

Relay Actuator Demo

Manual

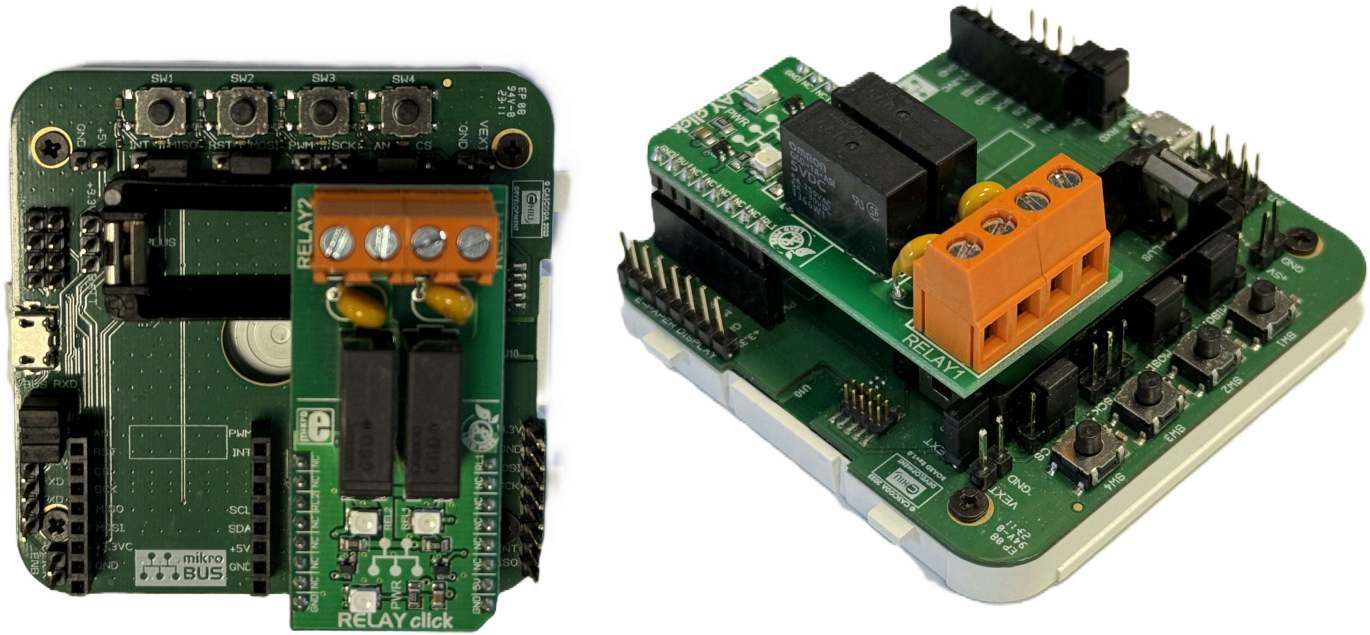


Table of contents

- [Relay Actuator Demo](#)
- [Table of contents](#)
- [Relay Actuator Demo Features](#)
 - [About Cascoda's KNX IoT Development Board](#)
 - [Using the Devboard as a Relay Actuator Demo](#)
 - [Demo Overview and Requirements](#)
- [General information](#)
 - [Document Version information](#)
 - [Used Terms](#)
 - [Safety instructions](#)
 - [Issues](#)
 - [Contact information](#)
- [Setup and Configuration](#)
 - [Device Startup](#)
 - [Hardware Setup](#)
 - [Flashing the Firmware](#)
 - [Commissioning](#)
 - [Thread Commissioning](#)
 - [KNX Commissioning](#)
 - [Adding the Device to a Topology](#)
 - [Creating a Configuration](#)
 - [Downloading the ETS configuration](#)
 - [Resetting the device](#)
- [Running the Demo with ETS](#)
- [Software Bill of Materials](#)
 - [Cascoda SDK](#)
 - [tinycbor](#)
 - [mbedtls](#)
 - [Openthread](#)
 - [Data points](#)
 - [Parameters](#)

Relay Actuator Demo Features

The relay actuator demo runs on Cascoda's KNX IoT Development Board, equipped with the RELAY click from MikroElektronika.

About Cascoda's KNX IoT Development Board

Cascoda's KNX IoT Development Board provides everything you need to develop your KNX IoT over Thread application with the Cascoda Chili module. It features Cascoda's SMARTRange™ technology to provide long-range Thread connectivity for whole-house coverage. In addition, Cascoda provides easy integration with Cascoda's dashboard and our KNX IoT Hub for easy configuration and backhaul connectivity.

The KNX-IoT Development Kit contains two Development Boards, and is available [from our distributors](#). You can view the [Getting Started Guide](#), and if you want detailed information on the Development Board you may [download the datasheet](#).

The KNX-IoT Development Board works seamlessly with ChiliCuisine™, Cascoda's low-code development environment.

Features:

- Certified Thread-based radio supporting meshed networking, with long range
- Easy integration with Cascoda's Border Router for IPv4/IPv6 backhaul connectivity
- Ultra-low-power battery operation & battery charging
- Four programmable buttons & LEDs
- E-Paper display integration
- A library of sensor & actuator plug-in boards, to develop your solution
- Automatic code generation for your chosen target combination of sensors/actuators/switches/leds

Specification:

- USB or UART connectivity
- 5V or 3.3V supply options
- 3.7V Li/LiPo battery integration with charging from a 5V supply or USB
- Two Mikroelektronika Click™ slots for a huge range of plug-in sensor/actuator connectivity
- Integrated chip antenna and an external antenna option
- Dimensions 66 x 64mm

Using the Devboard as a Relay Actuator Demo

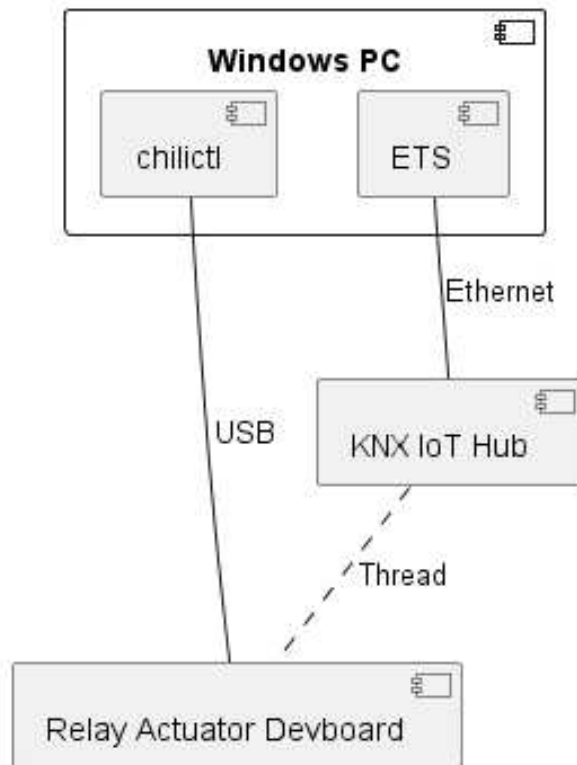
Once the device is fully installed and commissioned, it operates as relay actuator. It can react to the following commands, sent by another KNX entity (e.g. a switching sensor, or the ETS group monitor):

- On/Off Relay 1
- InfoOnOff Relay 1 (feedback)
- On/Off Relay 2
- InfoOnOff Relay 2 (feedback)

Therefore, using appropriate counterpart controllers, you can control the state of the two relays on the device.

NOTE! Do not power the board with a battery when using this demo. Instead, power via USB.

Demo Overview and Requirements



The Demo consists of controlling the Relay Actuator using the Group Monitor on ETS. In order to do this, you will need the following:

- Hardware requirements
 - A Windows PC
 - A Thread Border router. Cascoda recommends using the [KNX IoT Hub](#).
 - A development board with jumpers provided
 - Relay click board
 - USB-A to Micro-USB cable
- Software requirements
 - ETS version 6.2.2 or later
 - Cascoda Windows Tools
 - An application binary
 - A serial number binary
 - A project file to load into ETS

This manual will show you how to set up the hardware, flash the device, commission it onto the Thread network and onto KNX, and then finally being able to control it using ETS.

General information

Document Version information

This manual is amended periodically and will be brought into line with new software releases. The change status (date) can be found in the contents header. If you have a device with a later software version, please check www.cascoda.com to find out whether a more up-to date version of the manual is available.

Used Terms

Sign	Description
DANGER!	Indicates an immediately hazardous situation which will lead to death or severe injuries if it is not avoided.
CAUTION!	Indicates a potentially hazardous situation which may lead to trivial or minor injuries if it is not avoided.
WARNING!	Indicates a situation which may lead to damage to property if it is not avoided.
NOTE!	Indicates a situation which may lead to possible (known) side effects.

Safety instructions

DANGER! RELAY Click has exposed pins/pads. To stay safe, take precautions when applying high voltage to the RELAY Click board™. The RELAY Click board™ needs to be used by trained personnel only while applying high voltages.

Issues

Questions about the product?

You can reach the technical service of Cascoda under Tel. +44 (0)2380 638 111 or support@cascoda.com.

We need the following information to process your service request:

- Type of appliance (model name or item number)
- Description of the problem
- Serial number or software version
- Source of supply (dealer/installer who bought the device from Cascoda)

For questions about KNX functions:

- Version of the device application
- ETS version used for the project

Contact information

info@cascoda.com
Threefield House,
Threefield Lane,

Southampton,
SO14 3LP, UK

Setup and Configuration

Device Startup

The device starts up when plugged in via USB. As a reminder, do not use the battery for this demo.

Hardware Setup

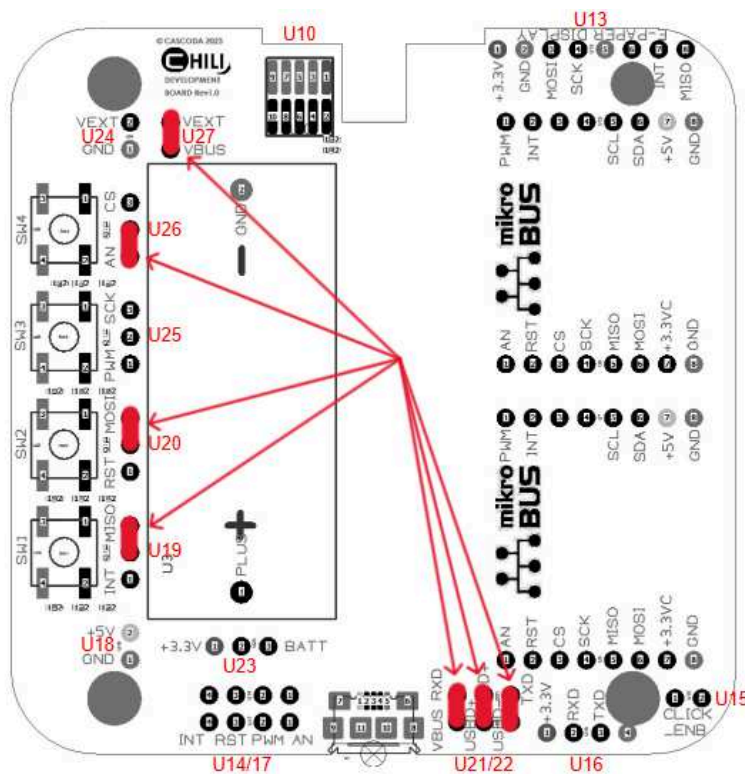
NOTE! Do not power the board with a battery when using this demo. Instead, power via USB.

For this demo, you will need the following hardware:

- Development board with jumpers provided
- RELAY Click
- USB-A to Micro-USB cable

To set up the hardware, follow the following steps:

1. When using the KNX IoT Development board, the jumpers need to be placed in the correct position, and this differs from application to application, as well as depending on which means of power is provided. For this demo, please make sure that your jumper configuration exactly matches what is shown in the picture below (see the arrows pointing to the red blocks, which represent the jumpers). Also, make sure that none of the jumpers are loose.



2. Place the RELAY Click, as shown in the pictures at the front page of this document. The end of the RELAY Click which has the relays should face the battery slot. You can insert the Click board on the right or left slot, both are the same.

CAUTION! Make sure the bottom of the RELAY click board does not make contact with the battery slot below it. Lift up the RELAY click slightly if needed.

3. Connect the USB cable into the USB port on the development board.

Flashing the Firmware

Make sure you have the following before getting started:

- An installation of Cascoda's Windows Tools. Please [download CascodaWindowsTools.zip and run the installer within](#). Two of the tools will be necessary for this guide, namely `chilictl.exe` and `serial-adapter.exe`. NOTE: By default, these tools are added to your PATH, enabling their execution in a shell in any directory. However, if this did not occur, you will only be able to execute the tools from within the directory in which they are installed. The default installation directory is `C:\Program Files(x86)\Cascoda Windows Tools`.
- The application binary `knx_relay_reed.bin` for the Relay actuator demo.
- The serial number binary `029B0000251.bin` for the Relay actuator demo.

Once you have all of the above, follow the steps below:

1. Connect the development board to a Windows PC via USB.
2. Determine the development board's unique serial number (note: this is unrelated to the KNX serial number, which we talk about in the rest of this document), by using the Cascoda Windows Tool `chilictl.exe`, using the command shown below. Note that this will list all connected devices, so if you have multiple devices and want to identify which one is the devboard, then run the command once with the devboard disconnected, then again with it connected, so that you can identify the new one that appears:

```
$ chilictl.exe list
2024-08-08 12:26:45.919 NOTE: Host Cascoda SDK v0.23-27-g0392f99 Jun 17 2024
Device Found:
  Device: Chili2
  App: mac-dongle
  Version: ???
  Serial No: 831980C06D0F3D2A
  Path: \\?\hid#vid_0416&pid_5020#a&ab509a761&0000#{4d1e55b2-f16f-11cf-88cb-001111000030}
  Available: No
  External Flash Chip Available: No
Device Found:
  Device: Chili2
  App: knx_relay_reed
  Version: v0.23-985-g74ba8ac3
  Serial No: 669F33625BDE480C
  Path: \\?\hid#vid_0416&pid_5020#a&13e664b5&1&0000#{4d1e55b2-f16f-11cf-88cb-001111000030}
  Available: Yes
  External Flash Chip Available: Yes
```

3. Flash the application binary `knx_relay_reed.bin` using `chilictl.exe`. Use the command shown in the picture below. The number that comes after `-s` should be the serial number that you have identified in the previous step.

```
$ chilictl.exe flash -s 669F33625BDE480C -cf knx_relay_reed.bin
Last write time of knx_relay_reed.bin is: 08/08/2024 11:40
2024-08-08 12:29:41.406 NOTE: Host Cascoda SDK v0.23-27-g0392f99 Jun 17 2024
1 devices found.
Flasher [669F33625BDE480C]: INIT → REBOOT
Flasher [669F33625BDE480C]: REBOOT → ERASE
Flasher [669F33625BDE480C]: ERASE → FLASH
Flasher [669F33625BDE480C]: FLASH → VERIFY
Flasher [669F33625BDE480C]: VERIFY → VALIDATE
Flasher [669F33625BDE480C]: VALIDATE → COMPLETE
```

4. Double check that the application binary is now flashed onto the device, by running the same command as step 2, and making sure that the "App" name is `"knx_relay_reed.bin"`.
5. Flash the serial number binary `029B0000251.bin` using `chilictl.exe`, with the command shown below.

NOTE! This serial number binary should only be flashed on 1 device per network partition. If two or more devices on the same Thread network have been flashed with this serial number binary, the demo won't work. If you need to have multiple Relay actuator devices on the same network, please contact us so that we can provide you with a different unique serial number.

```
$ chilibl.exe flash -s 669F33625BDE480C -m 029B00000251.bin
2024-08-08 12:31:23.804 NOTE: Host Cascoda SDK v0.23-27-g0392f99 Jun 17 2024
1 devices found.
Flasher [669F33625BDE480C]: INIT → REBOOT
Flasher [669F33625BDE480C]: REBOOT → ERASE
Flasher [669F33625BDE480C]: ERASE → FLASH
Flasher [669F33625BDE480C]: FLASH → VERIFY
Flasher [669F33625BDE480C]: VERIFY → VALIDATE
Flasher [669F33625BDE480C]: VALIDATE → COMPLETE
```

Your device is now ready to start the commissioning process!

Commissioning

After powering the device using the USB cable, the device will enter a commissioning phase. Thread commissioning needs to be done first, since this will enable IPv6 communication. Next, KNX commissioning needs to be done using ETS (version ETS 6.2.2 or later).

For both Thread and KNX commissioning, the QR code below will be needed.



In case you are unable to scan the QR code, you can instead copy/paste the following:

41S029B00000251+3ZEUI:03BBE306C89826FB.P:57KC2E9GG3.PA:G2P45SUYP

Check out the youtube video [here](#), demonstrating the process of doing Thread and KNX Commissioning using a QR code scanner.

Thread Commissioning

Thread commissioning is adding the device to the thread network. To be able to do so, one needs to have a Thread Border router. Cascoda recommends using the [KNX-IOT-HUB](#).

NOTE: The steps below assume that you already have a Thread network set up using the KNX IoT Hub. If you don't, then click [here](#) for instructions on how to do that (follow the part regarding using the Web UI).


1. Hover over "Thread" in the menu bar at the top, and click on "Add Device".
2. There is an input field to put QR code information. Using a QR code scanner, scan the QR code provided in this manual and then click "Submit".

No password set!

There is no password set on this router. Please configure a root password to protect the web interface.

Add Joiner in Network: OpenThread

QR Code Entry Manual Entry



Request Camera Permissions
[Scan an Image File](#)

Back to View Submit

3. Now wait for the device to join the Thread network. When it does, you will see it show up in the "Neighbors" table.

No password set!

There is no password set on this router. Please configure a root password to protect the web interface.

Thread View: OpenThread (wpan0)

This is the list and topograph of your thread network.

List Topology Graph IPv6 Addresses

Leader Situation of Network

Leader Router Id	Partition Id	Weighting	Data Version	Stable Data Version
37	618116596	64	27	20

Neighbors

RLOC16	Role	Age	Avg RSSI	Last RSSI	Mode	Extended MAC
0x9402	Child	6	-31	-32	rdn	066bd20a17a739bd
Pending	New Joiner	Pending	Pending	Pending	Pending	b7092300cef82669

Remove

NOTE: The above steps are enough to move forward with the demo. See [here](#) for more details about Thread commissioning.

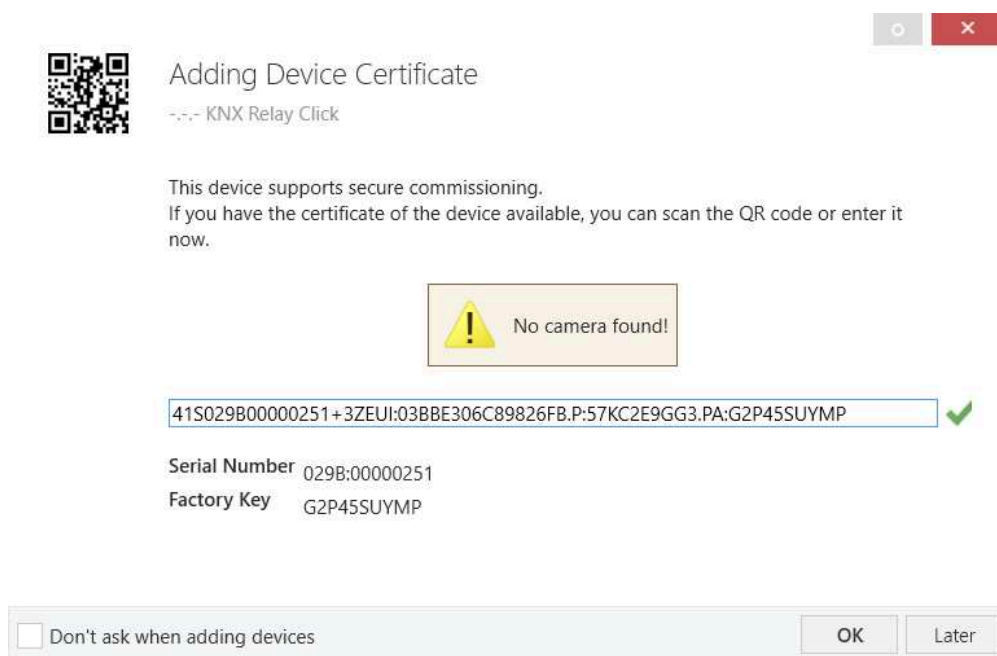
KNX Commissioning

KNX commissioning is adding the device to an ETS project. Since KNX IoT is a secure KNX protocol, one needs to have the security credentials and the serial number of the device. This information is contained in the QR code.

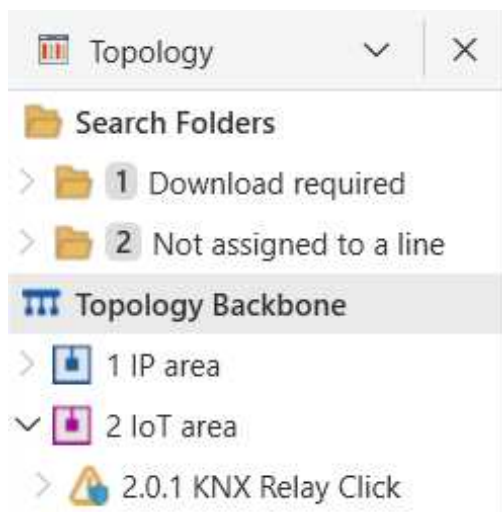
The device can only be added to a KNX IoT Area or Line. When the device is added to a KNX IoT area or Line, the credentials can be supplied. ETS can scan the QR code with the camera (or 2D bar code scanner).

Adding the Device to a Topology

1. Open the project provided in ETS version 6.2.2 or later.
2. Open the "Devices" panel, and select the device that you will use for this demo. In the "Properties" section on the right, click "Add Device Certificate". A window will pop up, with an input field for the QR code information. Use a QR code scanner to scan the QR code provided in this manual. (There is also an option to use a camera).



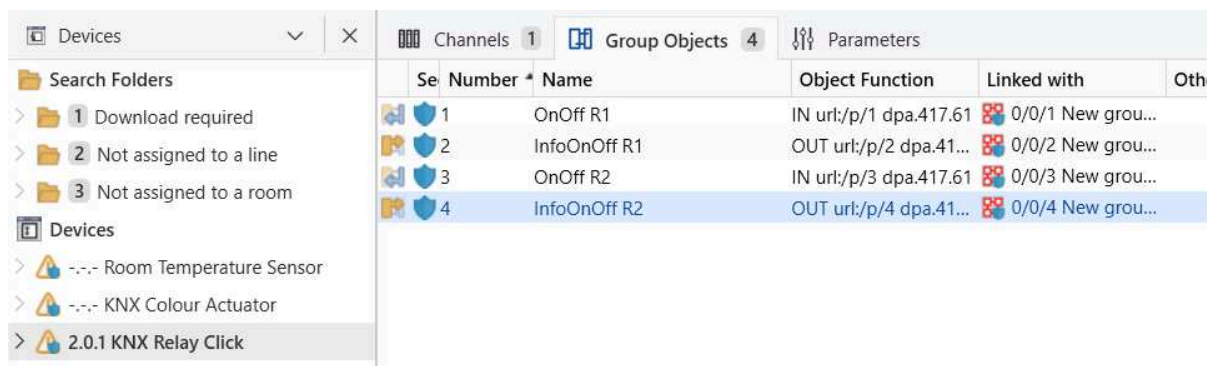
3. Open a "Topology" panel. Drag and drop the device into an IoT area.



Creating a Configuration

These steps create the group objects that will be used in the s-mode messages. By linking the group objects to the communication objects (data points), one can send s-mode messages to actuators, and receive s-mode messages from sensors.

1. Open a "Group Addresses" panel.
2. Select "Group Addresses", right click, select "Add Main Groups", click OK.
3. Select the newly created main group, right click, select "Add Middle Groups", click OK.
4. Select the newly created middle group, right click, select "Add Group Addresses", increase the count to however many group addresses you want (e.g. the same as the number of Group Objects that you would like to link), then press "OK".
5. Now open the "Group Objects" tab for your device, and link each group object that you want by right clicking, "Link with...", and then selecting the group address.



Se	Number	Name	Object Function	Linked with	Oth
1		OnOff R1	IN url:/p/1 dpa.417.61	0/0/1 New grou...	
2		InfoOnOff R1	OUT url:/p/2 dpa.41...	0/0/2 New grou...	
3		OnOff R2	IN url:/p/3 dpa.417.61	0/0/3 New grou...	
4		InfoOnOff R2	OUT url:/p/4 dpa.41...	0/0/4 New grou...	

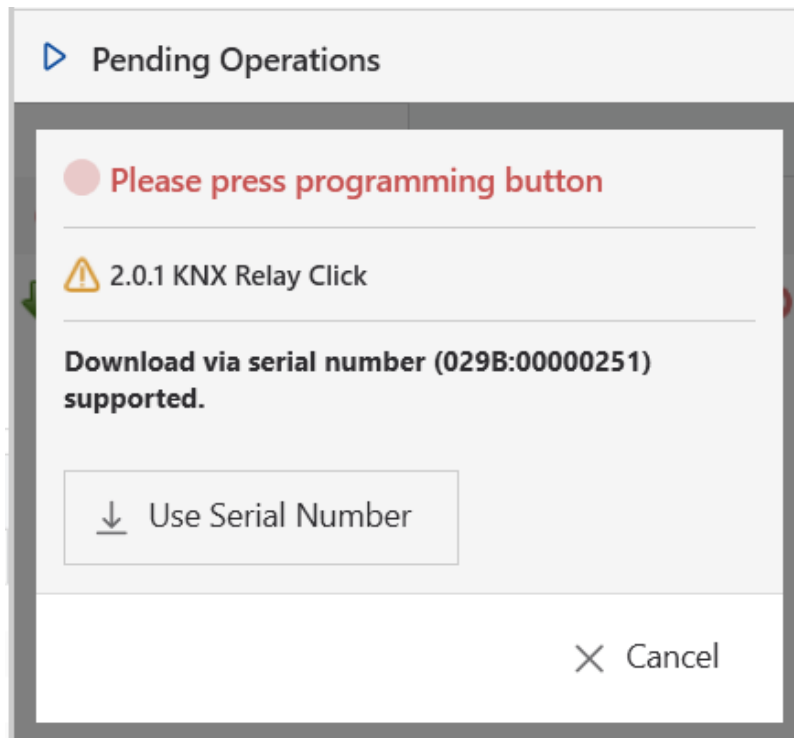
Downloading the ETS configuration

The downloading of the configuration can happen when the ETS data for the device is created, i.e.:

- The parameters are set
- The communication objects are connected

This has already been done with the previous two steps. Therefore, you can now download the configuration by selecting the device in the "Devices" panel, right click, hover over the "Download" option, and click "Download All". After waiting a few seconds, there will be a popup asking you to either:

- download by serial number
- download with programming mode



The download by serial number does not require any interaction with the device. All you have to do is click on the "Use Serial Number" button.

The download with programming button requires pressing the programming button. The **PROG** button is the SW4 button, which needs to be held down for 1 second. While the **PROG** button is pressed, the LED is on. When the **PROG** button is released and the device is in programming mode the LED will start blinking. Disabling the programming mode can be achieved by holding down the **PROG** button again for 1 second.

NOTE: The device has to be attached to a Thread network in order for it to enter programming mode. In case you are unable to put the device into programming mode, double check that it is indeed attached to a Thread network. And also make sure you are not releasing the button too soon (hold it down for > 1 second).

Once the download is complete, the device becomes fully operational and functional.

Resetting the device

NOTE! This section is informational. If you are simply following the steps to run the Demo with ETS, please do not reset the device!

The device allows resetting of KNX and Thread in separate steps. This allows that the KNX configuration can be reset to factory default, without resetting the connectivity part.

- Reset KNX

Reset of KNX is achieved by pressing the **PROG** button for 5 seconds. While the **PROG** button is pressed, the LED is on.

When the **PROG** button is released (after 5 sec), the LED will quickly flash 2 times.

NOTE! KNX Reset: this means that also the security credentials are removed. Hence ETS will download newly created device keys.

- Reset Thread

Reset of Thread is achieved by pressing the **PROG** button for 10 seconds. While the **PROG** button is pressed, the LED is on. When the **PROG** button is released (after 10 sec), the LED will slowly flash 3 times.

NOTE! Thread Reset: This means that the device needs to be added to the thread network again.

Running the Demo with ETS

You can run a demo using only 1 Relay actuator device and ETS. The Group Monitor can be used as a controller, sending messages to the configured datapoints on the device, and thus causing the device to behave as if it were receiving those same messages from another KNX IoT counterpart device.

NOTE: This section assumes that you have followed the Thread Commissioning and KNX Commissioning steps, and thus have already downloaded a configuration to your device using ETS.

With that being said, follow the steps below to control the Relay Actuator:

1. Open the Group Monitor (Panels > Diagnostics > Group Monitor), and click "Start".
2. To switch RELAY1 of the Relay device ON, select the Group Address linked with the "OnOff R1" Group Object, change the "Value" to "On", and click "Write". The Relay 1 of the device will turn on (this will make a clicking sound, and you will see the associated LED turn on). Send "Off" to turn it off. You can also observe the "InfoOnOff R1" feedback being sent on the appropriate group address.

The screenshot shows the ETS 2.0.1 KNX Relay Click interface. The top panel displays a table of Group Objects:

Se	Number	Name	Object Function	Linked with	Other Linked	Length	C	R	W	T	U	Data Type	Priority
1		OnOff R1	IN url:/p/1 dpa.417.61	0/0/1 New grou...		1 bit	C	-	W	-	U	switch	Low
2		InfoOnOff R1	OUT url:/p/2 dpa.417.51	0/0/2 New grou...		1 bit	C	R	-	T	-	switch	Low
3		OnOff R2	IN url:/p/3 dpa.417.61	0/0/3 New grou...		1 bit	C	-	W	-	U	switch	Low
4		InfoOnOff R2	OUT url:/p/4 dpa.417.51	0/0/4 New grou...		1 bit	C	R	-	T	-	switch	Low

The bottom panel shows the Group Monitor controls for Group Address 0/0/1. The Data point type is set to 1.001 switch, and the Value is set to Off. The Delay time is 0 seconds. The Write button is visible. Below the controls is a table of recent telegrams:

Destination	Destination Name	Building Function	Building Part	Hop	Type	DPT	Info
0/0/1	New group address				ValueWrite	1.001 switch	true On
0/0/2	New group address				ValueWrite	1.001 switch	true On
0/0/1	New group address				ValueWrite	1.001 switch	false Off
0/0/2	New group address				ValueWrite	1.001 switch	false Off

3. To switch RELAY2 of the Relay device ON, select the Group Address linked with the "OnOff R2" Group object, and the same thing as the previous step applies.

Software Bill of Materials

This paragraph contains the list of used open source software in this product.

Name	Version	License
Cascoda SDK	0.25	BSD-3-Clause
tinycbor	v0.6.0	MIT
mbedtls	2.16.2	Apache-2.0
Openthread	knx-v1.0.0	BSD-3-Clause

Cascoda SDK

- Description: Cascoda development
- License: BSD-3-Clause
- Version: 0.25
- URL: <https://github.com/Cascoda/cascoda-sdk>
- Notes: Chili2D/S SDK, various drivers

tinycbor

- Description: CBOR implementation
- License: MIT
- Version: v0.6.0
- URL: <https://github.com/intel/tinycbor>
- Notes: used for CBOR encoding/decoding

mbedtls

- Description: security constructs
- License: Apache-2.0
- Version: 2.16.2
- URL: <https://github.com/ARMmbed/mbedtls>
- Notes: used for encryption/decryption

Openthread

- Description: OpenThread, IPv6
- License: BSD-3-Clause
- Version: knx-v1.0.0
- URL: <https://github.com/Cascoda/openthread>
- Notes: Cascoda's port of OpenThread

Data points

url	name	instance	resource	type	interface	type	data type
"/p/1"	OnOff_R1	1	417.61		if.i		DPT_Switch
"/p/2"	InfoOnOff_R1	1	417.51		if.o		DPT_Switch
"/p/3"	OnOff_R2	2	417.61		if.i		DPT_Switch
"/p/4"	InfoOnOff_R2	2	417.51		if.o		DPT_Switch

Parameters

No parameters defined.