virtual KNX group addresses

Understanding the difference between these types helps determine which to use based on your system configuration.

Choosing Between Standard and Virtual Group Addresses

If you **have** real KNX devices:

- Use **standard** KNX group addresses for communication with commissioned KNX devices.
- Use **virtual** KNX group addresses for all other purposes.

If you **do not have** KNX devices:

- **Use only virtual** KNX group addresses.
- Do not use standard KNX group addresses, as this may cause bus issues and overload the controller.

Standard KNX Group Addresses

Standard KNX group addresses are used to communicate with **real KNX devices** connected to the KNX bus. These devices must be properly configured and commissioned using ETS.

Format

A standard KNX group address is a **16-bit integer**, represented in the format:

a/b/c – where each part is a decimal number separated by slashes:

	Size (bits)	Range
a	5	0 – 31
b	3	0 – 7
c	8	0 – 255

The address 0/0/0 is not allowed.

However, addresses with zeroes in one or two components are valid, such as:

- 0/0/1
- 2/0/0
- 0/3/0
- 0/4/5
- 6/0/7
- 8/1/0

Address Range

- Lowest valid address: 0/0/1
- Highest valid address: 31/7/255
- Total possible addresses: 65 535 (32 * 8 * 256 – 1 = 65 535)

Virtual KNX Group Addresses

Virtual KNX group addresses are used for internal communication within the controller and with the **IHC interface** and **Hybrid module**.

Format

A virtual group address is a **17-bit integer**, with the most significant bit always set to 1, represented in a three-part format:

a/b/c – where each part is a decimal number separated by slashes.

Virtual addresses use the same format as standard addresses but with a different range for the first component.

	Size (bits)	Range
a	6	32 – 63 (instead of 0 – 31)
b	3	0 – 7
c	8	0 – 255

Addresses with zeroes in one or two components are valid, such as:

- 32/0/0
- 63/1/0
- 42/0/7

Address Range

- Lowest valid address: 32/0/0
- Highest valid address: 63/7/255
- Total possible addresses: 65 536 (32 * 8 * 256 = 65 536)

Creating KNX Objects in the Controller

To use KNX group addresses, you must first create an object.

Detailed instructions are available in the **Configurator – Main Page** > **Objects** > **Add New Object** chapter of the Controller User Guide. This chapter also includes additional helpful information.

Steps to Create an Object

1. Navigate to: Start page of your controller > **Configurator** > **Objects** tab.
2. Click **Add New Object**.
3. Choose one of the following options:
   - **Create standard object**
   - **Create virtual object**

This opens the **Add New Object** dialog.

Group Address Field

- The **Group address** field will be automatically filled with the next available KNX group address (standard or virtual, depending on your selection).
- You can accept the suggested address or enter a custom one.
- The address must be unique. If already in use:
  - The field is underlined in red.
  - A red circle icon appears next to the field.
  - Hovering over the icon displays: **This address is already in use.**. Choose a different address and try again.

  NOTE: The group address cannot be changed later. To use a different address, create a new object.

Required Fields

Three mandatory fields must be filled:

- **Group address**
- **Object name**
- **Data type**

Object Name

- Must be unique.

- If already in use:
  - The field is underlined in red.
  - A red circle icon appears.
  - Hovering over the icon shows: **This name is already in use.**. Choose a different name and try again.
- The object name can be changed later.

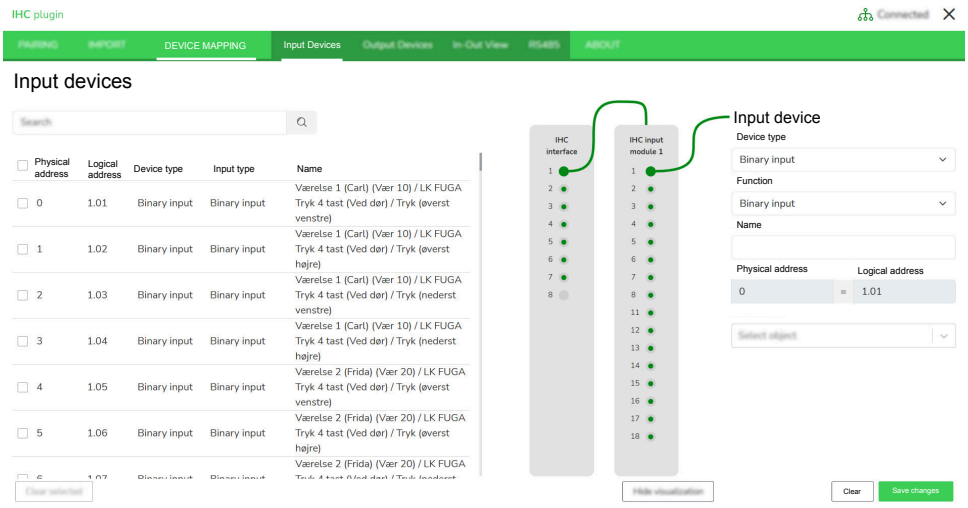
Data Type

- If unsure, select a data type that seems appropriate.
- The data type can be changed later.

Input Devices

The **Input devices** sub-tab allows you to configure and manage all IHC input devices connected to the system. This interface provides a visual and intuitive way to map physical IHC inputs to input functions implemented in the controller and to KNX group addresses (real or virtual).

Layout Overview



- **Left panel**: Displays a list of all imported or manually added IHC input device mappings. You can associate devices with a input module and select it for detailed configuration.
- **Center panel**: Shows a graphical representation of the IHC input module and the **IHC Interface**. This visual helps you understand how devices are physically connected and routed through the system.

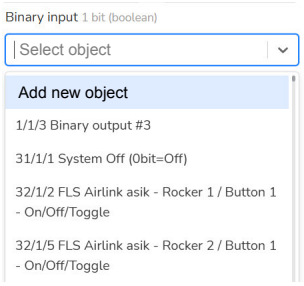
  NOTE: To save space, you can hide the center panel using the button **Hide visualization** under it.

  Right panel (device details): When you select a device from the list, its configuration details are shown here. This is also where you can adjust the device settings as needed. You can see the details as follows:

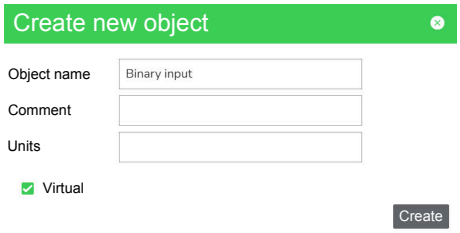
Device Details

1. **Device type** (Drop-down menu)
   Select the type of IHC input device. Supported types include:
   - Binary input
   - Push button
   - PIR sensor
   - Temperature sensor
2. **Function** (based on the selected device type):
   Choose the intended function like **Room temperature + Humidity**.
3. **Name**
   Assign a custom name to the device for easier identification in the project.
4. **Physical address** (Read only)
   This is an internally calculated address (range: 0 – 127) representing the device's physical position in the IHC system.
5. **Logical address** (Read only)
   Indicates how the signal is routed:
   - From the device to the **input module**
   - From the **input module** to the **IHC Interface**

6. **Mappings and attributes** Depending on the selected function, you can define one or more mappings and configure additional attributes such as:
   - **Select object**
     This option allows you to associate the IHC input device with a KNX object.
     You can select an already created object or you can create a new one.



Once you click on **Add new object**, new window will appear.



Here you can set the name of the object, comments, units, and specify whether it is a **Virtual** KNX object.

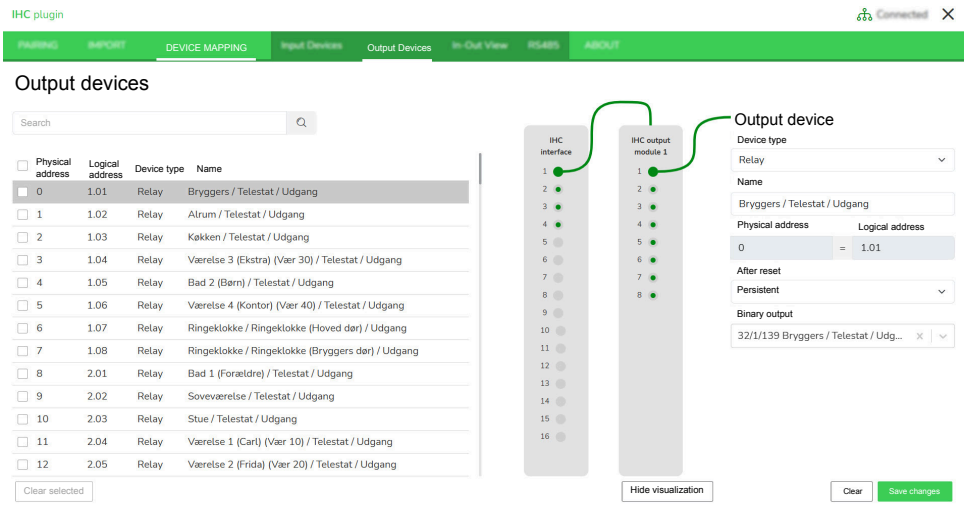
   - **Status KNX group address**: This address is used to report the current state of a device – typically whether it is on or off.
   - **Main KNX group address**: Defines the primary communication channel between the IHC device and the KNX bus.
   - **Additional KNX group addresses**: This field is used to configure supplementary communication channels for a device (e.g., for feedback, dimming level, etc.).
     A temperature sensor might use additional addresses to report humidity or floor temperature.
   - **Min. sending time**: This configuration parameter defines the minimum interval between two consecutive transmissions of the same value from an IHC input device to the KNX system.
     If a temperature sensor is configured with a minimum sending time of 10 seconds, it will not send the same temperature value more than once within that 10-second window – even if the value remains unchanged.
   - **Temperature delta**: Parameter that defines the minimum change in temperature that must occur before a new value is sent to the KNX system.
     If set to 0.5 °C, the sensor will only send a new temperature value if the current reading differs from the last sent value by at least 0.5°C.

Once you have completed the configuration of an input device, make sure to click **Save changes** to apply and store your settings. This ensures that your mappings are correctly registered and ready for integration with the rest of the system.

Output Devices

The **Output devices** sub-tab is used to configure and manage all IHC output devices connected to the system. This section allows you to define how each output behaves and how it integrates with the KNX network.

Layout Overview

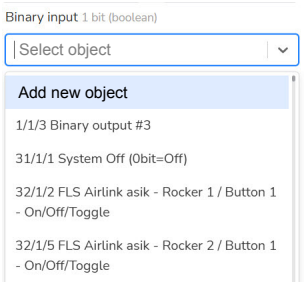


- **Left panel**: Displays a list of all imported or manually added IHC output device mappings. Each device is linked to a specific output module and can be selected for detailed configuration.
- **Center panel**: Shows a graphical representation of the IHC output module and the **IHC Interface**, helping you visualize how outputs are physically connected and routed.
- **Right panel (device details)**: When a device is selected from the list, its configuration options appear here. You can edit the following fields:

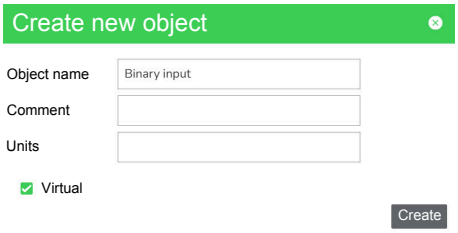
Device Details

1. **Device type**: Relay.
2. **Name**
   Assign a custom name to the output device for easier identification in your project.
3. **Physical address**
   Internally calculated address (range: 0 – 127) representing the device's physical position in the IHC system.
4. **Logical address**
   Indicates the routing of the signal from the **IHC Interface** to the output module.
5. **After reset** (**On**/**Off**/**Persistent**)
   Defines the default state of the output after a system reset.

6. **Select object**
   This option allows you to associate the IHC input device with a KNX object. You can select an already created object or you can create a new one.



   Once you click on **Add new object**, new window will appear.

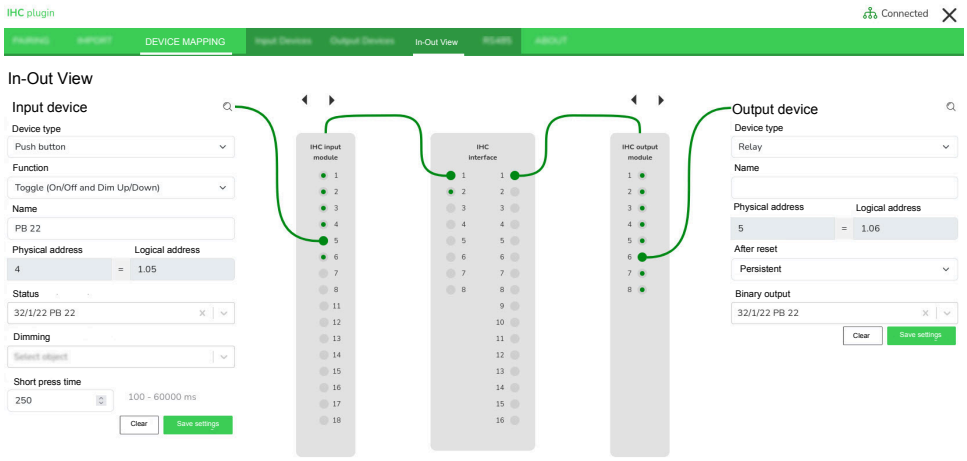


   Here you can set the name of the object, comments, units, and specify whether it is a **Virtual** KNX object.

Once you have configured an output device, click **Save changes** to apply and store your settings.

In-Out View

The **In-Out view** sub-tab provides a comprehensive graphical interface for managing the logical relationships between IHC input and output devices. It is designed to give installers a clear overview of how signals flow through the system.



- **Left panel**: Displays editable details of the selected **input device**, including fields such as **Device type**, **Function**, **Name**, and addressing.
- **Center panel**: Shows a visual representation of the **input module**, **IHC Interface**, and **output module**. Inputs and outputs are graphically linked, allowing you to easily trace and manage signal paths.
- **Right panel**: Displays editable details of the selected **output device**, including configuration fields such as **Device type**, **Name**, addressing, and reset behavior.

Both input and output devices include a **Save changes** button at the bottom of their respective panels.

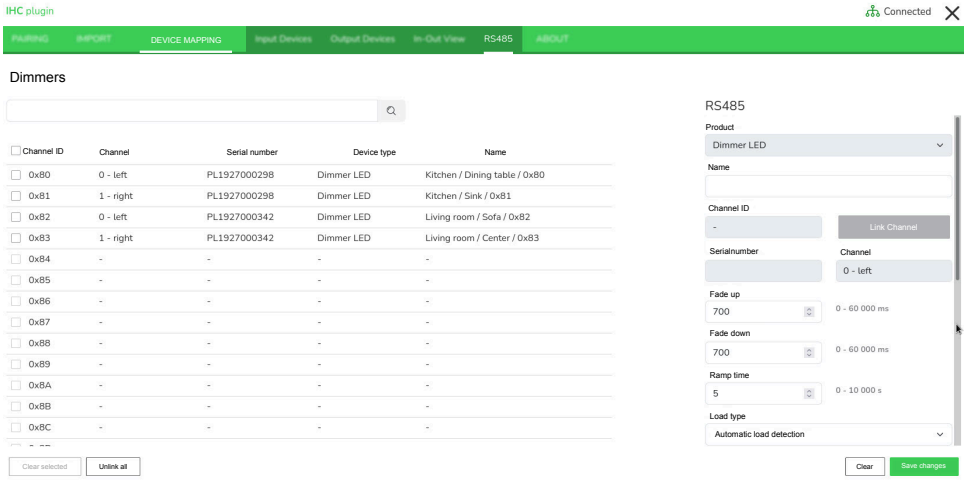
Be sure to save your changes after editing to ensure all mappings are applied correctly.

RS485

The **RS485** sub-tab is dedicated to configuring **LED dimmers** connected via the RS485 interface.

This section allows precise control over dimming behavior and integration with KNX group addresses.

Layout Overview



- **Left panel**: Displays a list of all connected RS485 dimmers, including channel ID, channel position (left/right), serial number, device type, and name. You can select one or multiple dimmers for configuration or unlinking.
- **Right panel (dimmer details)**: When a dimmer is selected, its configuration options appear here, including editable fields and action buttons.

Additional controls:

- **Search bar**: Quickly find a dimmer by name or ID.
- **Clear selected**: Deselects all currently selected dimmers.
- **Unlink all**: Removes all linked channels from the selected dimmers in one action.

Linking the Dimmer to a Channel

First you need to link the dimmer to one of the channels in the left panel as follows:

1. Select the dimmer in the left panel and click **Link Channel**.
2. The **Dimmer linking** window will appear with guided instruction.
3. Press and hold both buttons labeled 1 and 2 on the dimmer.
4. LEDs will start blinking blue.
5. Press one button to link corresponding channel ID (0 or 1) and click **Link**.
6. Optionally, select **Link both channels at once** to save time.
7. When a dimmer is linked, its configuration options appear in the right panel.

IMPORTANT:

- After linking, the **Link Channel** button changes to **Unlink Channel** when you click on a linked dimmer in the left panel.
- Use **Unlink all** in the bottom left corner to remove all linked channels from multiple selected dimmers at once.

Dimmer Configuration

The following fields are available:

1. **Product**
   Fixed as **Dimmer LED**. This field is not editable.
2. **Name**
   Assign a custom name to the dimmer for easy identification.
3. **Channel ID**
   Unique identifier for the dimmer channel.
4. **Link/Unlink Channel**
   A button to link/unlink this channel.
5. **Serial number**
   Displays the serial number of the dimmer.
6. **Channel**
   Indicates the channel number (up to 64) and its physical position (left or right) on the dimmer.
7. **Fade up / Fade down** (in milliseconds)
   Defines the time it takes for the light to fade up or down when dimming is triggered.
8. **Ramp time** (in seconds)
   Sets the duration for transitioning between dimming levels.
9. **Load type**
   Specifies the type of load connected:
   - **Automatic load detection**
   - **Resistive-capacitive load**
   - **Resistive-inductive load**

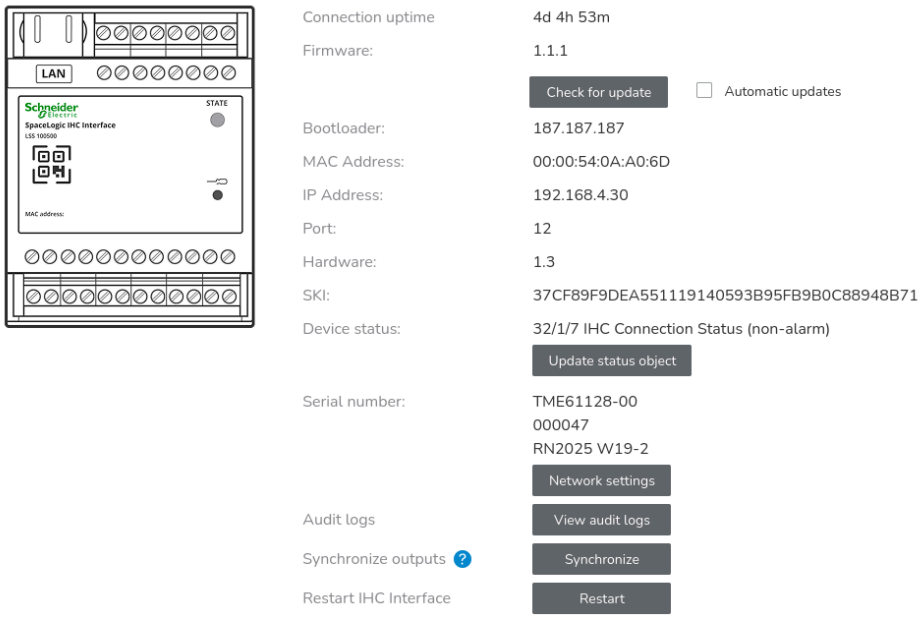
   TIP: If you are unsure, starting with **Automatic load detection** is recommended.
10. **Max. level / Min. level** (in %)
    Defines the upper and lower brightness limits for the dimmer.
11. **Select objects**
    Each dimmer can be linked to multiple KNX group objects for full integration. You can select an already created object or create a new one for each of the following:
    - **Status**: KNX group object for on/off status.
    - **Dimming**: KNX group object for dimming control (0 – 100 %).
    - **Level**: KNX group object for setting brightness level.
    - **Level feedback**: KNX group object for reporting current brightness.
    - **Status feedback**: KNX group object for reporting on/off state.

Once all parameters are configured, click **Save changes** to apply and store the settings.

This ensures the dimmer is correctly mapped and fully functional within the KNX system.

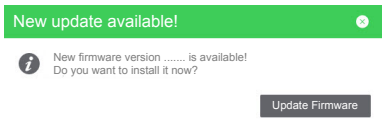
About

The **ABOUT** tab in the **IHC Plugin** provides detailed information about the connected **IHC Interface**. It includes hardware and network parameters, firmware update options, device status details, audit logs, output synchronization, and maintenance actions such as restarting the interface.



Displayed Information

- **Connection uptime**: Shows how long the **IHC Interface** has been continuously connected to the controller.
- **Firmware**: Displays the current firmware version installed on the **IHC Interface**.
  - You can check for a newer version by clicking **Check for update**.
  - If a new version is available, a dialog titled **New update available!** will appear.

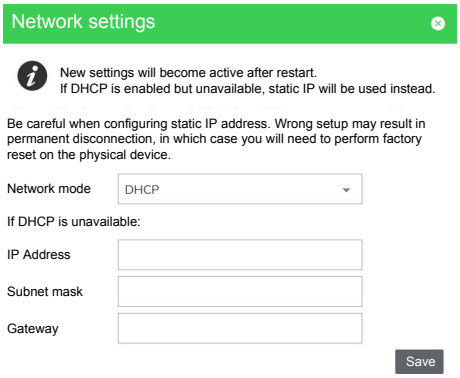


  - Click **Update firmware** to install the latest version.
  - If the update fails, try again later.
  - Enabling **Automatic updates** ensures the device updates itself whenever new firmware becomes available.
- **Bootloader**: Shows the version of the bootloader used during device startup.
- **MAC Address**: The unique hardware address of the device's network interface.
- **IP Address**: The current IP address assigned to the **IHC Interface**.
- **Port**: The communication port used by the plugin to communicate with the device.
- **Hardware**: The hardware version of the **IHC Interface**.
- **SKI**: The **Subject Key Identifier**, a unique identifier used in secure communication (e.g., for encryption or authentication).

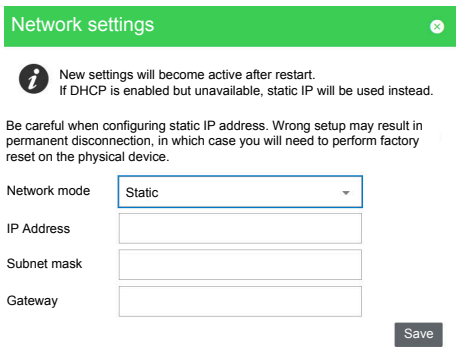
- **Device status**: This is a KNX communication object that reflects the online/offline status of the **IHC Interface**:
  - **1 =** the interface is online and connected
  - **0 =** the interface is offline or disconnected
  - The number shown (e.g., “32/1/7”) is the KNX group address and the object name assigned in the controller. You can configure the object’s name and group address in the **Configurator** > **Objects** tab of the controller.
- **Update status object**: Opens the standard KNX object selection dialog (the same as in Input/Output device mapping). This button allows you to:
  - Select a group object for the **Device status**.
  - Update or change the assigned object.
  - Write the current device status into the selected KNX object.
  - This is useful when manually synchronizing the KNX value with the actual online status of the interface.
- **Serial number**: The unique serial number of the **IHC Interface**.

Network Settings

- **Network Mode**: Defines how the IP address is assigned. Options include:
  - **DHCP**: The IP address is automatically assigned by the network.



  - **Static IP**: A fixed IP address is manually configured.



- **Static IP Address**: The manually assigned IP address used when DHCP is not available.
- **Subnet Mask**: Defines the network’s size and range.
- **Gateway**: The IP address of the router or gateway used to access external networks.

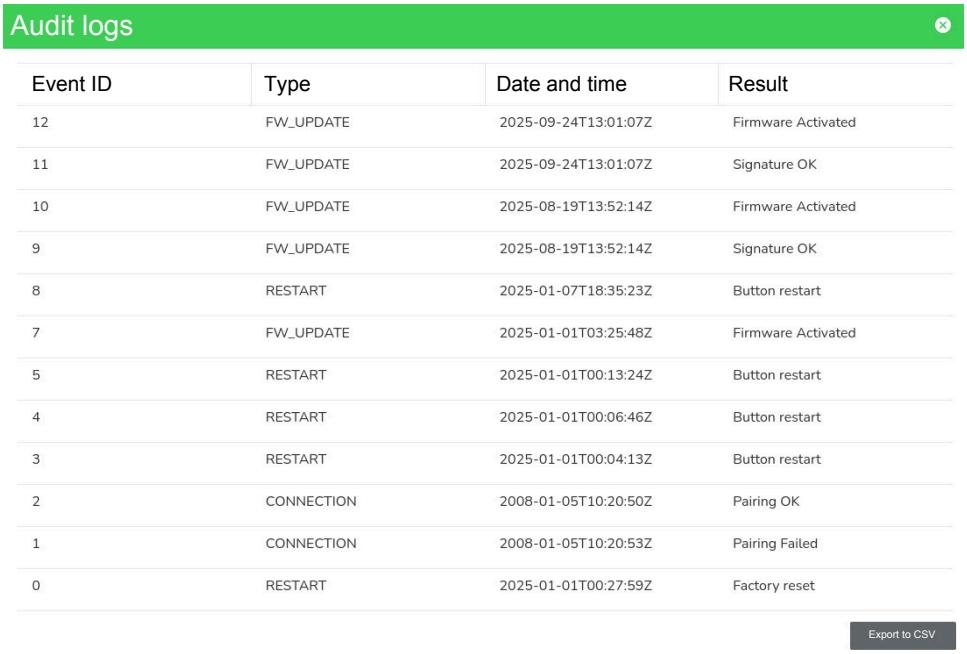
NOTE: If **DHCP** is enabled but unavailable, the system will automatically fall back to the **Static IP** configuration.

Audit Logs

The **View audit logs** feature provides a detailed record of system events for troubleshooting and monitoring purposes.

Displayed Information in audit logs:

- **Event ID**: A unique identifier for each logged event.
- **Type**: The category of the event (e.g., FW_UPDATE, RESTART, CONNECTION).
- **Date and Time**: When the event occurred. Audit logs use controller time instead of UTC.
- **Result**: The outcome of the event (e.g., Firmware Activated, Signature OK, Button restart, Pairing OK, Factory reset).



Available actions:

- **Export to CSV**: Allows you to download the audit log data for external analysis or record-keeping.

This functionality helps track firmware updates, restarts, connection attempts, and other critical actions, ensuring transparency and aiding in diagnostics.

Synchronize Outputs

The **Synchronize** button attempts to read the **actual physical state** of all IHC outputs and synchronize them with the configured KNX group objects.

This ensures the KNX objects reflect the true current state of the outputs.

This synchronization happens automatically whenever the device reconnects, but the button allows you to trigger it manually when needed.

Typical use cases:

- after restoring communication,
- after editing output mapping,
- after manual override on the IHC modules,
- when KNX values do not match physical output states.

Restart IHC Interface

The **Restart** button reboots the **IHC Interface** without deleting configuration or pairing. This is useful for resolving temporary communication issues or applying internal updates that require a restart.

Use Cases

This chapter presents practical examples of how to configure and use the **IHC Plugin**, focusing on LED dimmer setup, toggle button functionality, feedback integration, scripting, staircase timer functionality, and scene control.

LED Dimmer Configuration

You can configure LED dimmers either by importing an existing `.vis` project or manually using the **RS485** tab in the **IHC Plugin** web interface.

Importing LED Dimmer Configuration

If you already have an existing IHC project created in `.vis` format, you can import its configuration into the **IHC Plugin**. This allows you to quickly transfer LED dimmer settings without manually recreating each channel. However, only **linked** LED dimmers will be imported, as the channel ID is required to generate valid configuration records.

Before starting the import, make sure your physical LED dimmers are properly configured, linked (i.e. assigned with channel IDs), and connected to the RS485 interface.

Proceed as follows:

1. **Prepare your `.vis` project file**
   Ensure your `.vis` file includes LED dimmer configurations and that the dimmers are already linked and physically connected.
2. **Start the import process**
   Open the **IHC Plugin** and go to the **Import** tab. Follow the wizard to upload your `.vis` project file. Refer to the chapter Import Existing IHC Project, page 18.
3. **Verify linked channels**
   During import, only LED dimmers with valid channel IDs will be included. Unlinked dimmers will be skipped.
4. Automatic group address creation
   The plugin will automatically create all necessary KNX group addresses for the imported dimmer channels. These include objects for status, dimming, level, and feedback.
5. **Review and adjust settings**
   After the import is complete, go to the **RS485** tab to review the imported dimmer channels. You can rename them, adjust parameters, or link additional KNX objects if needed.

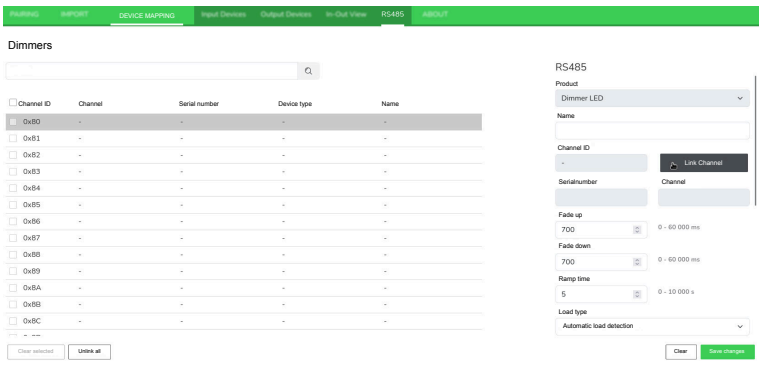
Manual Configuration

If you are not importing a `.vis` project, or if you need to set up new LED dimmers manually, you can configure them from scratch using the **RS485** tab in the **IHC Plugin**. This method gives you full control over channel linking, naming, and KNX group object assignment.

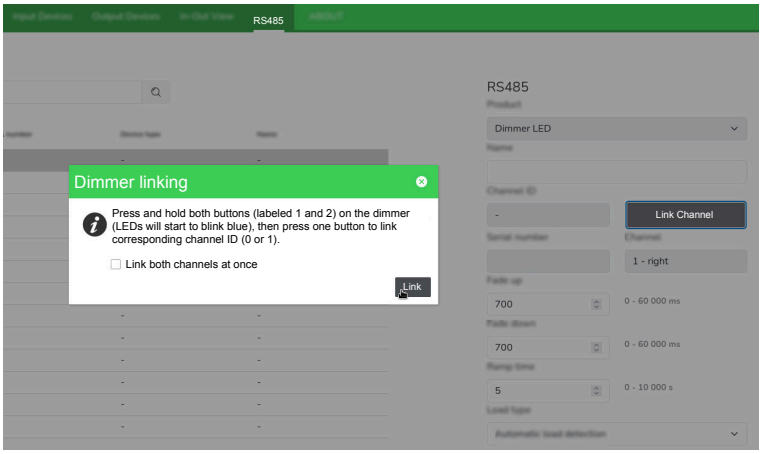
Before starting, it is recommended to clear any existing channel assignments to avoid conflicts.

Proceed as follows to configure LED dimmers from scratch:

1. **Open the RS485 tab**
   In the **IHC Plugin**, navigate to the **RS485** tab to access the LED dimmer configuration.
2. **Clear existing channel assignments**
   If you are reconfiguring an existing installation, it is recommended to use the **Unlink all** button first. This sends a broadcast message to all connected LED dimmers, instructing them to clear their channel IDs. This helps prevent issues caused by duplicate or conflicting channel assignments.
3. **Start linking the first channel**
   Select an unlinked channel from the list and click **Link Channel**.



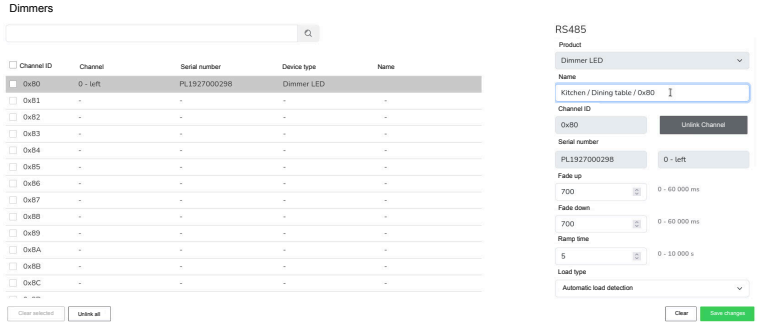
4. **Enter linking mode on the physical dimmer**
   A dialog window will appear, instructing you to press and hold both physical buttons on the LED dimmer until they start blinking. This puts the device into linking mode.



5. **Confirm the channel selection**

Press either the left or right button on the dimmer to confirm which channel you want to link. Click **Link** in the dialog.

6. **Review auto-filled fields**
   The **Channel ID** and group address fields are auto-filled. The **Channel** field will indicate whether you linked the left or right channel.
7. **Name the channel and adjust settings**
   Enter a custom **Name** and adjust additional parameters as needed.

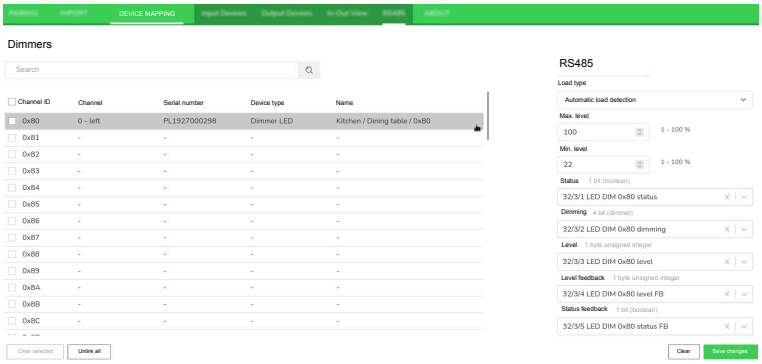


8. **Assign KNX group objects**
   You must assign KNX group objects to enable communication between the dimmer and other devices. You can select existing objects or create new ones using the **Add new object** option.

   Assign the following group objects:

   - **Status**: 1-bit group object used to send ON/OFF commands to the LED dimmer channel.
   - **Dimming**: 4-bit group object used to send dimming commands (e.g., dim up/down).
   - **Level**: 8-bit group object used to send specific brightness levels (0–100%).
   - **Level feedback**: 8-bit group object used to receive brightness level updates from the dimmer.
   - **Status feedback**: 1-bit group object used to receive ON/OFF status updates from the dimmer.

9. **Save the configuration**
   After completing the configuration, click **Save changes**.



10. **Repeat for additional channels**
    Continue configuring other dimmer channels by repeating the linking and setup process.

TIP: You can link both channels of a dimmer at once by checking the **Link both channels at once** option in the **Dimmer linking** dialog.

Checking LED Dimmer Functionality

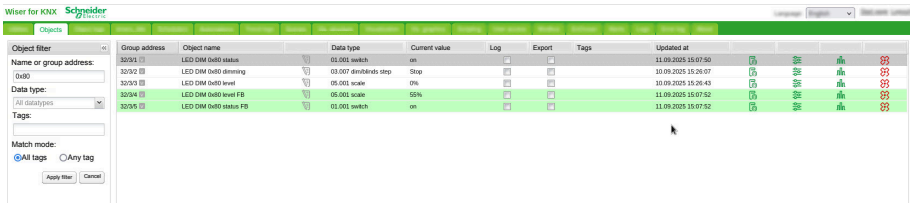
After configuring your LED dimmer channels, it is important to verify that they respond correctly to control commands and provide feedback as expected. You can do this directly from the controller interface by interacting with the KNX group objects assigned to each dimmer.

This process helps confirm that the dimmer is properly linked, that group object mappings are correct, and that feedback is functioning.

To check LED dimmer functionality, proceed as follows:

1. Open the controller´s Start page and navigate to the **Configurator**.
2. Go to the **Objects** tab and filter the list to locate the group objects assigned to your LED dimmer channel.

3. Test the objects: For example, find the Status object and change its value from "off" to "on". The corresponding LED dimmer channel should switch on, and you should see updates in the feedback objects within a few seconds.



NOTE: The plugin polls each dimmer channel every 2 seconds to retrieve the current light level. Channels without feedback objects are skipped. If more than 32 dimmer channels are configured, the polling interval may increase to avoid system overload.

Unlink/Clear Command

When managing LED dimmer channels in the **IHC Plugin**, you may need to remove or reset channel assignments. The plugin provides two distinct commands for this purpose: **Unlink** and **Clear**. Although they may seem similar, they serve different functions and have different effects on the system.

Understanding the difference between these commands is important to avoid accidental loss of configuration or unexpected behavior.

Command	What it does	Result
Unlink	Removes the channel ID from both the **IHC Plugin** configuration and the physical LED dimmer.	The dimmer channel will no longer respond to commands sent to its previous channel ID. Configuration data remains in the plugin.
Clear	Deletes all configuration data related to the dimmer channel from the **IHC Plugin** database.	The channel is fully reset and removed from the system. All settings are erased.

Use **Unlink** when you want to disconnect a dimmer channel but keep its configuration for reference or reuse.

Use **Clear** when you want to completely remove the channel and start fresh.

Advanced Channel Scan

There is a low-level command that allows users to check which channel IDs are already linked. Advanced users can create a script to execute this command:

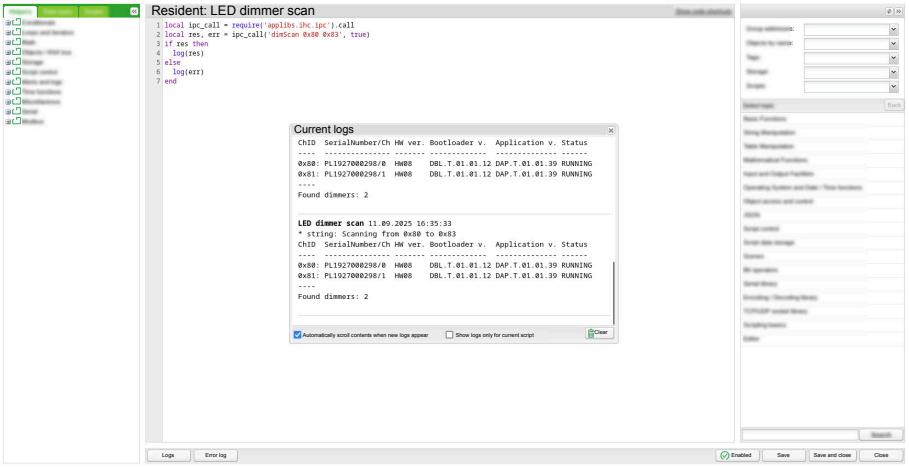
1. In your controller, open the **Configurator** > **Scripting** > **Resident** tab.
2. Add a new script with a 60-second “sleep” interval and the following content:

```
local ipc_call = require('applibs.ihc.ipc').call
local res, err = ipc_call('dimScan 0x80 0x83', true)
if res then
  log(res)
else
  log(err)
end
```

   NOTE: The command `dimScan 0x80 0x83` scans channel IDs in the range from 0x80 to 0x83.

3. Enable your script by clicking the **Disabled** button.

4. Check the **Logs** tab for results.



5. The scan result shows that two channels were found:

```
ChID  SerialNumber/Ch HW ver. Bootloader v.  Application v. Status
----  --------------- ------- -------------  -------------- ------
0x80: PL1927000298/0  HW08    DBL.T.01.01.12 DAP.T.01.01.39 RUNNING
0x81: PL1927000298/1  HW08    DBL.T.01.01.12 DAP.T.01.01.39 RUNNING
----
Found dimmers: 2
```

6. Disable the script afterward to prevent periodic execution.

LED Dimmer Toggle Button Functionality

Toggle buttons can be configured to control LED dimmers, including dimming and feedback via LED diodes.

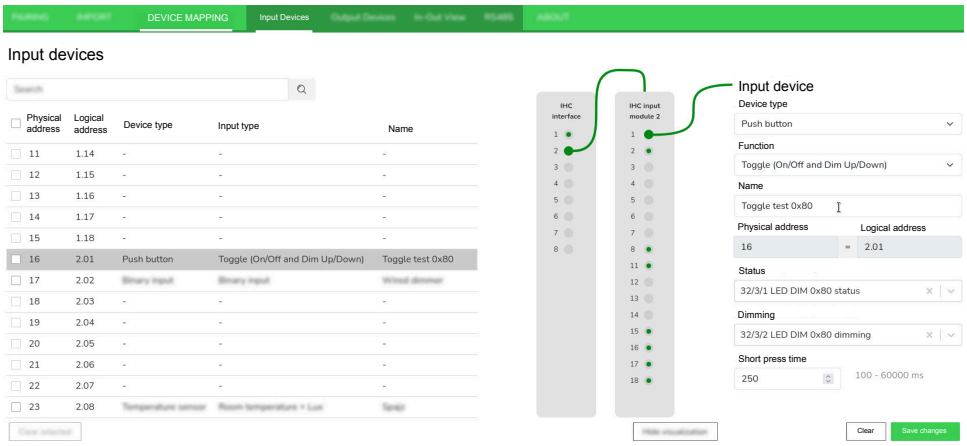
Configuring Binary Input as Toggle Button

In the **IHC Plugin**, binary inputs can be configured to act as toggle buttons that control LED dimmer channels.

> **NOTE:** If you imported a `.vis` project, all dataline inputs are initially configured as plain binary inputs by default. The parse script cannot determine their intended function.

Follow these steps to configure a binary input as a toggle button:

1. Open the **IHC Plugin** interface and go to the **DEVICE MAPPING** section. Select the **Input Devices** tab.
2. Locate the input device you want to configure – use its **Logical address** (corresponding to the input dataline) to identify it.
3. Change the **Device type** and **Function**:
   - Set the **Device type** to Push button.
   - Set the **Function** to Toggle (On/Off and Dim Up/Down).
4. Enter a name for the input device to make it easier to identify in your project.
5. Adjust the **Short press time** if needed. The default value is 250 milliseconds, which is suitable for most toggle button applications.

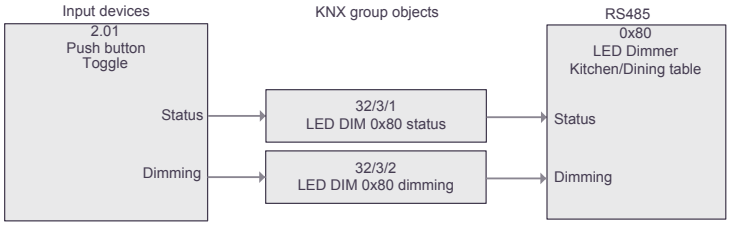


Once configured, the binary input will behave as a toggle button, ready to be linked to an LED dimmer channel using KNX group objects.

Connecting Toggle Button to LED Dimmer

In KNX-based systems, devices communicate using **group addresses**, which are assigned to **group objects** (also called simply “objects”). These group objects act as communication channels between devices.

In the **IHC Plugin**, you create connections between input devices (like toggle buttons) and output devices (like LED dimmers) by assigning them to the same group objects.

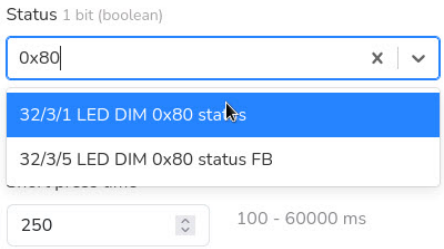


If you have already configured your LED dimmer channel (for example, with channel ID 0x80), you can now link a toggle button to it by following these steps:

1. In the **IHC Plugin**, go to **DEVICE MAPPING** > **Input Devices** and select the input you want to use as a toggle button.
2. In the input device settings, locate the **Status** field.
   Select the same group object that you previously assigned to the **Status** field of the LED dimmer channel. This ensures the button sends commands to the correct dimmer.
3. If you want the button to also control dimming (brightness adjustment), locate the **Dimming** field.
   Select the same group object used in the LED dimmer channel’s **Dimming** field.
4. Click **Save changes** to apply the settings.

The toggle button is now linked to the LED dimmer and should control it as expected.

TIP: The group object selector supports filtering. You can start typing part of the object name (e.g., 0x80) and the list will automatically narrow down to matching objects. This makes it easier to find and select the correct group object.



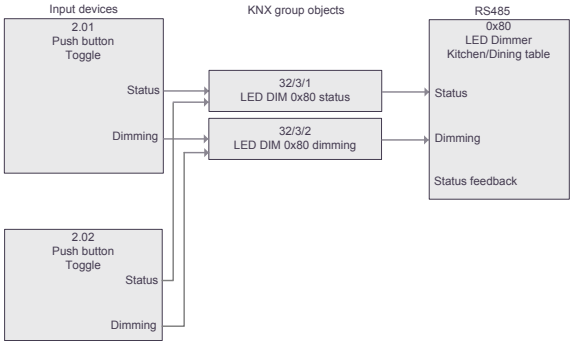
Multiple Toggle Buttons

In the **IHC Plugin**, you can configure multiple toggle buttons to control the same LED dimmer channel by assigning them to the same KNX group objects. Likewise, you can configure one toggle button to control multiple dimmer channels.

This setup is made possible by KNX group addresses, which act as shared communication links between devices. When multiple input devices (toggle buttons) use the same group object, they all send commands to the same output device (LED dimmer).

Follow these steps:

1. **Configure the first toggle button**
   Follow the steps described in Connecting Toggle Button to LED Dimmer, page 39 to configure the first input device as a toggle button and link it to the LED dimmer channel using the appropriate **Status** and **Dimming** group objects.
2. **Configure additional toggle buttons**
   For each additional toggle button:
   - Open the **Input Devices** tab in the **IHC Plugin**.
   - Select the input you want to configure.
   - Set the **Device type** to **Push button** and the **Function** to **Toggle (On/Off and Dim Up/Down)**.
   - Assign the **same group objects** in the **Status** and **Dimming** fields that were used for the first button.
   - Save the changes.



3. **Verify the setup**
   All configured toggle buttons should now control the same LED dimmer channel. Pressing any of them will send commands via the shared group objects.

   TIP: You can also reverse this setup: assign multiple LED dimmer channels to a single toggle button by linking them to the same group objects. This allows one button to control several lights simultaneously.

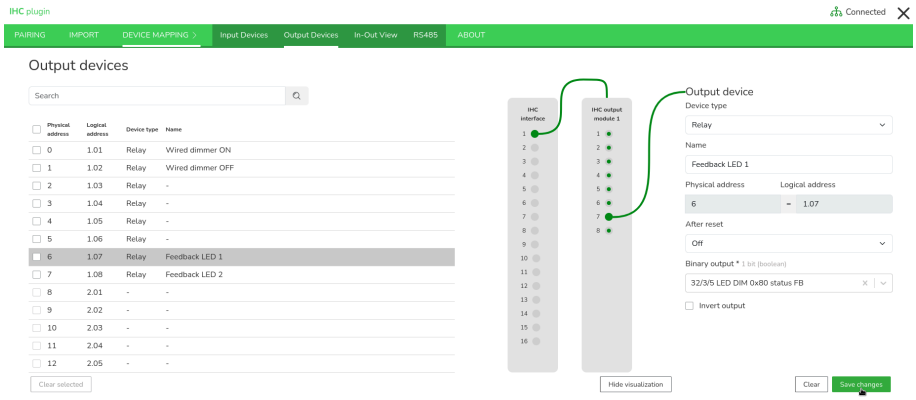
Feedback with LED Diode

To provide visual confirmation that a light or dimmer channel is active, you can connect an LED diode to a digital output on the IHC output 24 module. This diode will light up when the corresponding LED dimmer channel is switched on. The feedback is managed through KNX group objects and can be optimized using a simple event script.

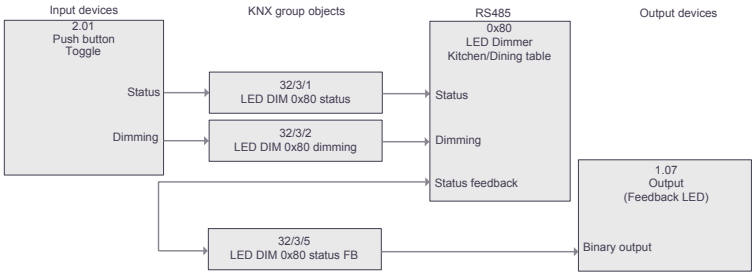
Configuring LED Diode Feedback

Follow these steps:

1. **Connect the LED diode to the output module**
   Physically wire the LED diode to a digital output on the IHC Output 24 module. For example, you can use output dataline **1.07**.
2. **Assign the binary output group object**
   In the IHC Plugin, configure the output device and assign a **Binary output** group object. This group object must be the **same** as the one used in the **Status feedback** field of the corresponding LED dimmer channel.



The connection via KNX group objects is the following:



3. **Save the configuration**
   Apply the changes to ensure the LED diode responds to the dimmer's status updates.

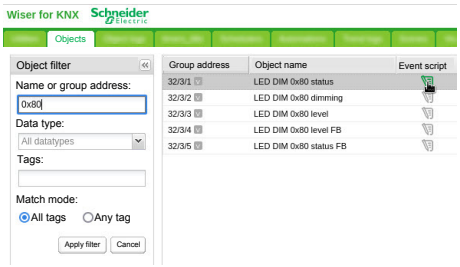
The LED dimmer channel sends feedback every 2 seconds, so the diode may not light up instantly after switching.

NOTE: You can invert the state of the feedback LED diode by selecting the **Invert output** option in the **Output device** configuration. In this case, the LED is switched ON when the LED dimmer channel is OFF, and vice versa. A typical use case for this configuration is when the LED diode is used as a navigation light in a dark room or corridor, where the LED should remain ON while the main light is OFF.

Improving Feedback Responsiveness

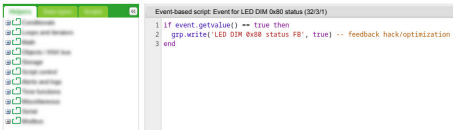
To make the LED diode respond immediately when the dimmer is switched on, you can create a simple event script.

1. **Open the event script editor in your controller**
   Go to the **Configurator** > **Objects** tab and locate the group object assigned to the LED dimmer channel's status.
2. **Create a new event script**
   Click the icon next to the object name to open the event script editor.



3. **Enter the script code**
   Paste the following code into the editor:

```
if event.getvalue() == true then
  grp.write('LED DIM 0x80 status FB', true) -- feedback
hack/optimization
end
```



   NOTE: Make sure the string `LED DIM 0x80 status FB` matches the actual name of your status feedback group object.
4. **Save and enable the script**
   Save the script and ensure it is enabled.

The LED diode will now update immediately when the corresponding dimmer channel is turned on.

Scene Control with LED Dimmers

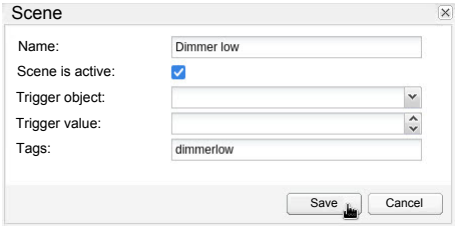
Scene control allows users to manage multiple LED dimmer channels simultaneously by recalling or storing predefined lighting levels. This functionality is integrated into the Wiser for KNX controller firmware and utilizes KNX group objects to execute scene actions.

Configuring Scene Control for LED Dimmers

Scene functionality is part of the Wiser for KNX controller firmware, using KNX group objects to send predefined values to multiple devices simultaneously.

To begin using scene control with LED dimmers, you first need to create and configure scenes in the controller´s **Configurator** plugin.

1. Open the **Configurator** plugin and navigate to the **Scenes** tab.
2. Click **Add scene** in the bottom-left corner.
3. In the **Scene** dialog, enter a unique name for the scene.



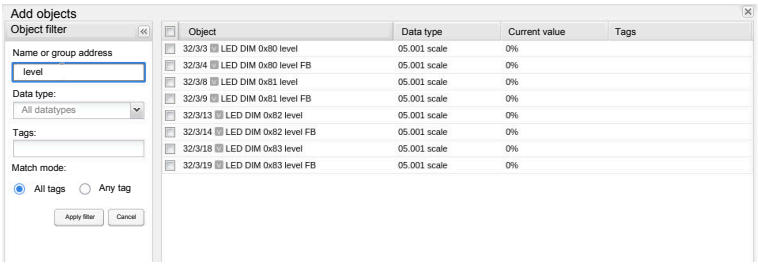
4. Leave **Trigger object** and **Trigger value** fields empty (not used in this setup).
5. Optionally, add **Tags** to organize and filter scenes.
6. Click **Save** to create the scene.

A new scene is successfully created in the Wiser for KNX controller.

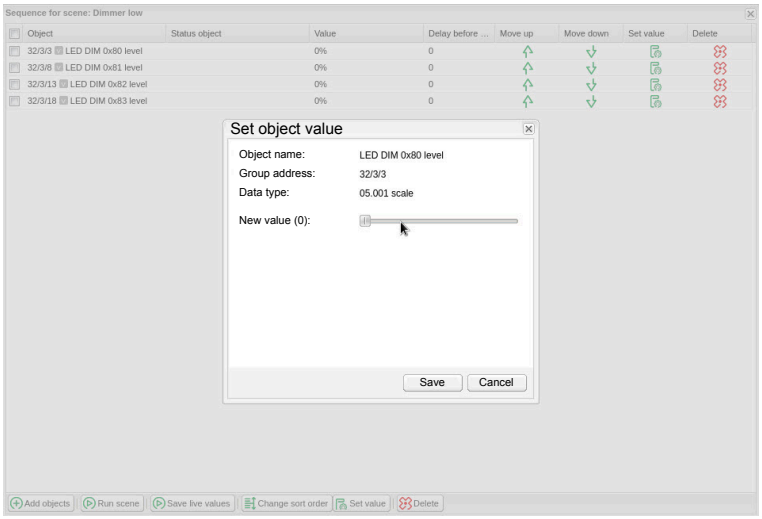
Scene Recall Functionality

To define what happens when a scene is recalled:

1. Click the sequence icon in the **Sequence** column of your scene.
2. Click **Add objects** to include KNX group objects in the scene.
3. The **Add objects** dialog appears, where you select multiple group objects to be part of the scene. Use the filter to locate relevant objects (e.g., those containing “level”).



4. Select all group objects you want to include in the scene (e.g., all LED dimmer level objects) and click **Save**.
5. Assign values to each object: Click the set value icon in the **Set value** column of the value.
6. In the **Set object value** dialog, set value from 0 to 100, where 0 = OFF and 100 = full brightness. Click **Save**.



7. To test the scene, click **Run scene** in the **Sequence for scene** dialog.

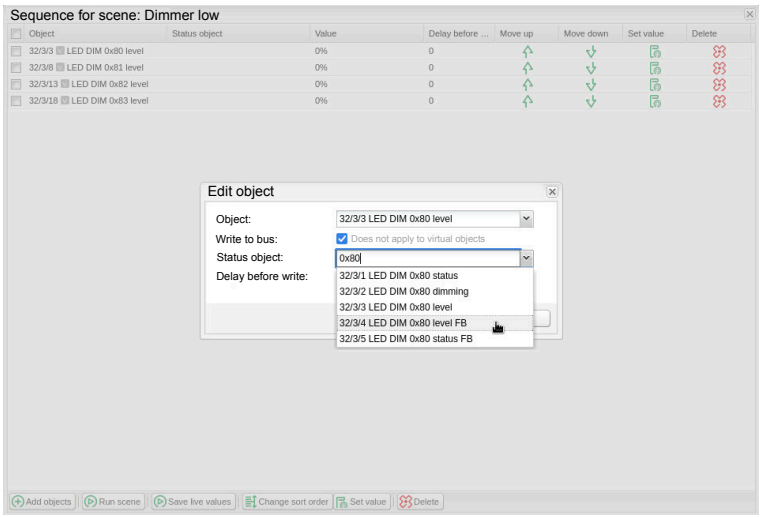
The scene is fully configured to recall predefined brightness levels for selected LED dimmer channels. When the scene is activated, the assigned values are sent to all included KNX group objects, instantly adjusting the lighting to the desired state.

Scene Store Functionality

Scene store functionality allows saving the current LED Dimmer levels into a scene.

To enable this:

1. Click the sequence icon in the **Sequence** column of your scene.
2. For each object in the scene sequence you want to edit, click its row in the **Sequence for scene** dialog.
3. In the **Edit object** dialog, select the appropriate level feedback object in the **Status object** field. Use filtering to find the correct object.



4. Click **Save** to apply changes.
5. Repeat for all objects in the sequence.

The scene is prepared to store the current brightness levels of the selected LED dimmer channels. When the store command is triggered, the scene will capture and save the real-time values from the assigned feedback objects, allowing accurate recall later.

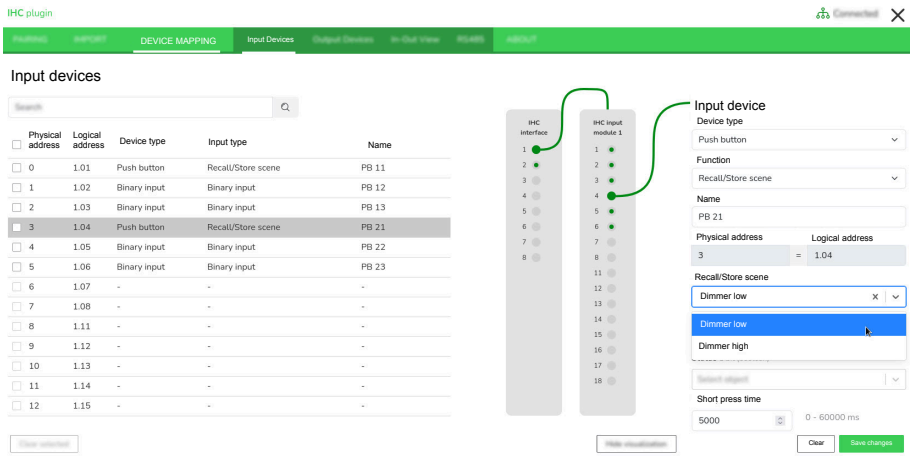
Configuring a Push Button for Scene Recall/Store

To enable physical control of scenes via push buttons, you can configure input devices in the **IHC Plugin** to recall or store scenes based on the duration of a button press.

To configure a push button for scene control:

1. Open the **IHC Plugin**. If the plugin is already open, close and reopen it to ensure the latest list of scenes is loaded.
2. Navigate to **DEVICE MAPPING** > **Input Devices** tab.
3. Choose the input you want to configure based on its **Logical address** (e.g., input dataline).
4. Change **Device type** to Push button.
5. Set **Function** to **Recal/Store scene**.

6. In the **Recal/Store scene** field, select the specific scene you created earlier.



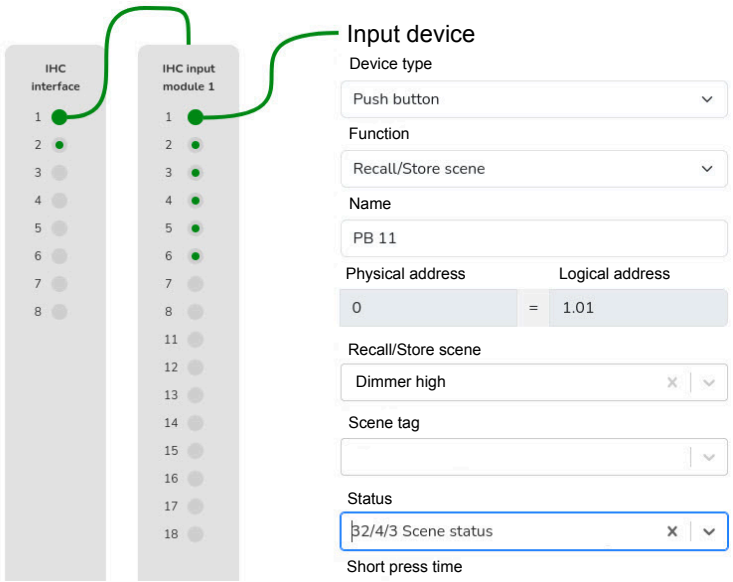
7. Alternatively, use the **Scene tag** field to recall all scenes sharing the same tag.
8. Define **Short press** time: Sets the threshold for distinguishing between a short and long press.
   Default value is 5000 milliseconds (5 seconds):
   - Short press (< 5s) → recalls the scene.
   - Long press (≥ 5s) → stores current levels into the scene.
9. Click **Save changes** to apply the setup.

The push button will be able to recall a scene with a short press and store current LED dimmer levels with a long press.

Feedback from Scene Recall/Store

When a scene is recalled or stored via a push button, the **Status** group object (1-bit) can be used for feedback:

- Value set to 1 (ON) when scene is recalled.
- Value set to 0 (OFF) when scene is stored.



This feedback can be used to trigger further actions, such as visual indication.

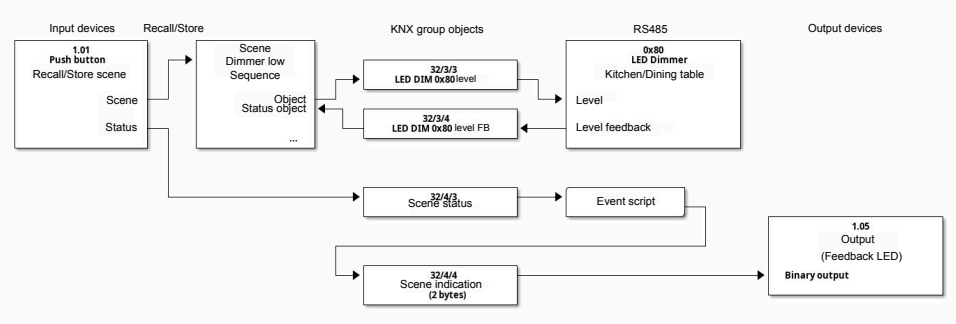
Using Feedback for Visual Indication

A common use case is blinking an LED connected to a digital output on the IHC Output 24 module. This can be achieved using an event script triggered by the **Status** object.

Example event script:

```
-- Scene activated
value = toboolean(event.getvalue()) and 300 or 2000
grp.write('Scene indication', value)
```

- `Scene indication` refers to a 2-byte group object. The output device should be configured as a **Binary output** to use this object.
- The script sets the output ON for:
  - 300 milliseconds when the scene is recalled.
  - 2000 milliseconds when the scene is stored.



Short/Long Press Functionality of a Push Button

Short/long press functionality has been implemented for push buttons. This functionality extends standard button operation and allows a single push button to control multiple actions depending on the press duration.

Short and long press functionality can be used in various use cases, for example:

- Controlling two different lights or switches using short press and long press of the same button.
- Controlling a single light with a short press and recalling a scene with a long press.
- Creating “Follow” functionality using a long press start and long press stop.

This section describes how to configure push buttons with short/long press functionality and provides several practical configuration examples.

Controlling Two Lights with a Single Button

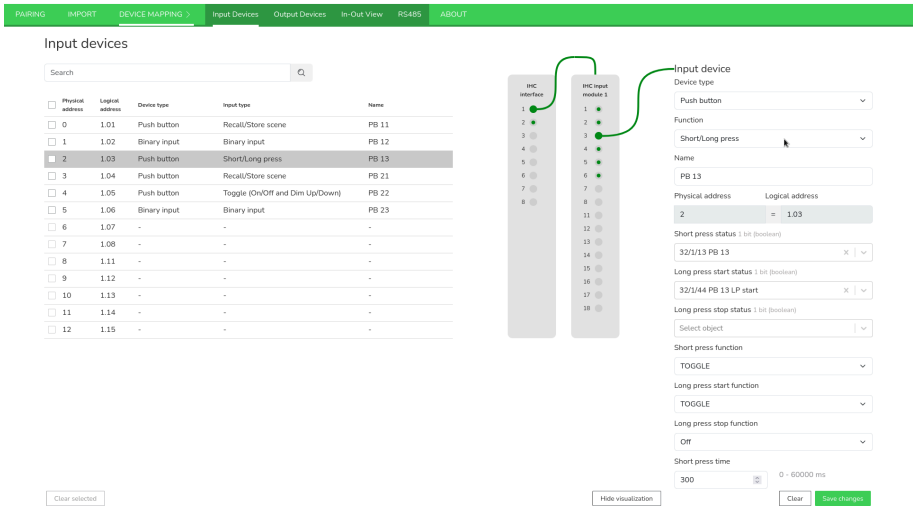
This use case demonstrates how to control two different lights using a single push button by applying **short and long press functionality**.

Configuration of the Push Button

1. Open **IHC Plugin** > **DEVICE MAPPING** > **Input Devices**.
2. Select the input device you want to configure.
3. Set:
   - **Device type**: Push button
   - **Function**: Short / Long press

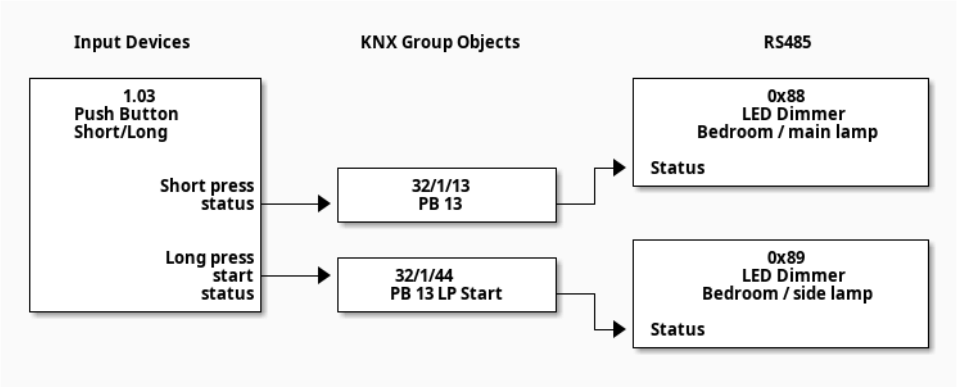
4. Assign KNX group objects for:
   - **Short press status**
   - **Long press start status**

     Existing group objects can be selected, or new ones can be created using the **Add new object** option.
5. Set the following functions:
   - **Short press function**: TOGGLE
   - **Long press start function**: TOGGLE
6. Save the configuration.



Functional Overview

The functional overview is illustrated in the diagram below.



- A **short press** of the push button toggles the first light.
- A **long press** of the same push button toggles the second light.

Each press type generates a separate status signal, which is mapped to individual KNX group objects and controls a different output device. This allows two independent lights or switches to be controlled using a single physical button.

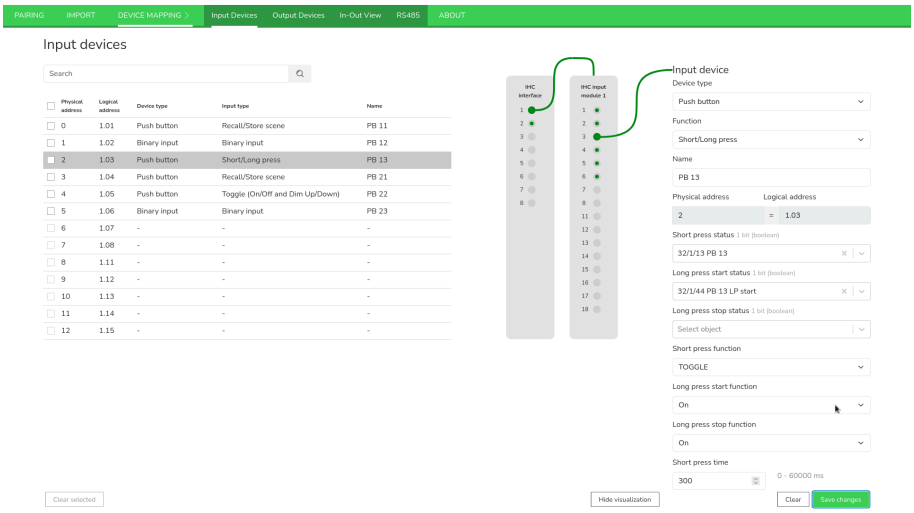
Controlling One Light and Recalling a Scene with a Single Button

This use case demonstrates how to control a single light with a **short press** of a push button and recall a **scene** (for example a central OFF scene) with a **long press** of the same button.

Configuration of the Push Button

1. Open **IHC Plugin** > **DEVICE MAPPING** > **Input Devices**.
2. Select the input device you want to configure.
3. Set:
   - **Device type**: Push button
   - **Function**: Short / Long press
4. Assign KNX group objects for:
   - **Short press status**
   - **Long press start status**

     Existing group objects can be selected, or new ones can be created using the **Add new object** option.
5. Configure the press functions:
   - **Short press function**: TOGGLE
   - **Long press start function**: ON
6. Save the configuration.



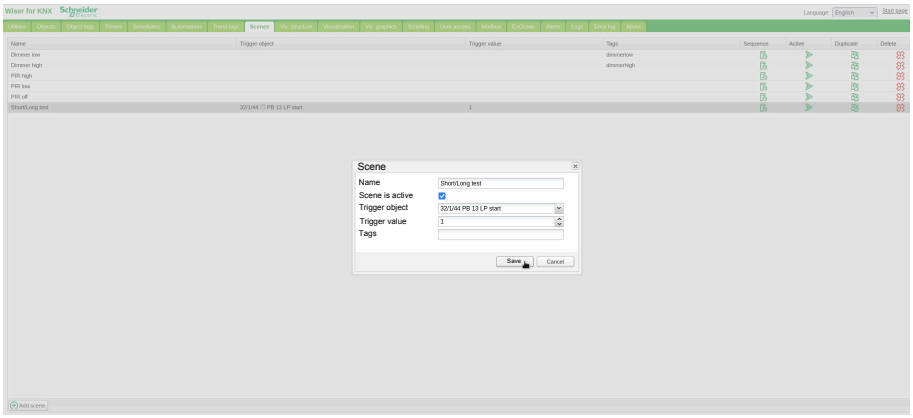
The short press of the button is used to toggle the assigned light on or off. The long press start signal is used to trigger a scene.

Scene Configuration

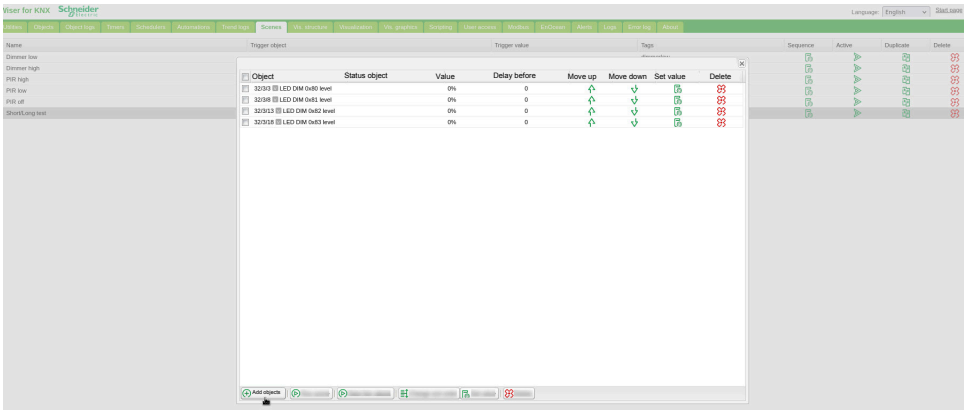
The next step is to create a scene that will be triggered by the long press of the button.

1. In your controller, open the **Configurator**.
2. Navigate to the **Scenes** tab.
3. Click **Add scene** in the bottom-left corner.
4. Fill in the **Name** field (for example Short/Long test). The scene name must be unique.
5. Ensure that the **Scene is active** option is enabled.
6. Select the **Trigger object**, which must be the KNX group object associated with the **Long press start status** of the push button.
7. Set the **Trigger value** to 1 (ON). The scene will be triggered when the long press starts, that is when the **Long press start status** is set to ON.

8. Click **Save**.



After creating the scene, click and add all KNX group objects that should be part of the scene sequence. Set their corresponding values for the scene recall (for example switching lights OFF for a central OFF scene).

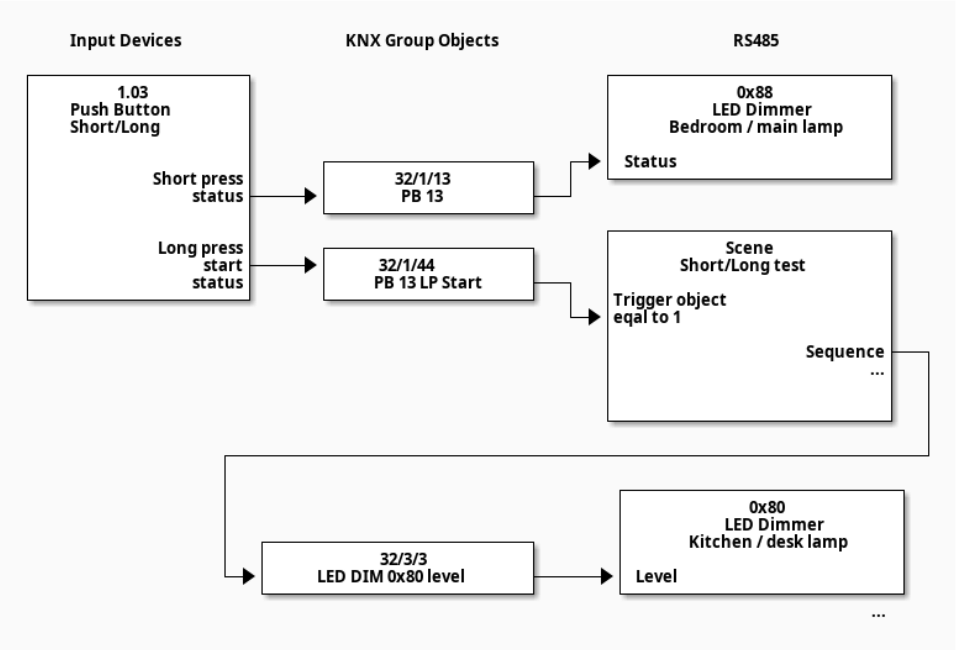


Functional Overview

The functional overview is illustrated in the diagram below.

- A **short press** of the push button toggles the associated light.
- A **long press** of the same button recalls the configured scene.
- The scene is triggered when the long press starts and the **Long press start status** changes to ON.

This configuration allows combining direct light control and scene recall using a single physical button.



Creating “Follow” Functionality with Short/Long Press

This use case describes how to create a simple **“Follow” functionality** using short/long press configuration of a push button. In this setup, **pressing and holding** the button generates an **ON** command on a group object, and **releasing** the button generates an **OFF** command on the same group object.

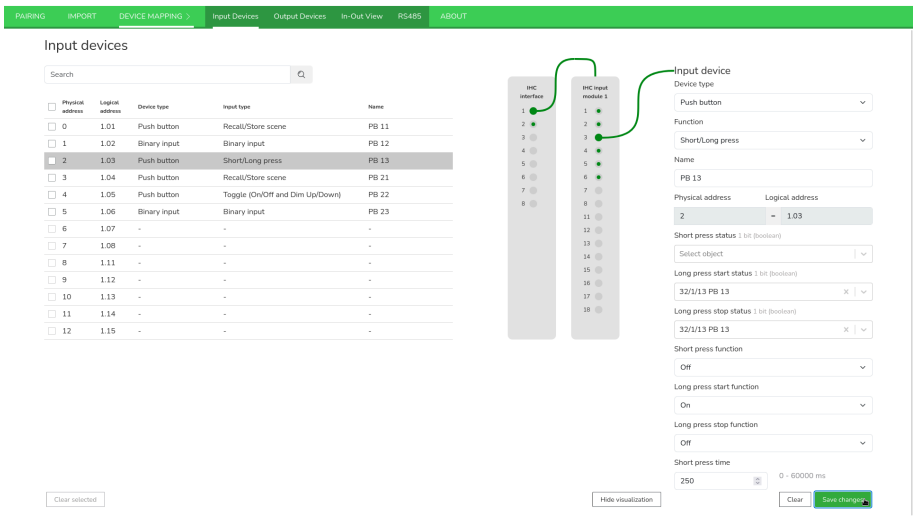
This behavior is typically used for functions such as temporary activation, momentary control, or “press-and-hold” operation.

Configuration of the Push Button

1. Open **IHC Plugin** > **DEVICE MAPPING** > **Input Devices**.
2. Select the input device you want to configure.
3. Set:
   - **Device type**: Push button
   - **Function**: Short / Long press
4. Assign the **same KNX group object** to both:
   - **Long press start status**
   - **Long press stop status**

     Existing group objects can be selected, or a new one can be created using the **Add new object** option.
5. Configure the press functions:
   - **Long press start function**: ON
   - **Long press stop function**: OFF

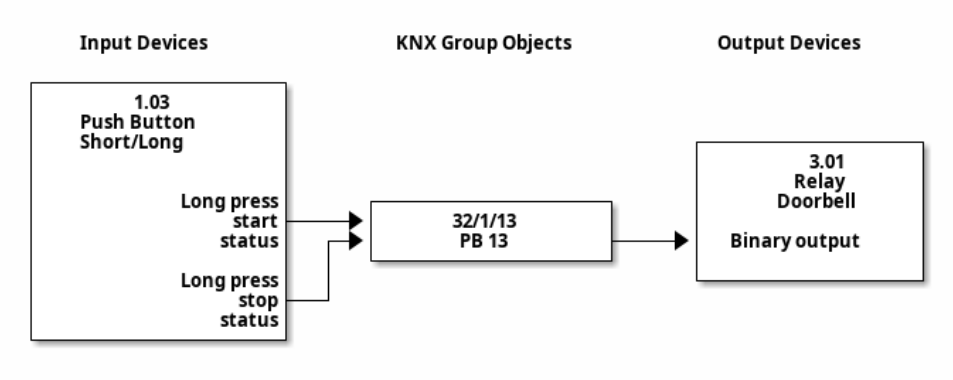
6. Save the configuration.



Functional Overview

The functional overview is illustrated in the diagram below.

- When the push button is **pressed and held**, the **long press start** event is generated and an **ON** value is sent to the assigned KNX group object.
- When the button is **released**, the **long press stop** event is generated and an **OFF** value is sent to the same group object.
- The KNX group object is linked to an output device, such as a binary output, which follows the button state.



This configuration provides a simple and reliable “Follow” behavior, where the controlled output remains active only while the button is pressed.

NOTE: The same functionality on an input device can also be achieved by using the **Binary input** device type with the **Doorbell** function. This alternative solution is simpler to configure and may be preferred for basic follow-type applications.

Staircase Timer Setup for LED Dimmers

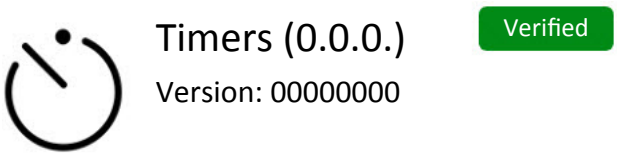
The staircase timer function allows lights to automatically turn off after a specified time interval. This is useful for staircases, hallways, or other areas where lights should not remain on indefinitely.

Installing the Timers Plugin

The staircase timer feature is provided by a separate plugin called **Timers**.

To install:

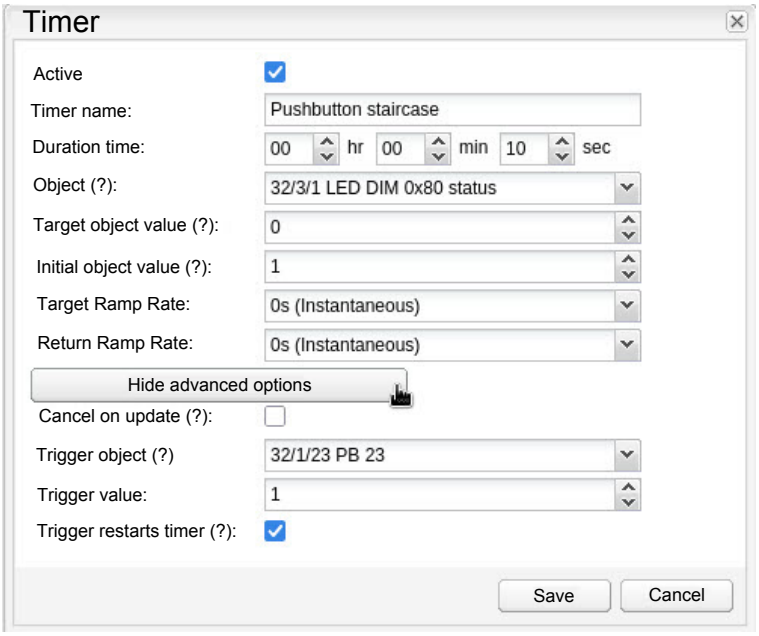
1. Open the Wiser4KNX web interface and go to the Markeptlace .
2. Locate the **Timers** plugin in the app store and install it.



3. After installation, the **Timers** tab will appear in the **Configurator** plugin.

Configuring Staircase Timer

1. Open the **Timers** tab in the **Configurator** plugin of your controller.
2. Click **Add timer** in the bottom-left corner.
3. Configure the following:
   - **Timer name**: Enter a descriptive name.
   - **Duration time**: Set the time after which the light should turn off.
   - **Target group object**: Select the KNX group object associated with the LED dimmer channel. For example, to control LED Dimmer channel you should select here the corresponding "status" group object).
   - **Initial object value**: Set to 1 (ON).
   - **Target object value**: Set to 0 (OFF).
4. Click **Show advanced options** and configure:
   - **Cancel on update**: Determines how the timer reacts when a new value is received while the timer is already running. Leave unchecked.
   - **Trigger object**: Select the KNX group object linked to the binary input you want to use for triggering the timer.
   - **Trigger value**: Set to 1 (ON) to start the timer when the button is pressed.
   - **Trigger restarts timer**: Enable this option to restart the timer if the button is pressed again before it expires.

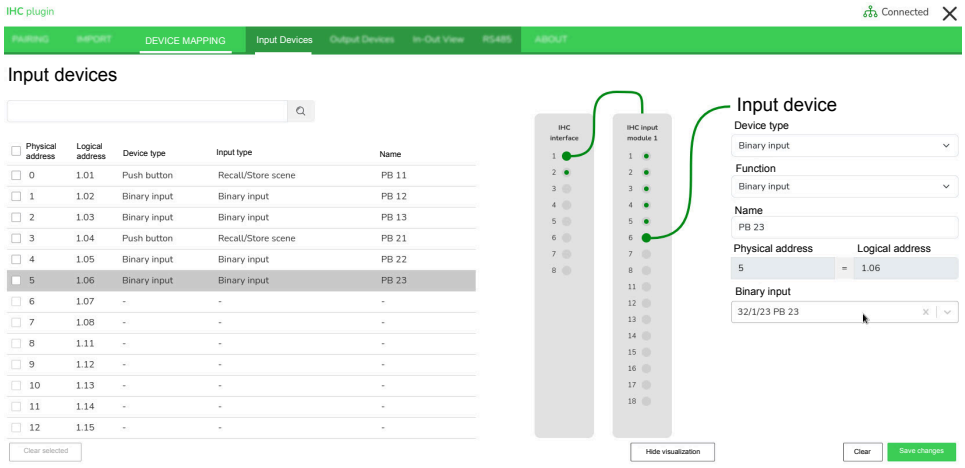


5. Click **Save** to apply the changes.

Linking Binary Input to LED Dimmer via Staircase Timer

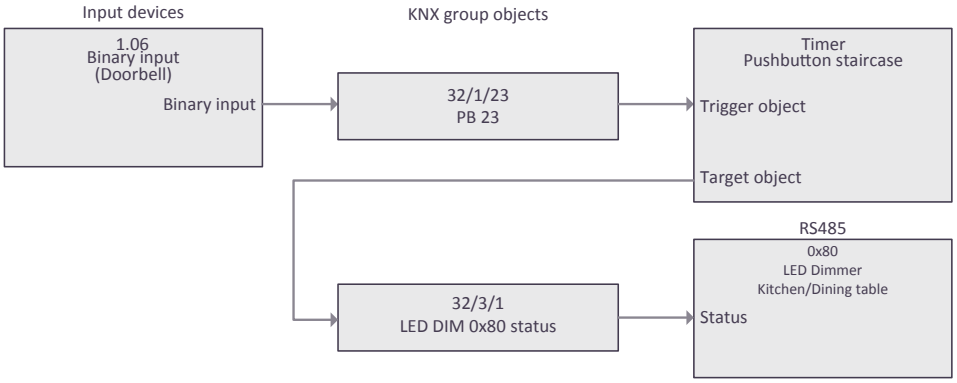
To trigger the staircase timer from a physical button:

1. Configure the binary input in the **Input Devices** tab of the **IHC Plugin**:
   - **Device type**: Binary input.
   - **Function**: Binary input or Doorbell.
2. Assign the binary input's KNX group object as the **trigger object** in the timer configuration.
3. Ensure the LED dimmer channel is linked to the correct **status** group object so the timer can control it.



Functional Overview

- Pressing the button sends an ON command to the LED dimmer and starts the timer.
- When the timer expires, it sends an OFF command to the same dimmer channel.
- If the button is pressed again before the timer expires, the timer restarts.



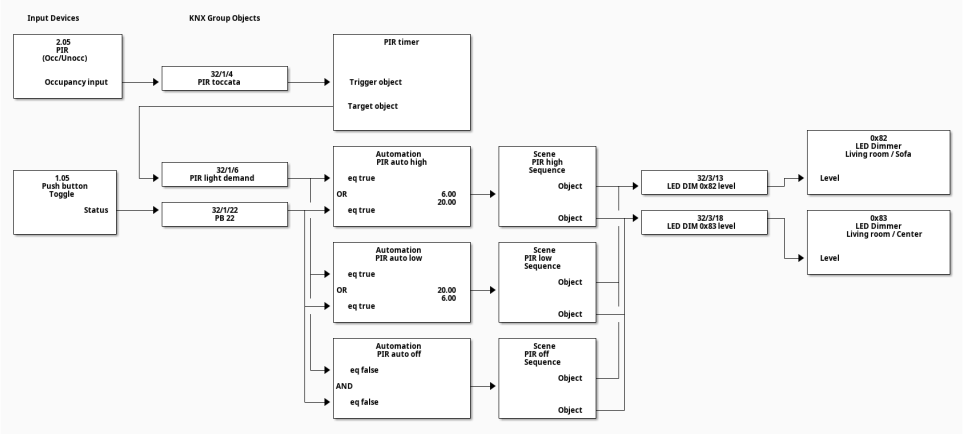
PIR Sensor Based Light Control

Very basic PIR sensor functionality can be achieved in a similar way as **staircase timer functionality, page 51**, simply by connecting a PIR sensor instead of a push button via a binary input. However, in real installations the end user usually expects some additional features, such as:

- Automatic disabling of the PIR sensor when a manual control of the light is detected (for example via a toggle button).
- Different light levels for different times of the day (for example reduced light level during night).
- Automatic disabling of PIR-based control during daytime, etc.

This use case introduces a more advanced solution by combining a **PIR timer** with a **toggle button** and **time-dependent automations**. The resulting configuration allows automatic lighting based on occupancy, while still providing intuitive manual control and flexible light behavior depending on the time of day.

The diagram below illustrates the logical structure of the PIR-based light control solution and the interaction between its individual components.



PIR Sensor Input Configuration

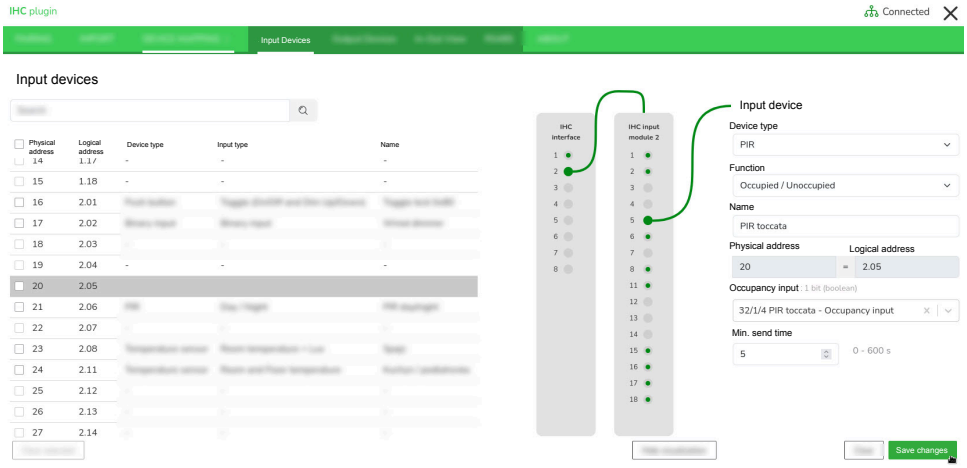
The corresponding input must be configured to work as a PIR occupancy detector.

Configure the input device as follows:

- **Device type**: PIR
- **Function**: Occupied / Unoccupied
- A new KNX group object can be created for the **Occupancy input** field, or an existing one can be selected.
- Recommended **Min. sending time** is 5 seconds. The optimal value depends on the behavior of the connected PIR sensor.

After configuration, the PIR sensor sends occupancy information to the assigned KNX group object, which is then used by the PIR timer and subsequent automations to control lighting behavior.

The figure below shows an example of a PIR input configured in the **Input Devices** tab of the **IHC Plugin**, including device type selection, function assignment, and occupancy group object mapping.



PIR Timer Configuration

The PIR timer is configured in a similar way as the **staircase timer, page 51** described earlier. It is used to switch the light off automatically after a defined time when no occupancy is detected.

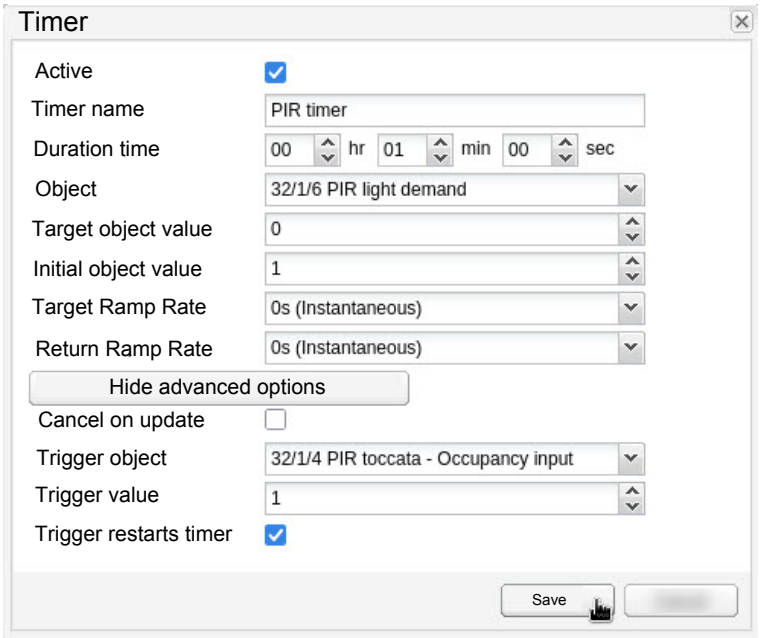
Before creating the timer, an additional **virtual KNX group object** must be created manually:

- Example: 32/1/6 PIR light demand
- Data type: 1-bit (boolean)

This object represents the output of the PIR timer and is later used by automations.

To configure the PIR timer:

1. In your controller, open **Configurator** > **Timers**.
2. Click **Add timer** to create a new timer.
3. Configure the timer parameters.
   The figure below shows an example of a PIR timer configuration, including the trigger object linked to the PIR occupancy input and the target object controlling the light.
4. Save the configuration.



Toggle Button Configuration

The toggle button used for **manual control of the light** is configured in a similar way as described in the **Toggle Button Functionality, page 38** section earlier in this user guide.

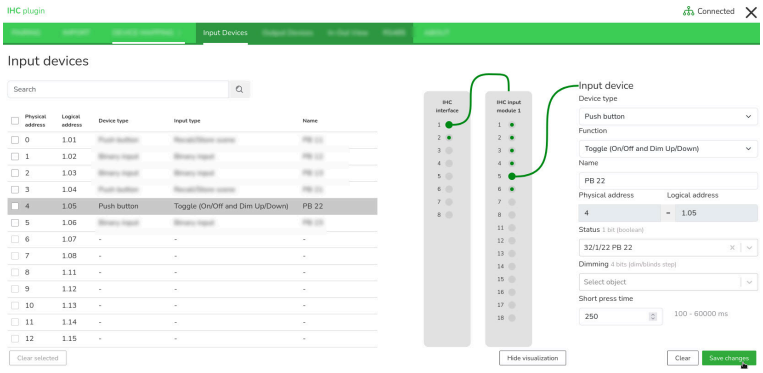
The main difference in this use case is that only the **Status** group object is required for basic ON/OFF control.

To configure the toggle button:

1. Open **IHC Plugin** > **DEVICE MAPPING** > **Input Devices**.
2. Select the input connected to the push button.
3. Configure the input as follows:
   - Device type: Push button
   - **Function**: Toggle (On/Off)
4. Assign a **Status** KNX group object for ON/OFF control.
5. Adjust the **Short press time** if required.
6. Save the configuration.

The toggle button allows the user to switch the light ON or OFF manually, independently of the PIR sensor state. When activated, the toggle button overrides PIR-based control until it is pressed again.

The figure below shows an example of a toggle button configuration in the **Input Devices** tab, including input selection, function assignment, and status group object mapping.



Scenes for the Light Level Control

To control different lighting levels depending on the situation and time of day, separate scenes are used.

Create three scenes with predefined light levels, for example:

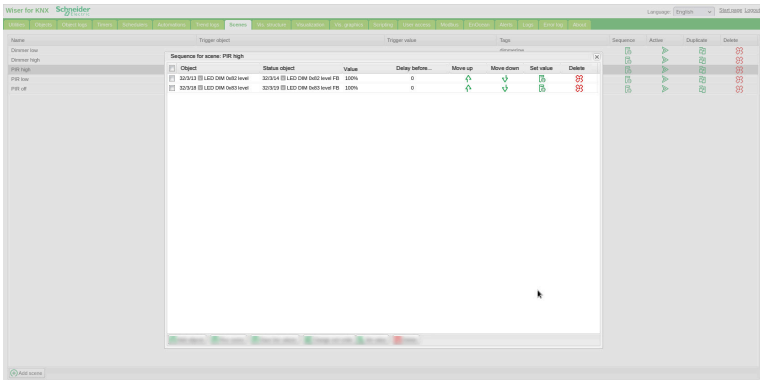
- **PIR high** – light level set to 100 %
- **PIR low** – light level set to 25 %
- **PIR off** – light level set to 0 %

To create the scenes:

1. In your controller, open **Configurator** > **Scenes**.
2. Click **Add scene** to create a new scene and assign a descriptive name (e.g. PIR high).
3. Add the relevant **level** group object of the LED dimmer to the scene.
4. Set the required brightness value.
5. Save the scene.
6. Repeat the same steps for the remaining scenes (PIR low and PIR off).

These scenes are later recalled by automations to apply the correct light level based on PIR activity, manual control, and time conditions.

The figure below shows an example of scene configuration in the **Scenes** tab, including predefined brightness values for the selected LED dimmer channel.



Automation Configuration

As a final step, automations are used to combine the PIR timer output and the toggle button state with time-based conditions.

NOTE: The **Automation** plugin must be installed. A suitable version is typically preinstalled in the Wiser4KNX controller.

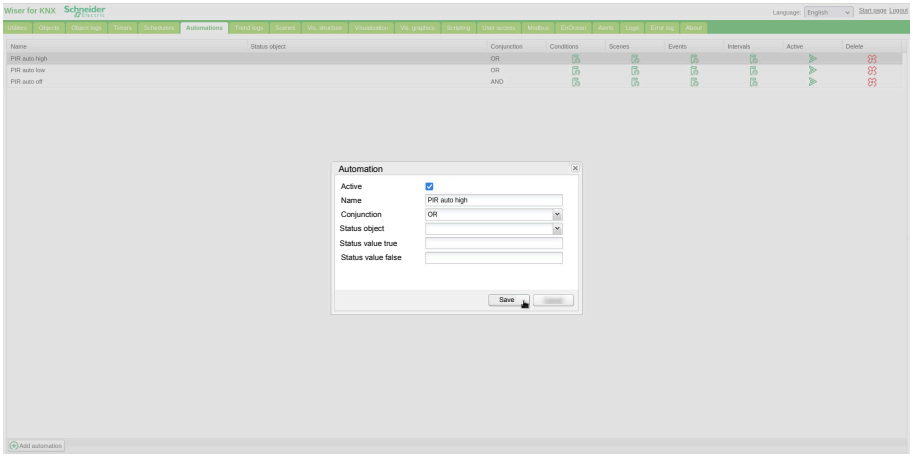
Three separate automations are created:

- one for switching the light ON to a high level,
- one for switching the light ON to a low level,
- one for switching the light OFF.

Automation for High Light Level

This automation controls the light during daytime.

1. In your controller, open **Configurator** > **Automations**.
2. Click **Add automation** to create a new automation.
3. Enter the name and set **OR** as a **Conjunction type**.
4. Click **Save** to confirm.



5. Click icon to open the **Conditions** dialog.
6. Click **Add condition** in the bottom-left corner of the **Conditions** dialog.
7. Fill in the condition parameters.

   First condition:

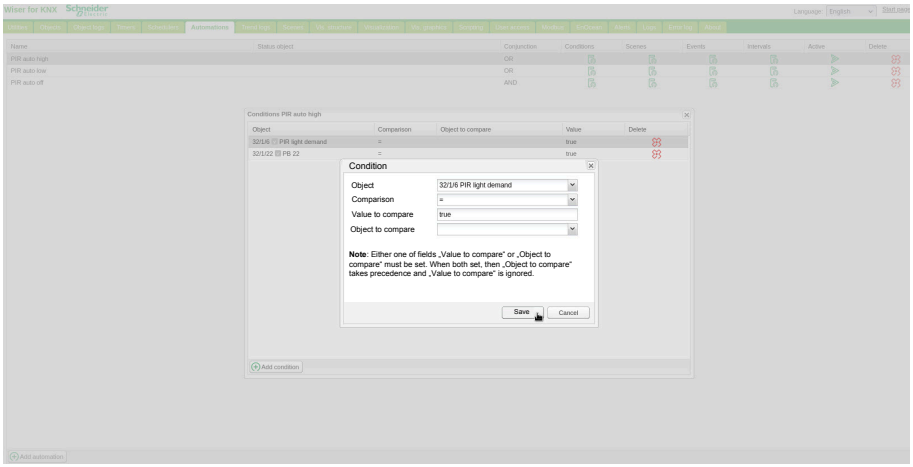
   - **Object**: *32/1/6 PIR light demand*
   - **Comparison**: =
   - **Value to compare**: `true`

     NOTE: It is important to use boolean value `true` here, not the numbers (1/0).

   Second condition:

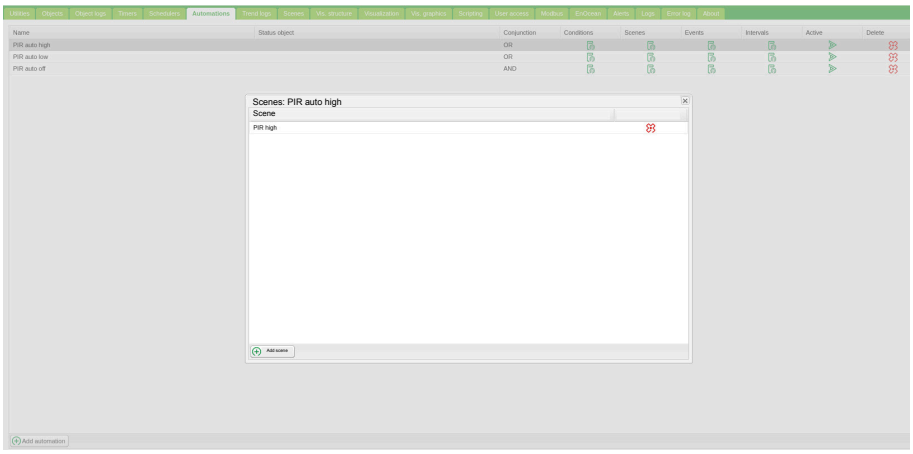
   - **Object**: *32/1/22 PB 22*
   - **Comparison**: =
   - **Value to compare**: `true`

     NOTE: It is important to use boolean value `true` here, not the numbers (1/0).
8. After adding both conditions, save your settings.



In the next step, you select the scene(s) to be recalled when the automation is triggered:

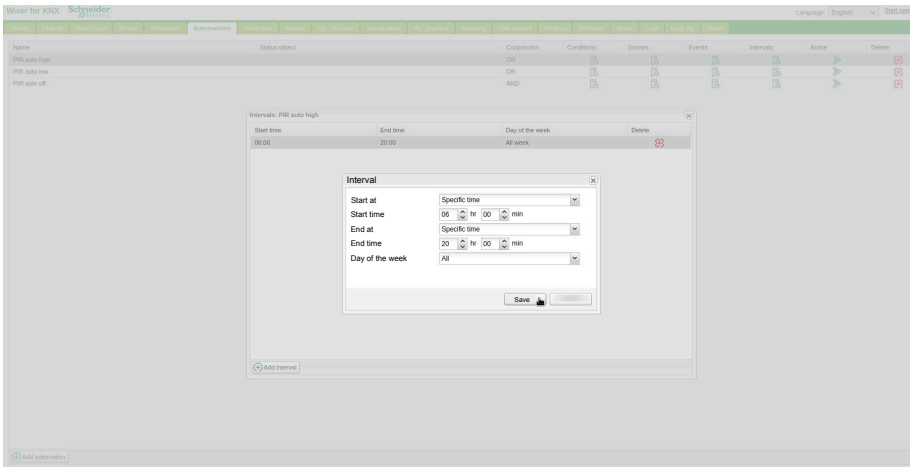
1. Click to open the **Scenes** dialog.
2. Click **Add scene** in the bottom left corner of the **Scenes** dialog.
3. Select the *PIR high* scene, click **Save**, and close the **Scenes** dialog.



In the next step, you define time interval when the automation will be active:

1. Click  to open the **Intervals** dialog.
2. Click **Add interval** in the bottom left corner of the **Intervals** dialog.
3. Set the start/end time of your interval and set the **Day of the week** parameter.

   NOTE: It is possible to set interval crossing midnight, e.g., from 20:00 till 6:00 next day. It is possible to select sunrise or sunset with given offset as interval start/end. In this case, the controller must have correct geographical location configured to calculate proper time of sunrise/ sunset.
4. You can add multiple intervals, e.g., to cover different time ranges during the day.
5. After adding all necessary intervals, click **Save**, and close the **Intervals** dialog.



Finally, activate your **Automation** by clicking the activation icon  .

Automation for Low Light Level

This automation controls the light during nighttime. The conditions are the same as those used for the high light level automation, page 59. The recalled scene is *PIR low*. The active time interval can be set, for example, from 20:00 to 06:00 the next day.

Automation for Switching Off the Light

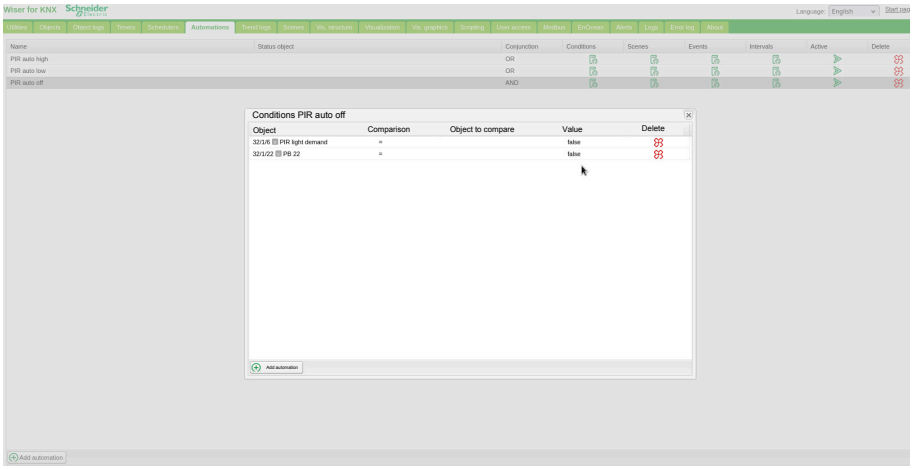
This automation switches the light off when neither PIR demand nor manual control is active.

- Create another automation, for example named *PIR auto off*, similarly as described above.
- Set the **Conjunction type** to **AND**.
- Configure the condition parameters.

  First condition:
  - **Object**: *32/1/6 PIR light demand*
  - **Comparison**: =
  - **Value to compare**: `false` (boolean value)

  Second condition:
  - **Object**: *32/1/22 PB 22*
  - **Comparison**: =
  - **Value to compare**: `false` (boolean value)

- The scene to be recalled is *PIR off*.
- No time intervals are defined, therefore the automation is active all the time.

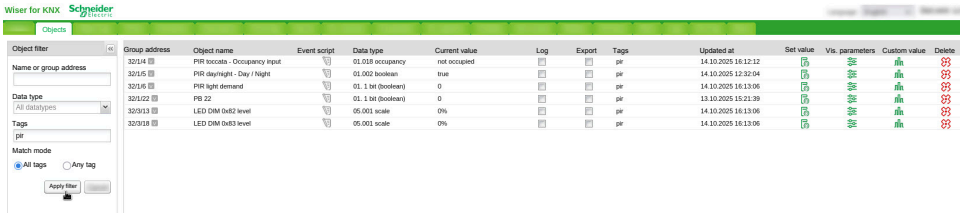


Testing the PIR Sensor Functionality

After creating all three automations, the PIR sensor functionality should be tested.

For easier testing and diagnostics, it is recommended to mark all participating KNX group objects with a common tag, for example `pir`. This allows filtering all relevant objects in the **Objects** tab of the **Configurator**.

In your controller, open **Configurator** > **Objects** and filter the object list using the selected tag. This view displays all input and output objects involved in the PIR control logic.



In this view, you can manually set values of **input objects**, for example to simulate PIR sensor activation or toggle button operation. Changes to **output objects** are displayed in real time.

Functional Overview

The behavior of this example should be as follows:

- When the PIR sensor is triggered (input object *31/1/4 PIR toccata – Occupancy input* is set to `true`), the light is switched ON:
  - to high level (100 %) if the current time is between **06:00 and 20:00**,
  - otherwise to low level (25 %).
- If no occupancy is detected and the PIR timer expires, the light is switched OFF.
- If the toggle button is pressed (input object *32/1/22 PB 22* is set to `true`), the light is switched ON independently of the PIR sensor state. The scene *PIR high* or *PIR low* is selected according to the current time.
- The light remains ON until the toggle button is pressed again (input object is set to `false`).

Possible Improvement

The group object *32/1/22 PB 22* can also be used in an **Output device / Binary output** connected to an LED in the push button. This provides visual feedback about the current state of the toggle button.

NTP Client/Server

The **Network Time Protocol (NTP)** is a networking protocol used for clock synchronization in computer systems over packet-switched, variable-latency data networks.

NTP Functionality Overview

- The **IHC Interface** runs an **NTP client**.
- This client synchronizes the internal clock of the **IHC Interface** with an NTP server.
- The internal clock is used for:
  - audit log time stamps,
  - validity checks of certificates during secure communication.

The **Wiser for KNX** controller can be configured as an **NTP server**. In this case, the **IHC Interface** synchronizes its time directly from the controller.

Configuring NTP Server on the Wiser for KNX Controller

The NTP server functionality is provided by the controller firmware and is configured using the controller web interface.

To configure the controller as an NTP server:

1. Open the controller web interface.
2. Navigate to **Configurator** > **System** > **Services**.
3. Select **NTP client/server** from the drop-down menu.
4. Set the **Local server static** option to **Enabled**. This makes the controller act as an NTP server.
5. Set the **Client status** option to **Disabled** if your controller has no access to the Internet NTP servers.
   If you enable **Client status**, up to four NTP server addresses can be configured.
   The servers must be reachable from the controller network.
6. Click **OK** to apply the changes.
7. Restart the controller if required.

After configuration, the controller provides time synchronization services to connected devices, including the **IHC Interface**.

Checking Current Wiser for KNX Controller Time

To check the current time on the controller:

1. Open the controller web interface.
2. Navigate to **Configurator** > **Date and time**.

The displayed time confirms that time synchronization is working correctly and serves as the reference for all time-dependent functions, such as automations, timers, scenes, and certificate validation.

Printed in:
Schneider Electric Danmark A/S
Lautrupvang 1
2750 Ballerup - Danmark
+ 45 88 30 20 00
www.lk.dk

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

260422_LSS100500_IHC_Interface_SW