Mesh Extender

Installation Manual

Table of contents

List of Figures

List of Tables

1. Mesh Extender Features

The Mesh Extender is essential to build out the Thread Mesh network, without adding additional functionality. The Mesh Extender can easily be added to the Thread network, to overcome dead spots in the network and thus cover a larger area than before. The Mesh Extender reduces dependency on the KNX IoT Hub (Thread border router), it routes all messages on the Thread network, even if the device is an end-device.

Features:

- KNX IoT Device, exposing the logical functions.
- Wireless Thread, router device, using Chili2S with Industry leading receive sensitivity
- Can act as a Thread Leader on the network.
- KNX heartbeat, each x seconds (configurable) the heart beat (CPU temperature) is send out, if the data point is linked.
- Input voltage 5-28 V DC so that works with KNX Bus power, DALI Bus power and USB.
- small size: aprox.: 42mm x 73mm x 12mm (W,L,H)

1.1. Logical functions

The logical functions are:

- OR
- AND
- XOR

The logical functions can have 2 to 5 inputs. The output datapoint will convey the result of the logical function.

The logical functions are implemented as module, 6 modules are available.

2. General information

2.1. 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.

2.2. 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.

Table 1: *Used Terms*

2.3. Safety instructions

Not applicable.

2.4. 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

2.5. Contact information

info@cascoda.com
Threefield House,
Threefield Lane,
Southampton,
SO14 3LP, UK

3. Technical Information

3.1. Storage Conditions

Parameter	Min	Typ	Max	Unit
Storage Temperature	-25		70	°C
Storage Humidity	0		65	%RH

Table 2: *Storage Conditions*

3.2. Operating Conditions

Parameter	Min	Typ	Max	Unit
Operating Temperature	0		50	°C
Supply Voltage	3.1	4.5	5.5	V

Table 3: *Operating Conditions*

3.3. Radio Specification

Protocol: KNX-IoT over Thread

MAC Protocol: IEEE 802.15.4

Configuration: 2.4 GHz, O-QPSK, 250 kbps, Channels 11-26

Parameter	Min	Typ	Max	Unit
Frequency Range	2405		2480	MHz
Transmit Power	0		9	dBm

Table 4: *Thread Radio Information*

4. Physical placement of the Mesh Extender

Since the Mesh Extender is a radio based device, the device should not be put in a metal surrounding. For example a metal filing cabinet.

The device should be in radio range of its parent (router) node.

4.1. Mounting

Attach to the wall/ceiling using four self-tapping flat-headed M3 screws with head not larger than 6mm in diameter & 3mm in height. The ideal length of the screw varies depending on the type of wall it is being installed into, but it must be at least 6mm long. A wall anchor may also be required depending on the wall material.

4.2. Device size

The following diagram is showing the size of the device. The sizes are in [mm].

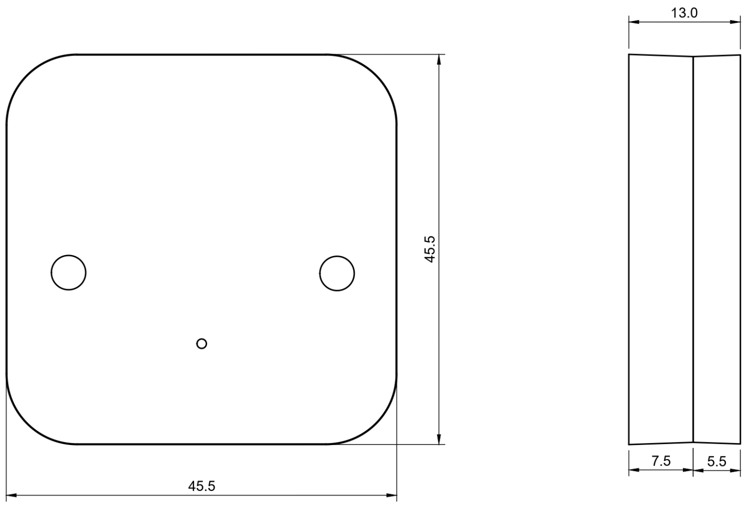

Fig 1: Device Dimensions.

5. Configuration

5.1. Device Startup

The device is start up when it is connected to the power supply.

5.2. Commissioning

Configuration is made using the KNX software as of ETS 6.3 or later. The product file can be downloaded from the ETS online catalogue. In this phase the QR code will be used that is supplied on the casing or in the supplied documentation.

The QR code can be used to for Thread commissioning and KNX commissioning.

> **NOTE!** Thread commissioning needs to be done before KNX commissioning, since this enables the IPV6 communication.

5.2.1. 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.

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

5.2.2. 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 an 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).

5.2.2.1. Downloading the ETS configuration

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

- The parameters are set
- The communication objects are connected

The download can be started in ETS, and one can use either:

- download by serial number
- download per programming mode

The download by serial number does not require any interaction with the device.

The download per programming button requires pressing the programming button. The

PROG button is on the front of the device and needs to be pressed 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 above will start flashing. Disabling the programming mode can achieved by pressing again the **PROG** button.

5.2.2.1.1. Reset of the Mesh Extender

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 achieve 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 achieve 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.

6. 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

Table 5: *Software Bill of Materials*

6.1. 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

6.2. tinycbor

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

6.3. mbedtls

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

6.4. 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

7. KNX device information

Info Field	Value
Manufacturer	cascoda
Model	KNX Mesh Extender
Order_number	0009
Hardware_type	000000000002
Hardware version	[0, 0, 1]
Firmware version	[1, 2, 0]
Sleepy Device	No

7.1. Data points

url	name	instance	resource type	interface type	data type
"/p/Bout_1"	Boolean out Logic 1	1	1007.3072	if.o	DPT_Bool
"/p/B1_1"	B1 Logic 1	1	1007.3081	if.i	DPT_Bool
"/p/B2_1"	B2 Logic 1	1	1007.3082	if.i	DPT_Bool
"/p/B3_1"	B3 Logic 1	1	1007.3083	if.i	DPT_Bool
"/p/B4_1"	B4 Logic 1	1	1007.3084	if.i	DPT_Bool
"/p/B5_1"	B5 Logic 1	1	1007.3085	if.i	DPT_Bool
"/p/Bout_2"	Boolean out Logic 2	1	1007.3072	if.o	DPT_Bool
"/p/B1_2"	B1 Logic 2	1	1007.3081	if.i	DPT_Bool
"/p/B2_2"	B2 Logic 2	1	1007.3082	if.i	DPT_Bool
"/p/B3_2"	B3 Logic 2	1	1007.3083	if.i	DPT_Bool
"/p/B4_2"	B4 Logic 2	1	1007.3084	if.i	DPT_Bool
"/p/B5_2"	B5 Logic 2	1	1007.3085	if.i	DPT_Bool
"/p/Bout_3"	Boolean out Logic 3	1	1007.3072	if.o	DPT_Bool
"/p/B1_3"	B1 Logic 3	1	1007.3081	if.i	DPT_Bool
"/p/B2_3"	B2 Logic 3	1	1007.3082	if.i	DPT_Bool
"/p/B3_3"	B3 Logic 3	1	1007.3083	if.i	DPT_Bool

url	name	instance	resource type	interface type	data type
"/p/B4_3"	B4 Logic 3	1	1007.3084	if.i	DPT_Bool
"/p/B5_3"	B5 Logic 3	1	1007.3085	if.i	DPT_Bool
"/p/Bout_4"	Boolean out Logic 4	1	1007.3072	if.o	DPT_Bool
"/p/B1_4"	B1 Logic 4	1	1007.3081	if.i	DPT_Bool
"/p/B2_4"	B2 Logic 4	1	1007.3082	if.i	DPT_Bool
"/p/B3_4"	B3 Logic 4	1	1007.3083	if.i	DPT_Bool
"/p/B4_4"	B4 Logic 4	1	1007.3084	if.i	DPT_Bool
"/p/B5_4"	B5 Logic 4	1	1007.3085	if.i	DPT_Bool

Table 6: *Data points*

7.2. Parameters

url	name	param type
"/p/pLE_1"	Logic Enable1	int
"/p/pLO_1"	LogicOperation1	int
"/p/pLP_1"	Polarity1	int
"/p/pLNR_1"	Inputs1	int
"/p/pLE_2"	Logic Enable2	int
"/p/pLO_2"	LogicOperation2	int
"/p/pLP_2"	Polarity2	int
"/p/pLNR_2"	Inputs2	int
"/p/pLE_3"	Logic Enable3	int
"/p/pLO_3"	LogicOperation3	int
"/p/pLP_3"	Polarity3	int
"/p/pLNR_3"	Inputs3	int
"/p/pLE_4"	Logic Enable4	int
"/p/pLO_4"	LogicOperation4	int
"/p/pLP_4"	Polarity4	int
"/p/pLNR_4"	Inputs4	int

Table 7: *Parameters*