

# ZIGBEE HEATING REGULATOR H10

## REFERENCE MANUAL



# Table of Contents

1. Overview	1
2. Features	2
3. Installation	6
3.1. Professional installation	6
4. Initial Device Start-Up	7
5. Man-Machine interface (MMI)	8
5.1. Common ubisys zigbee MMI	8
5.2. Power-Cycle Sequencing Factory Reset	9
5.3. Specific MMI for H10 Manual Operation and Output Level Monitoring	9
6. Zigbee Interface	11
6.1. Device Anatomy	11
6.2. Installation Code	13
6.3. Application Endpoint #0 - Zigbee Device Object	14
6.4. Application Endpoint #1 - Thermostat	15
6.4.1. Basic Cluster (Server)	15
6.4.2. Identify Cluster (Server)	15
6.4.3. Groups Cluster (Server)	16
6.4.4. Scenes Cluster (Server)	16
6.4.5. Thermostat Cluster (Server)	17
6.4.6. Thermostat Cluster (Client)	20
6.4.7. Temperature Measurement Cluster (Client)	20
6.4.8. Relative Humidity Measurement Cluster (Client)	20
6.4.9. Occupancy Sensing Cluster (Client)	20
6.5. Application Endpoint #2 - Thermostat	20
6.6. Application Endpoint #3 - Thermostat	21
6.7. Application Endpoint #4 – Thermostat	21
6.8. Application Endpoint #5 – Thermostat	21
6.9. Application Endpoint #6 – Thermostat	21
6.10. Application Endpoint #7 – Thermostat	21
6.11. Application Endpoint #8 – Thermostat	21
6.12. Application Endpoint #9 – Thermostat	22
6.13. Application Endpoint #10 – Thermostat	22
6.14. Application Endpoint #11 – Heating/Cooling Unit	22
6.14.1. Basic Cluster (Server)	23
6.14.2. Identify Cluster (Server)	23
6.14.3. Groups Cluster (Server)	23
6.14.4. Scenes Cluster (Client)	23
6.14.5. On/off Cluster (Client)	24
6.14.6. Level Control Cluster (Server)	25
6.15. Application Endpoint #12 – Heating/Cooling Unit	27

6.16. Application Endpoint #13 – Heating/Cooling Unit	27
6.17. Application Endpoint #14 – Heating/Cooling Unit	27
6.18. Application Endpoint #15 – Heating/Cooling Unit	27
6.19. Application Endpoint #16 – Heating/Cooling Unit	27
6.20. Application Endpoint #17 – Heating/Cooling Unit	28
6.21. Application Endpoint #18 – Heating/Cooling Unit	28
6.22. Application Endpoint #19 – Heating/Cooling Unit	28
6.23. Application Endpoint #20 – Heating/Cooling Unit	28
6.24. Application Endpoint #21 – Master Control	28
6.24.1. Basic Cluster (Server)	29
6.24.2. Identify Cluster (Server)	29
6.24.3. Temperature Measurement Cluster (Client)	29
6.25. Application Endpoint #232 - Device Management	29
6.25.1. Basic Cluster (Server)	30
6.25.2. Identify Cluster (Client)	30
6.25.3. Commissioning Cluster (Server)	31
6.25.4. Over-the-Air Upgrade Cluster (Client)	31
6.26. Application Endpoint #242 - Zigbee Green Power	31
6.26.1. Green Power Cluster (Server)	31
6.26.2. Green Power Cluster (Client)	33
7. H10-B Base Module	35
7.1. Theory of Operation	35
7.1.1. Proportional/Integral (PI) Control Loop	36
7.2. Using H10 for Electrical Underfloor Heating	36
8. H10-XP Circulation Pump Extension Module	37
9. H10-XS Heating and Cooling Demand Extension Module	38
10. H10-XI Legacy Wired Thermostats Interface Extension Module	39
11. H10-XW 1-Wire® Bridge Extension Module	40
11.1. Supported Sensors	40
12. Physical Dimension	42
13. Ordering Information	44
14. General Terms & Conditions Of Business	45
15. Declaration Of Conformity	46
16. Revision History	47
17. Contact	48

## 1. Overview

Thank you for purchasing the ubisys heating regulator H10, a 10-fold thermostat based on PI closed-loop control with integrated PWM output controllers for heating and/or cooling in two-pipe or four-pipe systems.

You have decided for a high-quality product with first-rate support!

This reference manual provides operating and maintenance instructions, interface specifications, command references and more. It is primarily intended for system integrators, not end-users.

An installation guide specific to your region is available as a separate document and is included in printed form in the product package.

If you have any questions or need additional support, please visit the support pages that best fit your background:

If you are a consumer (private household) or installer, please visit the Smart Home support pages at <http://www.ubisys.de/en/smarthome/support.html> for contact details.

As a commercial customer, please visit the Engineering support pages at <https://www.ubisys.de/en/business-customers/> for contact details.

## 2. Features

- Advanced zigbee thermal regulator for up to 10 individual heating and/or cooling zones
- Modular design – perfectly suited for adapting to customer's heating infrastructure and retrofitting:
  - H10/230-B base module with zigbee interface for direct drive of up to 10 230V~ valves
  - H10/24-B base module with zigbee interface for direct drive of up to 10 24V= valves
  - H10-XP extension module for circulation pump control, 2 motor control outputs 230V~/500VA
  - H10-XS extension module for heating and cooling demand, 2 SPDT outputs 230V/1,800VA
  - H10-XW extension module for connecting a multitude of 1-Wire® temperature sensors
    - 10 forward flow temperatures, or a single global forward flow temperature
    - 10 return flow temperatures
    - 10 class A local (room) temperatures – e.g. wall-mount sensor
    - 10 class B local (room) temperatures – e.g. in-floor sensor
    - 1 outdoor temperature
  - H10/230-XI extension module for connecting legacy 230V~ in-wall thermostats
  - H10/24-XI extension module for connecting legacy 24V= in-wall thermostats
- Base module H10-B for direct drive of thermoelectric radiator valves:
  - either 10 heating or cooling zones (one valve control output per zone, two-pipe system)
  - or 5 heating and cooling zones (two valve control outputs per zone, four-pipe system)
  - advanced PI control loop to determine heating demand and cooling demand
  - available in two flavours: H10/230-B for 230V~/50Hz valves, H10/24-B for 24V valves
  - H10/230-B incorporates universal power supply: 85V...264V, 47Hz...440Hz
  - comprises a powerful zigbee 3.0 interface with routing functionality
  - features a real-time clock with back-up battery for autonomous schedules
  - works with temperature, occupancy or combined environmental sensors, both “classic” and energy-harvesting zigbee green power <sup>[1]</sup>sensors
  - Can work in conjunction with other thermostats in a leader/follower setup; specifically, compatible with Legrand's in-wall thermostat UI Model No. 0 663 40
- Different operating modes:
  - Under normal operating conditions thermostat set-points, occupancy information and temperature sensors are used for closed-loop control of the temperature
  - When sensor reports are missing the device enters back-up mode and applies default valve opening values with different pre-sets for summer and winter seasons
  - Automatic temperature regulation of the built-in thermostat can be overruled by setting arbitrary steering values per valve output, e.g. 10%, 50% or 100% using zigbee level steering
  - Pass-through mode allows legacy wired in-wall thermostats to control valve outputs, i.e. the valve behaves like it was directly wired to the legacy thermostat (requires H10-XI extension)

- Manual mode allows valve control using a button (bypassing valve settings determined by either the zigbee thermostat, legacy thermostats, or zigbee level steering)
- Supports groups, scenes, bindings, and reporting
- Reliable, unattended, maintenance-free operation; designed for DIN rail mounting in hot water/cold water underfloor heating distribution cabinets
- H10/230-B, H10/24-B and H10-XP employ solid state switching for highest efficiency and durability
- H10/230-B and H10/24-B employ staggered switching of valves to spread excessive valve turn-on currents over time
- Made in Germany using high-quality, enduring parts for many years of life expectancy
- Low power dissipation: 0.3W
- On-board MCU: Advanced 32-bit ARM Cortex-M4 running at 48MHz with 128KB SRAM
- On-board PHY: Texas Instruments CC2520, 5dBm transmit power, -98dBm receiver sensitivity
- On-board inverted-F antenna
- Leverages ubisys IEEE 802.15.4 MAC and ZigBee stack for best-in-class reliability and performance
- Supports all channels in the 2.4 GHz band, i.e. channels 11-26 as per IEEE 802.15.4: Primary = { 11, 15, 20, 25 }; Secondary = { 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26 }
- Supports joining centralized and distributed security networks as router
- Supports forming simple centralized security networks as Coordinator and Trust Center
- Supports forming distributed security networks as router
- Three pre-configured Trust Center Link-Keys for joining:
  - Global Default Trust Center Link-Key (“ZigBeeAlliance09”)
  - ZigBee 3.0 Global Distributed Security Link-Key <sup>[2]</sup>
  - Device-individual link-key derived from installation code – also printed as text and QR barcode
- Extended neighbour table with up to 78 entries for routers and end-devices - more than three times the capacity required by the standard (25)
- Extended routing table with up to 96 entries for ad hoc and many-to-one routes - nearly ten times the capacity required by the standard (10)
- Extended buffering for sleeping end-devices with up to 24 buffers - 24 times the capacity required by the standard (1)
- Extended APS duplicate rejection table with up to 64 slots - 64 times the capacity required by the standard (1)
- Extensive transmit and receive queues for optimum through-put and minimum packet drop rate
- Reliable and scalable network-wide broadcasts featuring passive acknowledgments
- Reliable packet forwarding with automatic network-level retries
- Very sophisticated routing algorithm for reliable ad hoc routing – avoids routing loops even in case of concurrent route requests with overlapping source/destination
- Firmware upgradable over-the-air during normal operation using ZigBee OTA Upgrade Cluster

- Man-Machine-Interface
  - network steering, factory reset etc. using common ubisys 1-button + 1-LED interface
  - separate push-button, 1 red/green and 10 channel LEDs for manual operation and monitoring
- Flame retardant housing (V-0); black, RAL 9005

[1] H10-B is a Green Power Combined device with proxy and sink (including compact reporting feature)

[2] Currently, pre-certification key (D0...:DF)



## 3. Installation

### 3.1. Professional installation

Please refer to the hardware installation guide, which was included in the product package. It provides detailed installation instructions. Utilizing the services of a professional installer/electrician is imperative to avoid health hazards and damages to your heating/cooling infrastructure.

## 4. Initial Device Start-Up

The device will search for an open ZigBee network to join when you first apply power to it.

Open the ZigBee network for joining and then power-up the device.

H10 will blink quickly to indicate a search in progress. Once a search cycle is complete it will either blink five times slowly to indicate it has joined a network, or blink three times quickly to indicate a joining failure – for example no network found, not permitted onto the network, etc. It will continue to search for a network in case of failure.

Subsequently, when the device is power-cycled, it will blink five times slowly to indicate it is operating as a router on the network, ten times slowly to indicate that it is operating as a coordinator and trust center or blink quickly to indicate it is searching for a network to join. When commissioned, the router's LED will remain off after the five/ten blink cycles during normal operation. Afterwards, it will turn on only for as long as it is permitting other devices to join the network.

After reboot, if the device is properly commissioned, it resumes operation doing a "silent rejoin", i.e. it does not broadcast a device announcement in this case. After joining a network, the device will prolong the joining window by three minutes via a fresh ZDO permit joining request.



When the device is power-cycled, its thermostat endpoints are likely to not know enough about the heating/cooling zone conditions to immediately resume where they left off before the power-down. Give the device a few moments until sensor reports arrive, H10 has gathered enough insight into heating/cooling space conditions, and the control loop kicks in. Until then, the device will operate in back-up mode.

## 5. Man-Machine interface (MMI)

### 5.1. Common ubisys zigbee MMI

H10 offers a push-button behind a tiny hole in the front-face of the device, as well as a LED right next to it, i.e. the common MMI shared by most ubisys zigbee products. This man-machine interface provides access to a menu. In addition, it provides a handy shortcut to factory reset the device <sup>[3]</sup>.



To factory reset the device, keep the button pressed for approximately 10 seconds until the LED starts to flash. This is equivalent to selecting menu item #5.

Notice that the device also supports a power-cycle sequencing factory reset as detailed in [Section 5.2](#)

To enter the menu, press and hold the push-button for more than a second until you see three short flashes followed by a sequence of one blink, pause, one blink, pause, etc. This indicates that you have successfully entered the menu. With each short button press (less than a second), you advance through the menu. For example, pressing the button once brings you to the second menu item, which H10 indicates by two blinks, pause, two blinks pause, etc. Once you have reached the menu item that you want to run, press and hold the button for more than a second to execute the selected item. The exact meaning of each menu item depends on the particular device and may vary from firmware version to firmware version.

This is the menu for the Heating Regulator H10:

Menu Item #	Operation
1	<b>Zigbee Commissioning: Network Steering</b> While in this mode a single press on the button instigates Zigbee Network Steering ("EZ-mode"). If the device is on a network it will open the network for new devices for three minutes, otherwise it will attempt to join an open network. If the network is already open, the device will close the network instead such that repetitive presses toggle the network permit joining state. The LED is on, when the network is open for new devices, and off otherwise.
2	<b>Zigbee Commissioning: Finding &amp; Binding</b> Instigates the Zigbee Finding & Binding procedure ("EZ-mode") on an initiator or target endpoint. Target endpoints will enter identify mode for three minutes. Initiator endpoints will query targets and create suitable bindings to these targets. After leaving the menu in this mode, the first button press starts the selection of the endpoint and each subsequent press (within half a second) increments the endpoint number. Once the desired endpoint is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled.
3	<b>Zigbee Commissioning: Clear Bindings</b> Clears bindings on an initiator endpoint. Select this menu item and leave menu mode. Afterwards the device expects the selection of an initiator endpoint. The first button press starts the selection of the endpoint and each subsequent press (within half a second) increments the endpoint number. Once the desired endpoint is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. This mode is active for one round only. Afterwards the device reverts to mode #1.
4	<b>Zigbee Commissioning: Set Device Role and Factory Reset</b> Selects the Zigbee device role for this device on the network, resets the remaining settings to factory defaults and restarts the device. Select this menu item and leave menu mode. Afterwards the device expects the selection of an option. The first button press starts the selection of the option and each subsequent press (within half a second) increments the option number. Once the desired option is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. The following options are available: Option #1: Join an existing Zigbee network as router Option #2: Form a new distributed security network as the first router Option #3: Form a new centralized security network as the coordinator and trust center

Menu Item #	Operation
5	<p><b>Factory Reset</b></p> <p>Put the device into the same state as it was when it left the factory, then reboot. The only exception is the outgoing network security frame counter, which is preserved across factory resets. The device will broadcast a network leave indication.</p>
6	<p><b>Zigbee Commissioning: Advanced Commands</b></p> <p>Provides a number of advanced Zigbee commissioning command options. Select this menu item and leave menu mode. Afterwards the device expects the selection of an option. The first button press starts the selection of the option and each subsequent press (within half a second) increments the option number.</p> <p>Once the desired option is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. The following options are available:</p> <p>Option #1: Perform a simple reset (reboot), then continue operating seamlessly (silent re-join)</p> <p>Option #2: Perform a simple reset (reboot), then re-join the network</p> <p>Option #3: Perform a full factory reset, including security frame counters and preserved settings</p>
7	Reserved for internal use. Do not leave the menu in this mode.

## 5.2. Power-Cycle Sequencing Factory Reset

It is possible to instigate a factory reset using a special power-cycle sequence. This is equivalent to selecting menu item #5, with the advantage that you need no access to the device itself (only to its power supply).

1. Power the device for at least four seconds.
2. Interrupt the power supply for at least a second.
3. Reapply power for less than two seconds but more than half a second.  
Notice that at the end of this cycle, the device is off and should remain off for at least a second.
4. Repeat the previous step two more times, for a total of three short power cycles.
5. Apply power to the device and leave it powered on.

The device will now factory reset and reboot.

After above sequence, the dimmer will flash the connected light three times to indicate the factory reset sequence is in progress. This is the default reset sequence, equivalent to selecting menu item #5 as described above, or keeping the menu button pressed for more than 10 seconds, or removing the device from the network using a leave request.

## 5.3. Specific MMI for H10 Manual Operation and Output Level Monitoring

One operating button allows manual operation of the device without network connectivity. This can be useful during shell construction of a building when the zigbee network is not set-up yet, or during other network down-times due to device failures, maintenance, relevant devices running out of batteries etc.

A dedicated LED shows overall system status by turning green or red and ten green LEDs for each valve output are used to show current valve opening levels. The green status LED flashes once every minute and starts a time frame used to depict the valve opening level. The time the green status LED remains on equals 100% and each channel LED turns on for a fraction of this time, which is proportional to the duty cycle of the PWM signal the channel is generating.

To enter emergency mode, keep the right-hand button pressed for a while until the red status LED illuminates. ☒

[3] The 10-second-press factory reset (equivalent to menu item #5) is available since firmware version 1.06.

## 6. Zigbee Interface

Please refer to the following IEEE and ZigBee Alliance documents, which apply to this product:

[R1] IEEE Standard 802 - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)

[R2] ZigBee Specification, Revision 21, Document No. 05-3474-21

[R3] ZigBee 2015 Layer PICS and Stack Profiles, Revision 6, Document No. 08-0006-06

[R4] ZigBee Cluster Library Specification, Revision 5, Document No. 07-5123-05

[R5] ZigBee Base Device Behavior Specification, Revision 13, Document No. 13-0402-13

[R6] ZigBee PRO Green Power Feature Specification, Revision 26, Document No. 09-5499-26

[R7] ZigBee Home Automation Public Application Profile 1.2, Revision 29, Document No. 05-3520-29

[R8] ZigBee Smart Energy Standard 1.1b, Revision 18, Document No. 07-5356-18

### 6.1. Device Anatomy

The heating regulator base module H10-B provides ten individual PWM outputs for a rated voltage of 230V~/50Hz and an apparent power of up to 10VA per channel (model H10/230-B) or 24V= and a power of 5W per channel (model H10/24-B), which allows utilizing it for typical use cases in EMEA countries, in particular hot-water underfloor heating in Northern Europe. It combines state-of-the-art solid-state technology with advanced firmware for a unique feature set. Being a mains-powered device, H10-B also acts as a zigbee router improving network connectivity and meshing capabilities as well as serving as a connection point (“parent”) for sleeping end-devices and green power devices (“proxy”).

H10 provides 24 zigbee application endpoints:

Endpoint #	Profile	Application
0 (0x00)	0x0000: ZigBee Device Profile	ZigBee Device Object (ZDO) – standard management features
1 (0x01)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #1 (and output #10 when configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
2 (0x02)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #2 (and output #9 when configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
3 (0x03)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #3 (and output #8 when configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
4 (0x04)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #4 (and output #7 when configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
5 (0x05)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #5 (and output #6 when configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.

Endpoint #	Profile	Application
6 (0x06)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #6 (unless the thermostat on endpoint 5 is configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
7 (0x07)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #7 (unless the thermostat on endpoint 4 is configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
8 (0x08)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #8 (unless the thermostat on endpoint 3 is configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
9 (0x09)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #9 (unless the thermostat on endpoint 2 is configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
10 (0x0A)	0x0104: Common Profile (HA)	Thermostat (0x0301). This endpoint provides temperature regulation and controls valve output #10 (unless the thermostat on endpoint 1 is configured for a four-pipe system). It supports groups and scenes. The identify cluster allows for push-button commissioning as a target. In addition, the thermostat client allows following another thermostat, temperature sensor client allows it to be bound to a local temperature sensor and occupancy allows it to be bound to an occupancy sensor.
11 (0x0B)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #1, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
12 (0x0C)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #2, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
13 (0x0D)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #3, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
14 (0x0E)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #4, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
15 (0x0F)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #5, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
16 (0x10)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #6, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
17 (0x11)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #7, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
18 (0x12)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #8, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
19 (0x13)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #9, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.

Endpoint #	Profile	Application
20 (0x14)	0x0104: Common Profile (HA)	Heating/Cooling Unit (0x0300). This endpoint provides direct steering of valve output #10, bypassing any thermostat control. The on/off and level control cluster on this endpoint can be used to set a specific duty cycle. It supports groups and scenes. The identify cluster allows for push-button commissioning as a target.
21 (0x15)	0x0104: Common Profile (HA)	Combined Interface (0x0007). This endpoint serves as a master control to coordinate output channels and also accepts reports from an optional outdoor temperature sensor. The outdoor temperature is made available to all thermostat endpoints. The identify cluster allows for push-button commissioning as a target.
232 (0xE8)	0x0104: Common Profile (HA)	Device Management (0x0507)
242 (0xF2)	0xA1E0: Green Power Profile	ZigBee Green Power Combined Proxy and Sink

The ubisys ZigBee manufacturer ID is 0x10F2. This manufacturer code is used to identify OTA upgrade images, for accessing manufacturer-specific ZCL attributes and commands etc.

## 6.2. Installation Code

This heating controller has a pre-configured link key, which is derived from the installation code printed on the housing in text format and as a two-dimensional barcode (QR code). The format specified in [R6], section 5.4.8.1.1 is used with a full 128-bit installation code + 16-bit CRC.

Earlier production lots used a proprietary QR code format, which contained additional information as illustrated in the following example:

```
1 ubisys2/R0/001FEE00000000FF/0F7C1CD805F91649EBA84580AA1CB432F51A/21
```

Here, “ubisys2” is the data format identifier, “R0” is the model string (this is just an example, it would be “H10” for this product), followed by the EUI-64, the installation code, and a checksum that covers the entire information (including model, EUI-64 and installation code), all separated by a single dash character ('/'). The check sum is an unsigned 8-bit integer, which is calculated by performing a byte-wise exclusive-or (XOR, “⊗”) of the ASCII characters of the model string, the binary representation of the EUI-64 (in big endian format), and the binary representation of the install code.

For the example above, this calculation yields:

```
1 52 ('R') ⊗ 30 ('0') ⊗
2 00 ⊗ 1F ⊗ EE ⊗ 00 ⊗ 00 ⊗ 00 ⊗ 00 ⊗ FF ⊗
3 0F ⊗ 7C ⊗ 1C ⊗ D8 ⊗ 05 ⊗ F9 ⊗ 16 ⊗ 49 ⊗ EB ⊗ A8 ⊗ 45 ⊗ 80 ⊗ AA ⊗ 1C ⊗ B4 ⊗ 32 ⊗ F5
   ⊗ 1A = 21
```



The data format “ubisys1” is identical to “ubisys2” except for the check sum, which is present, but invalid. If you encounter “ubisys1” labels, then do not verify the trailing check sum field.

With availability of a standard recommended zigbee QR code format, all ubisys products are now using the recommended format. For H10, specifically, it is currently structured as follows:

```
1 Z:001FEE0000003605$I:E09F774092FB1AF6A6C42B288DC813CBB762%G#M:H10
```



### 6.3. Application Endpoint #0 - Zigbee Device Object

Please refer to the ZigBee Specification [R2] for details on the ZigBee Device Object (ZDO) and the protocol used for over-the-air communication, called the ZigBee Device Profile (ZDP). Notice that the ZDP is fundamentally different from typical application endpoints, which build on the ZigBee foundation framework and the ZigBee Cluster Library (ZCL).

The ubisys ZigBee H10 supports the following ZDO services:

Primitive	Description
nwk_addr_req/ nwk_addr_rsp	<b>Network address request/response</b> Translates a 64-bit IEEE address into a 16-bit network short address. Use only when really required, because this message employs a network-wide broadcast (flooding)
ieee_addr_req/ ieee_addr_rsp	<b>IEEE address request/response</b> Translates a 16-bit network short address into a 64-bit IEEE address.
node_desc_req/ node_desc_rsp	<b>Node descriptor request/response</b> Returns information such as the manufacturer ID, power supply, etc.
power_desc_req/ power_desc_rsp <sup>[4]</sup>	<b>Power descriptor request/response</b> Returns information such as the power source and mode.
active_ep_req/ active_ep_rsp	<b>Active endpoints request/response</b> Returns a set of available application endpoints on the device.
simple_desc_req/ simple_desc_rsp	<b>Simple descriptor request/response</b> Returns a descriptor for a certain application endpoint with a list of available services (clusters).
match_desc_req/ match_desc_rsp	<b>Match descriptor request/response</b> Searches for a certain cluster or set of clusters and returns the matching endpoints, if any.
device_annce	<b>Device announcement</b> Advertises the presence of a new device in the network.
parent_annce/ parent_annce_rsp <sup>[5]</sup>	<b>Parent announcement/response</b> This is part of the Zigbee 2015 end-device child management feature.
system_server _discovery_req/ system_server _discovery_rsp <sup>[6]</sup>	<b>System server discovery request/response</b> Provides the ability to discover system servers, in particular the network manager.
bind_req/ bind_rsp	<b>Bind request/response</b> Creates an application binding
unbind_req/ unbind_rsp	<b>Unbind request/response</b> Removes an application binding
mgmt_nwk_disc_req/ mgmt_nwk_disc_rsp <sup>[7]</sup>	<b>Management: Network discovery request/response</b> Instructs the device to perform a network discovery and report the results back.
mgmt_lqi_req/ mgmt_lqi_rsp	<b>Management: Neighbor table request/response</b> Returns information about neighboring devices, including the link quality, device type etc.
mgmt_rtg_req/ mgmt_rtg_rsp	<b>Management: Routing table request/response</b> Returns information about routes established on the device.
mgmt_bind_req/ mgmt_bind_rsp	<b>Management: Binding table request/response</b> Returns information about application bindings on the device.
mgmt_leave_req/ mgmt_leave_rsp	<b>Management: Leave request/response</b> Makes the device leave the network or removes one of its end-device children.
mgmt_permit_joining _req/ mgmt_permit_joining _rsp	<b>Management: Permit joining request/response</b> Opens the network for new devices to join.
mgmt_nwk_update _req/ mgmt_nwk_update _notify <sup>[7]</sup>	<b>Management: Network update request/response/notification</b> Performs energy scans, instigates a channel change or assigns the network manager.

## 6.4. Application Endpoint #1 - Thermostat

Use this application endpoint in (I) a two-pipe system to control the heating or cooling unit connected to output #1, or (II) in a four-pipe system to control the heating unit connected to output #1, and the cooling unit connected to output #10.

This is a Finding & Binding target endpoint.

The application endpoint exposes the following clusters:

Cluster	Direction	Description
0x0000	Inbound (Server)	<b>Basic</b> Provides basic information about the device, such as the manufacturer ID, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network.
0x0003	Inbound (Server)	<b>Identify</b> Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding.
0x0004	Inbound (Server)	<b>Groups</b> Allows adding this endpoint to one or more groups. Afterwards the endpoint can be addressed using the group address. This is also a prerequisite for scenes. You may also query group membership and delete group associations.
0x0005	Inbound (Server)	<b>Scenes</b> Allows storing one or more scenes per group, where each scene consists of a pre-set on/off state value. You may either store the current values as a scene, or provide scene settings when adding a scene, or delete scenes.
0x0201	Inbound (Server)	<b>Thermostat</b> Provides temperature control.
0x0201	Outbound (Client)	<b>Thermostat</b> Receives reports from other thermostats.
0x0402	Outbound (Client)	<b>Temperature Measurement</b> Receives reports from temperature sensors.
0x0405	Outbound (Client)	<b>Relative Humidity Measurement</b> Receives reports from relative humidity sensors.
0x0406	Outbound (Client)	<b>Occupancy Sensing</b> Receives reports from occupancy sensors.

### 6.4.1. Basic Cluster (Server)

Please refer to [Section 6.25.1](#) for details.

### 6.4.2. Identify Cluster (Server)

In identify mode, the channel LED will be toggled once every second, the output itself is not affected.

Attributes supported:

Attribute	Type	Description
0x0000	unsigned16	<b>IdentifyTime</b> The remaining time, in seconds, the device will be identifying itself.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Identify</b> Puts the device into identify mode, or terminates identify mode.
0x01	<b>Query Identify</b> Determines whether the device is currently identifying, and returns the remaining time if so.

### 6.4.3. Groups Cluster (Server)

Attributes supported:

Attribute	Type	Description
0x0000	bitmap8, read-only	<b>NameSupport</b> Always set to 0: This implementation does not support storing names for groups.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Add Group</b> Adds the endpoint to a group.
0x01	<b>View Group</b> Determines whether the device belongs to a group and returns the group name, if supported.
0x02	<b>Get Group Membership</b> Returns the set of groups this endpoint belongs to.
0x03	<b>Remove Group</b> Removes this endpoint from the specified group. Also removes all scenes that refer to this group.
0x04	<b>Remove All Groups</b> Removes this endpoint from all groups. Also removes all scenes that refer to any of the existing groups.
0x05	<b>Add Group if Identifying</b> Adds this endpoint to the group, if the endpoint is identifying.

### 6.4.4. Scenes Cluster (Server)

Attributes supported:

Attribute	Type	Description
0x0000	unsigned8, read-only	<b>SceneCount</b> Holds the total number of scenes (across all groups) currently stored on the device.
0x0001	unsigned8, read-only	<b>CurrentScene</b> If the SceneValid attribute is true, this attribute, together with the CurrentGroup attribute, indicates the currently active scene.
0x0002	unsigned8, read-only	<b>CurrentGroup</b> If the SceneValid attribute is true, this attribute, together with the CurrentScene attribute, indicates the currently active scene.
0x0003	bool, read-only	<b>SceneValid</b> If true, the scene identified by CurrentGroup and CurrentScene is currently active, i.e. all device attribute values match the values in the scene field set.
0x0004	bitmap8, read-only	<b>NameSupport</b> Always set to 1: This implementation does support storing names for scenes.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Add Scene</b> Adds a scene with or without a scene field set
0x01	<b>View Scene</b> Returns the scene field set, name and transition times for a scene.
0x02	<b>Remove Scene</b> Removes a scene from the scene table.
0x03	<b>Remove All Scenes</b> Removes all scenes that belong to a particular group.
0x04	<b>Store Scene</b> Stores the device's current state as a scene or updates a previously stored scene accordingly.
0x05	<b>Recall Scene</b> Reverts the device's current state using the values from the previously stored field set.
0x06	<b>Get Scene Membership</b> Returns the set of scenes (within the scope of the specified group) currently stored on the device.
0x40	<b>Enhanced Add Scene</b> Similar to Add Scene with higher transition time resolution (10ths of a second instead of once second)
0x41	<b>Enhanced View Scene</b> Similar to View Scene with higher transition time resolution (10ths of a second instead of once second)
0x42	<b>Copy Scene</b> Copies a scene in a single operation as an alternative to (Enhanced) View Scene, (Enhanced) Add Scene

#### 6.4.5. Thermostat Cluster (Server)



This cluster uses the binding table for managing reporting targets.

Attributes supported:

Attribute	Type	Description
0x0000	signed16, read-only, reportable	<b>LocalTemperature</b> Indicates the local room temperature in the heating/cooling zone. This might be measured using remote zigbee temperature or environmental
0x0001	signed16, read-only, reportable	<b>OutdoorTemperature</b> Indicates the outdoor temperature in the heating/cooling zone. This might be measured using remote zigbee temperature or environmental
0x0002	bool, read-only, reportable	<b>Occupancy</b> Determines whether the heated/cooled space is occupied or not, as measured locally or remotely (over the network).
0x0003	signed16, read-only	<b>AbsoluteMinimumHeatingSetpoint</b> Indicates the lowest permissible value for the occupied and unoccupied heating setpoints. In anti-freeze mode, this is also the value being used to control the temperature.
0x0004	signed16, read-only	<b>AbsoluteMaximumHeatingSetpoint</b> Indicates the highest permissible value for the occupied and unoccupied heating setpoints.
0x0005	signed16, read-only	<b>AbsoluteMinimumCoolingSetpoint</b> Indicates the lowest permissible value for the occupied and unoccupied cooling setpoints.
0x0006	signed16, read-only	<b>AbsoluteMaximumCoolingSetpoint</b> Indicates the highest permissible value for the occupied and unoccupied cooling setpoints.
0x0007	unsigned8, read-only, reportable	<b>PICoolingDemand</b> Quantifies the amount of cooling required from a cooling device in order to reach the active cooling setpoint, as determined by the PI control loop. This is a steering value in the range 0...100%.
0x0008	unsigned8, read-only, reportable	<b>PIHeatingDemand</b> Quantifies the amount of heating required from a heating device in order to reach the active heating setpoint, as determined by the PI control loop. This is a steering value in the range 0...100%.
0x0011	signed16, reportable	<b>OccupiedCoolingSetpoint</b> Specifies the cooling mode setpoint when the room is occupied.
0x0012	signed16, reportable	<b>OccupiedHeatingSetpoint</b> Specifies the heating mode setpoint when the room is occupied.

Attribute	Type	Description
0x0013	signed16, reportable	<b>UnoccupiedCoolingSetpoint</b> Specifies the cooling mode setpoint when the room is unoccupied.
0x0014	signed16, reportable	<b>UnoccupiedHeatingSetpoint</b> Specifies the heating mode setpoint when the room is unoccupied.
0x0019	signed8, read-only	<b>MinimumSetpointDeadband</b> Minimum dead-band, in units of 0.1°C, between related heating and cooling setpoints. The respective setpoints must be separated by at least this value. The dead-band prevents heating and cooling units from working against each other around the desired target temperature. This dead-band can be considered a comfort zone, i.e. the thermostat will not heat or cool the space as long as the local temperature resides within the bounds set forth by the applicable heating and cooling setpoints.
0x001b	enum8, reportable	<b>ControlSequenceOfOperation</b> Specifies the overall operating environment of H10, and thus the possible system modes that H10 can operate in.
0x001c	enum8, reportable	<b>SystemMode</b> Specifies the current operating mode of the thermostat. H10 supports OFF, AUTO, COOL, HEAT and EMERGENCY HEATING system mode options.
0x001e	enum8, reportable	<b>RunningMode</b> The running mode of the thermostat, i.e. the actual mode it is currently in, which can be either OFF, COOL or HEAT.
0xFFFF	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

#### Manufacturer-specific attributes supported:

Attribute	Type	Description
0x0000	signed8, persistent, singleton	<b>ClassBTemperatureOffset</b> A temperature offset applied to "class B" local temperature readings.
0x0001	signed8, persistent, singleton	<b>ReturnFlowTemperatureWeight</b> Allows to specify the weight of the return flow temperature vs. forward flow temperature when estimating the local temperature using flow temperatures.
0x0002	structure, read-only	<b>RawOutdoorTemperature</b> A structure, where the first field is the outdoor temperature and the second is the corresponding time-stamp.
0x0003	structure, read-only	<b>RawLocalTemperatureA</b> A structure, where the first field is the "class A" local temperature and the second is the corresponding time-stamp.
0x0004	structure, read-only	<b>RawLocalTemperatureB</b> A structure, where the first field is the "class B" local temperature and the second is the corresponding time-stamp.
0x0005	structure, read-only	<b>RawForwardFlowTemperature</b> A structure, where the first field is the forward flow temperature and the second is the corresponding time-stamp.
0x0006	structure, read-only	<b>RawReturnFlowTemperature</b> A structure, where the first field is the return flow temperature and the second is the corresponding time-stamp.
0x0007	bitmap64, read-only	<b>InstalledExtensions</b> A set of flags, each indicating whether a specific option, i.e. extension module, is installed and available.

#### Cluster commands supported:

Command	Description
0x00	<b>Setpoint Raise/Lower</b> Receives this command from a client to increase or decrease the indicated setpoint(s) with the specified amount, in steps of 0.1°C.

### 6.4.5.1. InstalledExtensions Attribute

This attribute provides a set of flags, each indicating whether a specific option, i.e. extension module or wired sensor, is installed and available. Extension modules are not hot-pluggable, they are only installed while the H10-B base module is not powered; hence, options are enumerated only once after power up.

Option	Bit #	Description
H10-XP	0	Circulation Pump Control Extension Module installed
H10-XS	1	Heating/Cooling Demand Extension Module installed
H10-XI	2	Wired Thermostat Interface Extension Module installed
H10-XW	7	1-Wire® Bridge Extension Module installed
H10-XW-O	8	Outdoor temperature sensor installed
H10-XW-F	9	Forward flow temperature sensor, applicable to all channels, installed
H10-XW-F01	10	Forward flow temperature sensor for channel #1 installed
H10-XW-F02	11	Forward flow temperature sensor for channel #2 installed
H10-XW-F03	12	Forward flow temperature sensor for channel #3 installed
H10-XW-F04	13	Forward flow temperature sensor for channel #4 installed
H10-XW-F05	14	Forward flow temperature sensor for channel #5 installed
H10-XW-F06	15	Forward flow temperature sensor for channel #6 installed
H10-XW-F07	16	Forward flow temperature sensor for channel #7 installed
H10-XW-F08	17	Forward flow temperature sensor for channel #8 installed
H10-XW-F09	18	Forward flow temperature sensor for channel #9 installed
H10-XW-F10	19	Forward flow temperature sensor for channel #10 installed
H10-XW-R01	20	Return flow temperature sensor for channel #1 installed
H10-XW-R02	21	Return flow temperature sensor for channel #2 installed
H10-XW-R03	22	Return flow temperature sensor for channel #3 installed
H10-XW-R04	23	Return flow temperature sensor for channel #4 installed
H10-XW-R05	24	Return flow temperature sensor for channel #5 installed
H10-XW-R06	25	Return flow temperature sensor for channel #6 installed
H10-XW-R07	26	Return flow temperature sensor for channel #7 installed
H10-XW-R08	27	Return flow temperature sensor for channel #8 installed
H10-XW-R09	28	Return flow temperature sensor for channel #9 installed
H10-XW-R10	29	Return flow temperature sensor for channel #10 installed
H10-XW-A01	30	Class A local temperature sensor for channel #1 installed
H10-XW-A02	31	Class A local temperature sensor for channel #2 installed
H10-XW-A03	32	Class A local temperature sensor for channel #3 installed
H10-XW-A04	33	Class A local temperature sensor for channel #4 installed
H10-XW-A05	34	Class A local temperature sensor for channel #5 installed
H10-XW-A06	35	Class A local temperature sensor for channel #6 installed
H10-XW-A07	36	Class A local temperature sensor for channel #7 installed
H10-XW-A08	37	Class A local temperature sensor for channel #8 installed
H10-XW-A09	38	Class A local temperature sensor for channel #9 installed
H10-XW-A10	39	Class A local temperature sensor for channel #10 installed
H10-XW-B01	40	Class B local temperature sensor for channel #1 installed
H10-XW-B02	41	Class B local temperature sensor for channel #2 installed
H10-XW-B03	42	Class B local temperature sensor for channel #3 installed
H10-XW-B04	43	Class B local temperature sensor for channel #4 installed
H10-XW-B05	44	Class B local temperature sensor for channel #5 installed
H10-XW-B06	45	Class B local temperature sensor for channel #6 installed
H10-XW-B07	46	Class B local temperature sensor for channel #7 installed
H10-XW-B08	47	Class B local temperature sensor for channel #8 installed
H10-XW-B09	48	Class B local temperature sensor for channel #9 installed

Option	Bit #	Description
H10-XW-B10	49	Class B local temperature sensor for channel #10 installed

For example, if the H10-XW option is installed, and H10-XW-F as well as H10-XW-R01, H10-XW-R02, H10-XW-R03, H10-XW-R04 and H10-XW-R05 are connected, InstalledOptions would equal 0x001F0280. Notice, that there is no distinction between 24V and 230V versions of an option, if both exist. It is not permissible to connect two instances of the same extension module.

#### 6.4.6. Thermostat Cluster (Client)

The thermostat client cluster receives attribute reports for setpoints, local temperature, occupancy, system mode etc. from a leader <sup>[6]</sup> thermostat in a group of thermostats. This allows the leader to align one or more thermostats in the same space.

#### 6.4.7. Temperature Measurement Cluster (Client)

The temperature measurement client cluster receives attribute reports from temperature sensors and interprets the measured values as local temperature. Notice that this feature can be used in one of several ways: Classic zigbee temperature sensors can be directly bound such that they send their reports directly to the thermostat endpoint. Similarly, zigbee green power temperature sensors can be directly paired with the thermostat endpoint and used to the same effect. Finally, a gateway can collate temperature readings from different devices in the same space and provide a single temperature report to the thermostat with an average temperature. Notice this might interface with local temperature reports from a leader thermostat. Make sure you use either approach, otherwise the most recent report will currently take precedence.

#### 6.4.8. Relative Humidity Measurement Cluster (Client)

The relative humidity measurement client cluster receives attribute reports from relative humidity sensors. Reports are currently not evaluated; this feature is intended for future use.

#### 6.4.9. Occupancy Sensing Cluster (Client)

The occupancy sensing client cluster receives attribute reports from occupancy sensors and interprets the measured values as occupancy in the heating/cooling space. This will bring different heating and cooling setpoints into effect. Notice that this feature can be used in one of several ways: Classic zigbee occupancy sensors can be directly bound such that they send their reports directly to the thermostat endpoint. Similarly, zigbee green power occupancy sensors can be directly paired with the thermostat endpoint and used to the same effect. A gateway can collate occupancy reports from different devices in the same space and provide a single occupancy report to the thermostat. Notice this might interface with occupancy reports from a leader thermostat. Make sure you use either approach, otherwise the most recent report will currently take precedence.

### 6.5. Application Endpoint #2 - Thermostat

Use this application endpoint in (I) a two-pipe system to control the heating or cooling unit connected to output #2, or (II) in a four-pipe system to control the heating unit connected to output #2, and the cooling unit connected to output #9.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#).

## 6.6. Application Endpoint #3 - Thermostat

Use this application endpoint in (I) a two-pipe system to control the heating or cooling unit connected to output #3, or (II) in a four-pipe system to control the heating unit connected to output #3, and the cooling unit connected to output #8.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.7. Application Endpoint #4 – Thermostat

Use this application endpoint in (I) a two-pipe system to control the heating or cooling unit connected to output #4, or (II) in a four-pipe system to control the heating unit connected to output #4, and the cooling unit connected to output #7.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.8. Application Endpoint #5 – Thermostat

Use this application endpoint in (I) a two-pipe system to control the heating or cooling unit connected to output #5, or (II) in a four-pipe system to control the heating unit connected to output #5, and the cooling unit connected to output #6.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.9. Application Endpoint #6 – Thermostat

Use this application endpoint in a two-pipe system to control the heating or cooling unit connected to output #6. Notice that if thermostat endpoint #5 is configured for a four-pipe system it will overrule thermostat endpoint #6. This endpoint cannot be configured for a four-pipe system.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.10. Application Endpoint #7 – Thermostat

Use this application endpoint in a two-pipe system to control the heating or cooling unit connected to output #7. Notice that if thermostat endpoint #4 is configured for a four-pipe system it will overrule thermostat endpoint #7. This endpoint cannot be configured for a four-pipe system.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.11. Application Endpoint #8 – Thermostat

Use this application endpoint in a two-pipe system to control the heating or cooling unit connected to output #8. Notice that if thermostat endpoint #3 is configured for a four-pipe system it will overrule thermostat endpoint #8. This endpoint cannot be configured for a four-pipe system.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as



described in [Section 6.4](#) .

## 6.12. Application Endpoint #9 – Thermostat

Use this application endpoint in a two-pipe system to control the heating or cooling unit connected to output #9. Notice that if thermostat endpoint #2 is configured for a four-pipe system it will overrule thermostat endpoint #9. This endpoint cannot be configured for a four-pipe system.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.13. Application Endpoint #10 – Thermostat

Use this application endpoint in a two-pipe system to control the heating or cooling unit connected to output #10. Notice that if thermostat endpoint #1 is configured for a four-pipe system it will overrule thermostat endpoint #10. This endpoint cannot be configured for a four-pipe system.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #1, as described in [Section 6.4](#) .

## 6.14. Application Endpoint #11 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #1. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #1, or (II) in a four-pipe system this is the heating unit controlled by thermostat #1.

This is a Finding & Binding target endpoint.

The application endpoint exposes the following clusters:

Cluster	Direction	Description
0x0000	Inbound (Server)	<b>Basic</b> Provides basic information about the device, such as the manufacturer ID, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network.
0x0003	Inbound (Server)	<b>Identify</b> Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding.
0x0004	Inbound (Server)	<b>Groups</b> Allows adding this endpoint to one or more groups. Afterwards the endpoint can be addressed using the group address. This is also a prerequisite for scenes. You may also query group membership and delete group associations.
0x0005	Inbound (Server)	<b>Scenes</b> Allows storing one or more scenes per group, where each scene consists of a pre-set on/off state value. You may either store the current values as a scene, or provide scene settings when adding a scene, or delete scenes.
0x0006	Inbound (Server)	<b>On/off</b> Allows steering of the associated output, i.e. when used, disables the thermostat and uses the steering values provided by the level control cluster on this endpoint.
0x0006	Inbound (Server)	<b>Level Control</b> Allows steering of the associated output, i.e. when used, disables the thermostat and uses the steering values provided by the level control cluster on this endpoint.
0x0201	Outbound (Client)	<b>Thermostat</b> Receives PI heating demand or PI cooling demand steering reports from a thermostat.

### 6.14.1. Basic Cluster (Server)

Please refer to [Section 6.25.1](#) for details.

### 6.14.2. Identify Cluster (Server)

In identify mode, the channel LED will be toggled once every second, the output itself is not affected.

Attributes supported:

Attribute	Type	Description
0x0000	unsigned16	<b>IdentifyTime</b> The remaining time, in seconds, the device will be identifying itself.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Identify</b> Puts the device into identify mode, or terminates identify mode.
0x01	<b>Query Identify</b> Determines whether the device is currently identifying, and returns the remaining time if so.

### 6.14.3. Groups Cluster (Server)

Attributes supported:

Attribute	Type	Description
0x0000	bitmap8, read-only	<b>NameSupport</b> Always set to 0: This implementation does not support storing names for groups.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Add Group</b> Adds the endpoint to a group.
0x01	<b>View Group</b> Determines whether the device belongs to a group and returns the group name, if supported.
0x02	<b>Get Group Membership</b> Returns the set of groups this endpoint belongs to.
0x03	<b>Remove Group</b> Removes this endpoint from the specified group. Also removes all scenes that refer to this group.
0x04	<b>Remove All Groups</b> Removes this endpoint from all groups. Also removes all scenes that refer to any of the existing groups.
0x05	<b>Add Group if Identifying</b> Adds this endpoint to the group, if the endpoint is identifying.

### 6.14.4. Scenes Cluster (Client)

Attributes supported:

Attribute	Type	Description
0x0000	unsigned8, read-only	<b>SceneCount</b> Holds the total number of scenes (across all groups) currently stored on the device.
0x0001	unsigned8, read-only	<b>CurrentScene</b> If the SceneValid attribute is true, this attribute, together with the CurrentGroup attribute, indicates the currently active scene.
0x0002	unsigned8, read-only	<b>CurrentGroup</b> If the SceneValid attribute is true, this attribute, together with the CurrentScene attribute, indicates the currently active scene.
0x0003	bool, read-only	<b>SceneValid</b> If true, the scene identified by CurrentGroup and CurrentScene is currently active, i.e. all device attribute values match the values in the scene field set.
0x0004	bitmap8, read-only	<b>NameSupport</b> Always set to 1: This implementation does support storing names for scenes.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Add Scene</b> Adds a scene with or without a scene field set
0x01	<b>View Scene</b> Returns the scene field set, name and transition times for a scene.
0x02	<b>Remove Scene</b> Removes a scene from the scene table.
0x03	<b>Remove All Scenes</b> Removes all scenes that belong to a particular group.
0x04	<b>Store Scene</b> Stores the device's current state as a scene or updates a previously stored scene accordingly.
0x05	<b>Recall Scene</b> Reverts the device's current state using the values from the previously stored field set.
0x06	<b>Get Scene Membership</b> Returns the set of scenes (within the scope of the specified group) currently stored on the device.
0x40	<b>Enhanced Add Scene</b> Similar to Add Scene with higher transition time resolution (10ths of a second instead of once second)
0x41	<b>Enhanced View Scene</b> Similar to View Scene with higher transition time resolution (10ths of a second instead of once second)
0x42	<b>Copy Scene</b> Copies a scene in a single operation as an alternative to (Enhanced) View Scene, (Enhanced) Add Scene

#### 6.14.5. On/off Cluster (Client)



This cluster uses the binding table for managing reporting targets.

Attributes supported:

Attribute	Type	Description
0x0000	bool, read-only, reportable	<b>OnOff</b> Indicates the state of the valve output, either on = 'true' or off = 'false'. Notice this works in conjunction with the level control cluster on this endpoint.

Cluster commands supported:

Command	Description
0x00	<b>Turn off</b> Turns the output off. The attached load will be disconnected from the mains.

Command	Description
0x01	<b>Turn on</b> Turns the output on. The attached load will be connected to the mains.
0x02	<b>Toggle</b> Turns the output off, if it was turned on or turns the output on, if it was turned off.

### 6.14.6. Level Control Cluster (Server)



This cluster uses the binding table for managing reporting targets.

The level control cluster allows adjusting the PWM output duty-cycle between 0% and 100% of the cycle time, taking into account valve polarity. For instance, if the cycle time is one minute and the output steering level is set to 20%, and a valve that is normally closed is attached, the output will be powered for 12 seconds, then turned off for 48 seconds. In the same example, if the valve was normally open, the output would be powered for 48 seconds, then turned off for 12 seconds.

Attributes supported:

Attribute	Type	Description
0x0000	unsigned8, read-only, reportable	<b>CurrentLevel</b> Indicates the current level of the dimmer output, where 0 = 'off', 254 = '100%'.
0x0001	unsigned16, read-only	<b>RemainingTime</b> Specifies the amount of time, in units of 0.1 seconds, required to reach the specified target level
0x000F	bitmap8, persistent	<b>Options</b> <sup>[9]</sup> Specifies the default command options for commands that support options; these are applied either when the option value and mask are not included in the command payload, or the mask indicates that certain default option bits apply. Factory default setting for 'execute if off' = '1'.
0x0010	unsigned16, persistent	<b>OnOffTransitionTime</b> Specifies the amount of time, in units of 0.1 seconds, which will be used during a transition to either the on or off state, when an on/off/toggle command of the on/off cluster is used to turn the light on or off
0x0011	unsigned8, persistent	<b>OnLevel</b> Specifies the level that shall be applied, when an on/toggle command causes the light to turn on. When this attribute is set to the invalid value (0xFF), the light will return to the previously active level (before it was turned off), when it is turned on again
0x4000	unsigned8, persistent	<b>StartupLevel</b> <sup>[10]</sup> Specifies the initial level to be applied after reboot. When this attribute is set to the invalid value (0xFF), the light will return to the previously active level (before power was cut), when it is turned on again. Otherwise the current level will be set to the value specified here subject to range limitations imposed by the ballast configuration cluster on this endpoint
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Manufacturer-specific attributes supported:

Attribute	Type	Description
0x0001	bitmap8, persistent	<b>ValveType</b> The least significant bit of this attribute specifies the valve position when no power is supplied to the valve. The attached valve may be normally closed (bit #0 = '0') or normally open (bit #0 = '1').
0x0002	unsigned8, persistent	<b>CyclePeriod</b> Determines the duration of a single PWM cycle, in integer multiples of 15 seconds, minus one. For instance, a value of 0 equals 15 seconds, whereas a value of 119 indicates 30 minutes
0x0003	enum8, persistent, singleton	<b>Season</b> Specifies whether the device is in winter or summer mode. This will affect the steering level pre-sets while the device is in back-up mode, for instance. A value of '0' indicates winter mode and standard back-up values apply, whereas a value of '1' indicates summer mode and alternate back-up values apply. This singleton attribute is shared across all endpoints.

Attribute	Type	Description
0x0004	unsigned8, persistent	<b>BackupLevel</b> An output steering level which is applied to the valve output during winter time in case none of the other steering values is valid.
0x0005	unsigned8, persistent, singleton	<b>AlternateBackupLevel</b> An output steering level which is applied to the valve output during summer time in case none of the other steering values is valid.
0x0006	unsigned8, persistent	<b>LowerRange</b> A lower bound on the steering level. Output steering values below LowerRange will be clamped to LowerRange.
0x0007	unsigned8, persistent	<b>UpperRange</b> An upper bound on the steering level. Output steering values above UpperRange will be clamped to UpperRange.
0x0008	unsigned8, persistent, singleton	<b>PumpThresholdOn</b> A threshold, applicable to steering levels across all outputs, above which the appropriate circulation pump for either the heating or cooling pipe will be turned on. In other words, if any steering level exceeds the threshold, the pump will be turned on. The resulting control output is available on the H10-XP extension module.
0x0009	unsigned8, persistent, singleton	<b>PumpThresholdOff</b> A threshold, applicable to steering levels across all outputs, below which the appropriate circulation pump for either the heating or cooling pipe will be turned off. In other words, if all steering levels drop below the threshold, the pump will be turned off. The resulting control output is available on the H10-XP extension module.
0x000A	unsigned8, persistent, singleton	<b>HeatingDemandEnableThreshold</b> A threshold on the heating demand. When the heating demand of any channel exceeds this threshold, the device will indicate heating demand to a heating source (e.g. boiler or heat-pump) using the according output of the H10-XS extension module. This singleton attribute is shared across all endpoints.
0x000B	unsigned8, persistent, singleton	<b>HeatingDemandDisableThreshold</b> A threshold on the heating demand. When the heating demand of all channels drops below this threshold, the device will no longer indicate heating demand to a heating source (e.g. boiler or heat-pump) using the according output of the H10-XS extension module. This singleton attribute is shared across all endpoints.
0x000C	unsigned8, persistent, singleton	<b>CoolingDemandEnableThreshold</b> A threshold on the cooling demand. When the cooling demand of any channel exceeds this threshold, the device will indicate cooling demand to a cooling source (e.g. chiller or heat-pump) using the according output of the H10-XS extension module. This singleton attribute is shared across all endpoints.
0x000D	unsigned8, persistent, singleton	<b>CoolingDemandDisableThreshold</b> A threshold on the cooling demand. When the cooling demand of all channels drops below this threshold, the device will no longer indicate cooling demand to a cooling source (e.g. chiller or heat-pump) using the according output of the H10-XS extension module. This singleton attribute is shared across all endpoints.

#### Cluster commands supported:

Command	Description
0x00	<b>Move To Level</b> Moves the current level to a certain target level within a specified transition time. Supports command options <sup>[9]</sup> .
0x01	<b>Move</b> Moves the level either up or down at a specified rate. Supports command options <sup>[9]</sup> .
0x02	<b>Step</b> Increments or decrements the level by a certain amount within a specified transition time. Supports command options <sup>[9]</sup> .
0x03	<b>Stop</b> Stops any level change in progress due to a move, move to level, step or recall scene command.
0x04	<b>Move To Level with on/off</b> Moves the current level to a certain target level within a specified transition time. Maintains the companion on/off cluster's OnOff attribute.
0x05	<b>Move with on/off</b> Moves the level either up or down at a specified rate. Maintains the companion on/off cluster's OnOff attribute.

Command	Description
0x06	<b>Step with on/off</b> Increments or decrements the level by a certain amount within a specified transition time. Maintains the companion on/off cluster's OnOff attribute.
0x07	<b>Stop</b> Stops any level change in progress due to a move, move to level, step or recall scene command.

### 6.15. Application Endpoint #12 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #2. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #2, or (II) in a four-pipe system this is the heating unit controlled by thermostat #2.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

### 6.16. Application Endpoint #13 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #3. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #3, or (II) in a four-pipe system this is the heating unit controlled by thermostat #3.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

### 6.17. Application Endpoint #14 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #4. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #4, or (II) in a four-pipe system this is the heating unit controlled by thermostat #4.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

### 6.18. Application Endpoint #15 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #5. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #5, or (II) in a four-pipe system this is the heating unit controlled by thermostat #5.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

### 6.19. Application Endpoint #16 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #6. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #6, or (II) in a four-pipe system this is the cooling unit controlled by thermostat #5.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

## 6.20. Application Endpoint #17 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #7. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #7, or (II) in a four-pipe system this is the cooling unit controlled by thermostat #4.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

## 6.21. Application Endpoint #18 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #8. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #8, or (II) in a four-pipe system this is the cooling unit controlled by thermostat #3.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

## 6.22. Application Endpoint #19 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #9. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #9, or (II) in a four-pipe system this is the cooling unit controlled by thermostat #2.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

## 6.23. Application Endpoint #20 – Heating/Cooling Unit

Controls the heating or cooling unit connected to output #10. In (I) a two-pipe system this is the heating or cooling unit controlled by thermostat #10, or (II) in a four-pipe system this is the cooling unit controlled by thermostat #1.

Except for the associated output channels, this endpoint is otherwise similar to endpoint #11, as described in [Section 6.4](#).

## 6.24. Application Endpoint #21 – Master Control

Coordinates overall device operation and accepts temperature readings from an outdoor temperature sensor.

This is a Finding & Binding target endpoint.

The application endpoint exposes the following clusters:

Cluster	Direction	Description
0x0000	Inbound (Server)	<b>Basic</b> Provides basic information about the device, such as the manufacturer ID, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network.
0x0003	Inbound (Server)	<b>Identify</b> Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding.

Cluster	Direction	Description
0x0402	Outbound (Client)	<b>Temperature Measurement</b> Receives reports form temperature sensors.

### 6.24.1. Basic Cluster (Server)

Please refer to [Section 6.25.1](#) for details.

### 6.24.2. Identify Cluster (Server)

In identify mode, the status LED will be toggled once every second.

Attributes supported:

Attribute	Type	Description
0x0000	unsigned16	<b>IdentifyTime</b> The remaining time, in seconds, the device will be identifying itself.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Identify</b> Puts the device into identify mode, or terminates identify mode.
0x01	<b>Query Identify</b> Determines whether the device is currently identifying, and returns the remaining time if so.

### 6.24.3. Temperature Measurement Cluster (Client)

The temperature measurement cluster on this endpoint accepts reports from an outdoor temperature sensor and dispatches this information to all thermostat endpoints, i.e. endpoints #1...#10, which will update their OutdoorTemperature attribute accordingly and might use this information for improved regulation.☒

## 6.25. Application Endpoint #232 - Device Management

This endpoint provides device management functions using standard Zigbee foundation paradigms and the Zigbee Cluster Library [R4]. You may use the standard ZCL frames to enumerate, read and write attributes, configure attribute reporting, invoke commands, etc.

The application endpoint exposes the following clusters:

Cluster	Direction	Description
0x0000	Inbound (Server)	<b>Basic</b> Provides basic information about the device, such as the manufacturer ID, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network.
0x0003	Outbound (Client) <sup>[11]</sup>	<b>Identify</b> Allows to identify targets during Finding & Binding push-button commissioning.
0x0015	Inbound (Server)	<b>Commissioning</b> Offers fine-grained commissioning options, for example, configuring a certain short address, channel mask, concentrator setup etc. This cluster is rarely used.



Cluster	Direction	Description
0x0019	Outbound (Client)	<b>OTA Upgrade</b> Pull-oriented firmware upgrade. Searches the network for mating servers and allows the server to control all stages of the upgrade process, including which image to download, when to download, at what rate and when to install the downloaded image.

### 6.25.1. Basic Cluster (Server)

The basic cluster might be accessible via more than one endpoint. Most of its attributes are singleton attributes, i.e. all instances of the cluster share the same attribute value storage.

Attributes supported:

Attribute	Type	Description
0x0000	unsigned8, read-only	<b>ZCLVersion</b> The version of the cluster library
0x0001	unsigned8, read-only	<b>ApplicationVersion</b> The application version
0x0002	unsigned8, read-only	<b>StackVersion</b> The stack version
0x0003	unsigned8, read-only	<b>HWVersion</b> Specifies the hardware revision number
0x0004	string, read-only	<b>ManufacturerName</b> "ubisys"
0x0005	string, read-only	<b>ModelIdentifier</b> "H10"
0x0006	string, read-only	<b>DateCode</b> "YYYYMMDD-XX-FBV", where YYYY = year, MM = month, DD = day of production (hardware), XX = production facility, V = factory block version. For example, "20130411-DE-FB0" refers to a device built April 11, 2013 in Germany and programmed with factory block format version 0.
0x0007	enum8, read-only	<b>PowerSource</b> Always set to mains-powered, single phase (regardless of the connected supply voltage)
0x0010	string, persistent	<b>LocationDescription</b> Empty string by default. Might be set by commissioning software during installation
0x0011	enum8, persistent	<b>PhysicalEnvironment</b> "Unspecified" by default. Might be set by commissioning software during installation
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

None

### 6.25.2. Identify Cluster (Client)

The client side of the identify cluster on this endpoint is used for push-button commissioning.

Cluster commands transmitted:

Command	Description
0x01	<b>Query Identify</b> This command is broadcast during push-button commissioning to identify suitable binding targets. Once target devices are selected, they will be queried for matching descriptors.

### 6.25.3. Commissioning Cluster (Server)

This implementation supports all attributes and most commands. However, the cluster has limited value. It is rarely used and might be removed in the future.

### 6.25.4. Over-the-Air Upgrade Cluster (Client)

The image type for H10 is 0x7B3B. You may request the latest firmware in ZigBee OTA image format to upgrade devices in the field using your own OTA server and back-end. If you operate the H10 together with the ZigBee/Ethernet Gateway ubisys G1, the latest firmware for H10 will always be available automatically and you do not need to request it explicitly.

## 6.26. Application Endpoint #242 - Zigbee Green Power

This endpoint provides the Zigbee Green Power feature according to the 2015 edition of the specification, i.e. including support for Green Power Devices with IEEE EUI-64 and bidirectional commissioning. The Zigbee traffic between Proxies and Sinks utilizes standard Zigbee foundation paradigms and the Zigbee Cluster Library [R4]. You may use the standard ZCL frames to enumerate, read and write attributes, invoke commands, etc.

The application endpoint exposes the following clusters:

Cluster	Direction	Description
0x0021	Inbound (Server)	<b>Zigbee Green Power Sink</b> Allows pairing the H10 with Zigbee Green Power switches. The sink can directly receive Green Power frames but can also utilize proxies in the network.
0x0021	Outbound (Client)	<b>Zigbee Green Power Proxy</b> Allows sinks on the network to configure this device as a "Proxy", i.e. access point for Zigbee Green Power Devices into the Zigbee mesh network.

### 6.26.1. Green Power Cluster (Server)

The server-side of the Green Power cluster provides the Zigbee Green Power Sink functionality, i.e. provides the ability to process Green Power frames. The device can receive those frames directly if they are in radio range; or it can process frames tunnelled over the Zigbee network via other Green Power Proxies. This implementation supports unidirectional and bidirectional [12] GPDs.

Attributes supported:

Attribute	Type	Description
0x0000	unsigned8, read-only	<b>gpsMaxSinkTableEntries</b> The number of sink table entries supported by this device
0x0001	extended raw binary, read-only, persistent	<b>SinkTable</b> Entries in the sink table create a pairing between the sink and a Green Power Device. For an operational pairing, one or more entries in the translation table are also required
0x0002	bitmap8	<b>gpsCommunicationMode</b> Determines the default communication mode between sink and proxies for new entries added by the sink to the sink table when paired with a GPD
0x0003	bitmap8	<b>gpsCommissioningExitMode</b> Specifies under which circumstances the sink should leave commissioning mode
0x0004	unsigned16	<b>gpsCommissioningWindow</b> Time, in seconds, before a commissioning session expires
0x0005	bitmap8, persistent	<b>gpsSecurityLevel</b> Determines the minimum security level a GPD is required to support as well as whether or not the Trust Center needs to be involved into the commissioning process

Attribute	Type	Description
0x0006	bitmap24, read-only	<b>gpsFunctionality</b> Indicates Green Power features and building blocks supported by this device
0x0007	bitmap24, read-only	<b>gpsActiveFunctionality</b> Allows to disable certain Green Power features on this device
0x0020	bitmap8, persistent	<b>gpSharedSecurityKeyType</b> Determines the security key type to use for devices with bidirectional commissioning capabilities, i.e. out-of-the-box individual key, shared GP key, etc.
0x0021	key128, persistent	<b>gpSharedSecurityKey</b> The 128-bit AES-CCM* key that is being used to secure Green Power data frames
0x0022	key128, persistent	<b>gpLinkKey</b> The 128-bit AES-CCM* key that is being used to deliver keying material to Green Power devices
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>GP Notification</b> Tunnels GP frames from a Green Power Device to one or more sinks or groups of sinks
0x04	<b>GP Commissioning Notification</b> Tunnels GP frames from a Green Power Device to a sink in commissioning mode
0x05	<b>GP Sink Commissioning Mode</b> Makes the sink enter or leave commissioning mode for a particular endpoint or all endpoints
0x09	<b>GP Pairing Configuration</b> Creates, updates or removes pairings by modifying the sink and translation table entries accordingly
0x0A	<b>GP Sink Table Request</b> Allows to query the sink table for a certain Green Power Device or read out the table in chunks

Cluster commands transmitted:

Command	Description
0x01	<b>GP Pairing</b> Creates, updates or removes proxy table entries
0x02	<b>GP Proxy Commissioning Mode</b> Makes the proxy enter commissioning mode for a particular sink, or leave commissioning mode
0x06	<b>GP Response</b> Tunnels GP data frames from a sink to a bidirectional Green Power Device
0x0A	<b>GP Sink Table Response</b> Conveys a set of sink table entries to a management application

## 6.26.1.1. Green Power Device Interaction

### 6.26.1.1.1. Commissioning a Green Power Sensor

The H10 can be put in commissioning mode either by using a suitable commissioning tool (for example the ubisys Network Manager) or by instigating Finding & Binding as a target. Once in commissioning mode, the device is ready to pair with Green Power sensors. Notice that endpoints #1 to #10 are commissioned individually, i.e. they can receive data from separate sensors.

### 6.26.1.1.2. Supported Device Types

The H10 can be paired with the following pre-defined Green Power Device types:

Device Type	Description
0x12	<b>Occupancy Sensor</b> Attribute reports for occupancy sensing cluster
0x30	<b>Temperature Sensor</b> Attribute reports for temperature measurement cluster
0x33	<b>Indoor Environmental Sensor</b> Attribute reports for temperature measurement and relative humidity measurement cluster

In addition, any Green Power device that sends at least one of the supported commands listed in the next section conveying attribute reports for either temperature measurement, relative humidity measurement, or occupancy sensing, is acceptable.

### 6.26.1.1.3. Supported Green Power Commands

The following Green Power commands are supported by default translations:

Command	Description
0xA0 <sup>[13]</sup>	Attribute Report
0xA1 <sup>[13]</sup>	Attribute Report, manufacturer-specific
0xA2 <sup>[13]</sup>	Attribute Report, multiple clusters
0xA3 <sup>[13]</sup>	Attribute Report, multiple clusters, manufacturer-specific
0xA6	Generic Tunnel
0xA8	Attribute Report, compact

### 6.26.2. Green Power Cluster (Client)

The client-side of the Green Power cluster provides the Zigbee Green Power Proxy functionality, i.e. makes the device act as an “access point” for Green Power Devices (GPDs). This implementation supports unidirectional and bidirectional <sup>[14]</sup>GPDs.

Attributes supported:

Attribute	Type	Description
0x0010	unsigned8, read-only	<b>gppMaxProxyTableEntries</b> The number of proxy table entries supported by this device
0x0011	extended raw binary, read-only, persistent	<b>ProxyTable</b> Entries in the proxy table create a link between Green Power Devices and Green Power Sinks
0x0016	bitmap24, read-only	<b>gppFunctionality</b> Indicates Green Power features and building blocks supported by this device
0x0017	bitmap24, read-only	<b>gppActiveFunctionality</b> Allows to disable certain Green Power features on this device
0x0020	bitmap8, persistent	<b>gpSharedSecurityKeyType</b> Determines the security key type to use for devices with bidirectional commissioning capabilities, i.e. out-of-the-box individual key, shared GP key, etc.
0x0021	key128, persistent	<b>gpSharedSecurityKey</b> The 128-bit AES-CCM* key that is being used to secure Green Power data frames
0x0022	key128, persistent	<b>gpLinkKey</b> The 128-bit AES-CCM* key that is being used to deliver keying material to Green Power devices
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification this implementation adheres to.

Cluster commands supported:

Command	Description
0x01	<b>GP Pairing</b> Creates, updates or removes proxy table entries
0x02	<b>GP Proxy Commissioning Mode</b> Makes the proxy enter commissioning mode for a particular sink, or leave commissioning mode
0x06	<b>GP Response</b> Tunnels GP data frames from a sink to a bidirectional Green Power Device
0x0B	<b>GP Proxy Table Request</b> Allows to query the proxy table for a certain Green Power Device or read out the table in chunks

Cluster commands transmitted:

Command	Description
0x00	<b>GP Notification</b> Tunnels GP frames from a Green Power Device to one or more sinks or groups of sinks
0x04	<b>GP Commissioning Notification</b> Tunnels GP frames from a Green Power Device to a sink in commissioning mode
0x0B	<b>GP Proxy Table Response</b> Conveys a set of proxy table entries to a sink or management application

[4] Available in ZigBee stack version 1.60 and above. Legacy ZCP requirement – do not use in applications

[5] Available in Zigbee stack version 1.56 and above.

[6] Available in Zigbee stack version 1.50 and above.

[7] Available in Zigbee stack version 1.61 and above.

[8] An example of a suitable leader is the Legrand in-wall thermostat UI Model No. 0 663 40

[9] Since application version 1.07. Prior, the hard-coded default for the “execute if off” option was always ‘1’.

[10] Since application version 1.07. Prior, the device would always have the output turned off after reboot.

[11] Since application version 1.04. Prior, the server side of identify was available.

[12] Bidirectional communication is currently limited to the commissioning stage

[13] Notice that a single translation table entry, 0xAF, is used to cover 0xA0, 0x0xA1, 0xA2 and 0xA3 commands

[14] Bidirectional communication is currently limited to the commissioning stage

## 7. H10-B Base Module

The H10-B base module is available in 230V~ and 24V= flavors, H10/230-B and H10/24-B, respectively. The base module contains the power supply, the zigbee interface, 10 control outputs for direct valve drive, a local man/machine interface, and two bus connectors for plug & play extension modules. Extension modules can be connected simply by attaching them to left-hand or right-hand side of the base module. Each module provides a connector for adding further modules in a cascade in order to adapt H10 to the specific of your heating and cooling system.

### 7.1. Theory of Operation

The H10 provides up to 10 individual thermostats with 0...100% level control steering. Each of the ten thermostats can be operated in heating mode or cooling mode in a two-pipe system. The first five thermostats also support auto mode in a four-pipe system – a mode, where the thermostat automatically transitions between heating and cooling without user intervention. Thermostats require knowledge of local temperature to operate. The local temperature can be provided in one of many ways:

- Classic zigbee temperature sensors configured for direct device-to-device communications
- Green Power temperature or indoor-environmental sensors configured for direct device-to-device communications; H10 acts as a Green Power sink and is able to process such sensor readings directly
- Other zigbee thermostats configured for direct device-to-device communications and reporting their local temperature and set-points to H10
- A hub receiving temperature readings from multiple sources, e.g. motion detectors or key pads with temperature sensors, and performing sensor fusion to provide a single, collated local temperature to the thermostat
- 1-Wire® temperature sensors directly measuring local temperature <sup>[15]</sup>
- 1-Wire® temperature sensors measuring forward and return flow temperatures <sup>[16]</sup>

In a two-pipe system, each of the thermostats (endpoints #1...#10) control a corresponding heating or cooling unit (endpoints #11...#20); in a four-pipe system a single thermostat controls both a heating and cooling unit.

A heating or cooling unit maintains a set of steering levels and applies the level with the highest priority, which is valid. A level is deemed valid if it's value is in the range 0...254. The following steering levels are available, in descending priority:

- **Emergency Steering Value.** This is enabled or disabled by local user interaction
- **Level Steering Value.** The value is set via remote level control steering
- **Thermostat 2 Steering Value.** Cooling demand from thermostat (only in four-pipe systems)
- **Thermostat 1 Steering Value.** Heating or cooling demand from thermostat
- **External Pass-Through Steering Value.** On/off signal provided by wired thermostat
- **Backup Steering Value.** Standard or alternate setting based on season of the year

For instance, if the system is not in emergency mode, no level override has been sent from the zigbee network, the thermostat is attached to the heating unit in a two-pipe system and has valid temperature sensor readings, the PI heating demand value steers the output PWM cycle. In the same example, if

the temperature sensor fails (e.g. runs out of batteries) and no external wired thermostat is connected, the heating/cooling zone enters backup mode, where the steering value is pre-determined, i.e. the standard backup steering value will be used during the winter season and the alternate backup steering value will be used during the summer season.

Similarly, if H10 is used in a first stage of retrofitting, temperature sensors might not be available, but existing wired thermostats might be connected using H10-XI. In this case, the existing thermostat signal is mirrored to the valve control output. Whenever the user wants he can override the pass-through signal with a steering level using the zigbee level control interface. This manual steering can be disabled by changing the thermostat mode and without a temperature sensor, the pass-through is in effect again.

### 7.1.1. Proportional/Integral (PI) Control Loop

The H10 employs ten independent proportional/integral control loops to provide heating and cooling demand steering levels to associated heating and cooling units. A dead-band between heating and cooling set-points prevents heating and cooling units from working against each other within the comfort zone. The controller is protected against integral wind-up.

## 7.2. Using H10 for Electrical Underfloor Heating

It is possible to use H10/24-B and H10/230-B to control electrical underfloor heating. However, it requires the addition of solid-state relays (SSRs) for each circuit because PWM outputs are only rated for direct drive for low-power valves, not for driving high-power loads directly. Typical heating foils easily require 10A or 16A per circuit. It is important to use solid-state relays because electromechanical relays are not well suited for PWM applications due to on/off cycle lifetime limitations. To select a suitable SSR make sure the control voltage of 24V= for H10/24-B and 230V~ for H10/230-B is supported and the SSR is rated for the voltage and current draw of the heating foils.

Examples of suitable SSRs:

- crydom DR22 family of solid-state relays, 48V...600V loads, 20A, 30A or 35A load
  - DR2260Dxxx series with control voltages of 4...32V= suitable for H10/24-B
  - DR2260Axxx series with control voltages of 90...280V suitable for H10/230-B
- Schneider Electric Series 861
  - 861SSRA208-DC-1 suitable for H10/24-B, 8A drive capability
  - 861SSRA210-DC-1 suitable for H10/24-B, 10A drive capability
- Omron Series G3PJ
  - G3PJ-215B-PU suitable for H10/24-B, 15A drive capability
  - G3PJ-225B-PU suitable for H10/24-B, 25A drive capability

Notice that in these constellations H10/24-B can be used to drive 230V~ heating elements. You can also combine H10/24-B with H10/230-XI to connect existing 230V~ wired thermostats, i.e. there is no restriction to combine H10/230-XI with H10/230-B, only.

[15] Requires H10-XW extension

[16] Requires H10-XW extension; local temperature is estimated based on flow temperatures

## 8. H10-XP Circulation Pump Extension Module

The H10-XP extension module allows integration of circulation pumps for two-pipe or four-pipe systems. In a two-pipe system, one output of H10-XP is used to control one circulation pump for the forward flow used to either heat or cool. In a four-pipe system, two separate outputs are available on H10-XP, one for the heating flow and one for the cooling flow. The output is enabled when any of the output levels exceeds the circulation pump turn-on threshold. It is disabled when all of the output levels drop below the circulation pump turn-off threshold.

The H10-XP is capable of directly driving two 230V~/50Hz pumps with 500VA each using its TRIAC outputs.



## 9. H10-XS Heating and Cooling Demand Extension Module

The H10-XS extension module allows to call for heat or cold and optionally notify a heat or cold source that heating and cooling demands are satisfied.

H10-XS features two single-pole double-through (SPDT) relays: The top one outputs a binary cooling demand signal, the bottom one outputs a binary heating demand signal. One SPDT is capable of directly driving a 230V~ heat source (e.g. boiler) with 1.800VA and the other one a cold source (e.g. chiller) with 1.800VA. Alternatively, each SPDT relay can be used as dry contact for a heat-pump or similar complex devices to call for heat, call for cold and optionally notify of “heating demand satisfied” and “cooling demand satisfied” conditions. H10 calls for heat when any of the heating unit levels exceeds the heating demand enable threshold; conversely, when all of the heating unit levels drop below the heating demand disable threshold, heating demand is satisfied.

H10 calls for cold when any of the cooling unit levels exceeds the cooling demand enable threshold; conversely, when all of the cooling unit levels drop below the cooling demand disable threshold, cooling demand is satisfied.

## 10. H10-XI Legacy Wired Thermostats Interface Extension Module

The H10-XI extension module allows to connect existing wired thermostats to H10 in retrofit deployments, where users may not want to install new thermostats or temperature sensors throughout the premise to keep upfront costs low for simple deployments. In such scenarios, H10-XI can connect existing thermostats, which will continue to operate as before, with the added value of being able to overrule thermostats in certain cases, e.g. turn off heating while abroad and turn it on again shortly before returning home. It is important for users to understand that this is likely a transitional step, and, at some point, temperature sensors or in-wall thermostat UIs should be installed to achieve the state-of-the-art regulation performance.

Two flavors of H10-XI are available, one for 24V= (H10/24-XI) and one for 230V~ (H10/230-XI). When a logic high level is present on any of the channel inputs, the resulting output channel will be steered to 100% (on), otherwise it will be steered to 0% (off). Prioritization ensures that when users install temperature sensors or thermostat UIs later on, advanced temperature regulation using H10's thermostat endpoints will take precedence over these external thermostat signals. Conversely, if these sensors fail (no battery, no network connectivity etc.), external thermostats are in charge of regulation again.

Besides the 10 input channels, which directly map to the 10 output channels for heating and cooling units, H10-XI also provides to auxiliary inputs marked "A" and "B".

## 11. H10-XW 1-Wire® Bridge Extension Module

The 1-Wire® bridge extension module H10-XW can be used to connect up to 64 1-Wire® devices to the H10. Notice that contrary to what the marketing name indicates, each sensor actually connects using two (ground, supply/data) or three wires (ground, supply, data). H10-XW provides four identical 1-Wire® ports, each with ground (0V), supply (+5V) and data. Any of these ports can be used to connect sensors. Multiple sensors can be connected to one port. The total number of sensors must not exceed 64.

### 11.1. Supported Sensors

Temperature sensors must be connected with all three wires to ensure stable operation and fast conversion times. All sensors offered by ubisys come pre-programmed and labelled for the entity they are supposed to measure, such that no user configuration is required at all when installing the device in the premise, making for an out-of-the-box plug & play sensor experience without guessing which sensor is which. Therefore, do not use other sensors.

Label	Identifier	Description
H10-XW-O	0xA050	Outdoor temperature
H10-XW-F	0xA450	Forward flow temperature, applicable to all channels
H10-XW-F01	0xA451	Forward flow temperature for channel #1
H10-XW-F02	0xA452	Forward flow temperature for channel #2
H10-XW-F03	0xA453	Forward flow temperature for channel #3
H10-XW-F04	0xA454	Forward flow temperature for channel #4
H10-XW-F05	0xA455	Forward flow temperature for channel #5
H10-XW-F06	0xA456	Forward flow temperature for channel #6
H10-XW-F07	0xA457	Forward flow temperature for channel #7
H10-XW-F08	0xA458	Forward flow temperature for channel #8
H10-XW-F09	0xA459	Forward flow temperature for channel #9
H10-XW-F10	0xA45A	Forward flow temperature for channel #10
H10-XW-R01	0xA551	Return flow temperature for channel #1
H10-XW-R02	0xA552	Return flow temperature for channel #2
H10-XW-R03	0xA553	Return flow temperature for channel #3
H10-XW-R04	0xA554	Return flow temperature for channel #4
H10-XW-R05	0xA555	Return flow temperature for channel #5
H10-XW-R06	0xA556	Return flow temperature for channel #6
H10-XW-R07	0xA557	Return flow temperature for channel #7
H10-XW-R08	0xA558	Return flow temperature for channel #8
H10-XW-R09	0xA559	Return flow temperature for channel #9
H10-XW-R10	0xA55A	Return flow temperature for channel #10
H10-XW-A01	0xA251	Class A local temperature for channel #1
H10-XW-A02	0xA252	Class A local temperature for channel #2
H10-XW-A03	0xA253	Class A local temperature for channel #3
H10-XW-A04	0xA254	Class A local temperature for channel #4
H10-XW-A05	0xA255	Class A local temperature for channel #5
H10-XW-A06	0xA256	Class A local temperature for channel #6
H10-XW-A07	0xA257	Class A local temperature for channel #7
H10-XW-A08	0xA258	Class A local temperature for channel #8
H10-XW-A09	0xA259	Class A local temperature for channel #9
H10-XW-A10	0xA25A	Class A local temperature for channel #10
H10-XW-B01	0xA351	Class B local temperature for channel #1
H10-XW-B02	0xA352	Class B local temperature for channel #2

Label	Identifier	Description
H10-XW-B03	0xA353	Class B local temperature for channel #3
H10-XW-B04	0xA354	Class B local temperature for channel #4
H10-XW-B05	0xA355	Class B local temperature for channel #5
H10-XW-B06	0xA356	Class B local temperature for channel #6
H10-XW-B07	0xA357	Class B local temperature for channel #7
H10-XW-B08	0xA358	Class B local temperature for channel #8
H10-XW-B09	0xA359	Class B local temperature for channel #9
H10-XW-B10	0xA35A	Class B local temperature for channel #10

H10 includes device drivers for 1-Wire® temperature sensors of the following types: ▪ DS1822 ▪ DS18B20 ▪ DS18S20 Identifiers are always encoded in the form 0xA<sub>m</sub>5<sub>n</sub>, where m is a type (0 = outdoor temperature, 2 = class A local temperature, 3 = class B local temperature, 4 = forward flow temperature, 5 = return flow temperature) and n is the scope (0 = global, 1...10 = channel #1...channel #10). Other values are reserved for future use. The identifier is programmed into the sensor's EEPROM at the factory. The information in this chapter is only for reference. There is no need for users to understand or program these values.

Forward flow temperatures are measured where e.g. hot water from a boiler or cool water from a chiller is fed into individual valves of a heating circuit manifold. Return flow temperatures are measured where water from radiating elements returns. In cases where forward temperature is always expected to be the same for all circuits (e.g. when underfloor heating manifold)

Class A temperatures are measured approximately 1.5m above ground, preferably in the area where persons are typically present and which should serve as the reference point for the regulation algorithm. Class B temperatures are measured close to the heating or cooling unit, e.g. in-floor between wounds for an underfloor heating system, or close to the knob for a radiator valve.

## 12. Physical Dimension

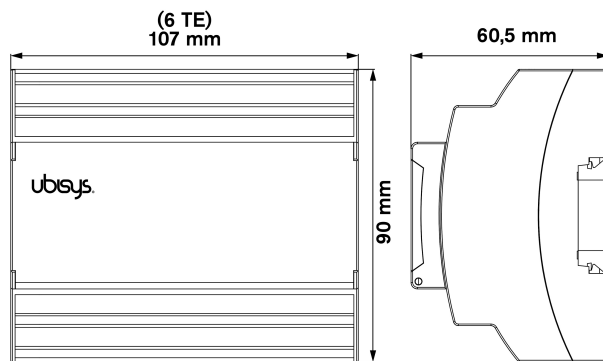


Figure 1. H10/230V-24V-B 10-Channel Zigbee Thermostat for 230V~/50Hz-24V Base Modules – Dimensions

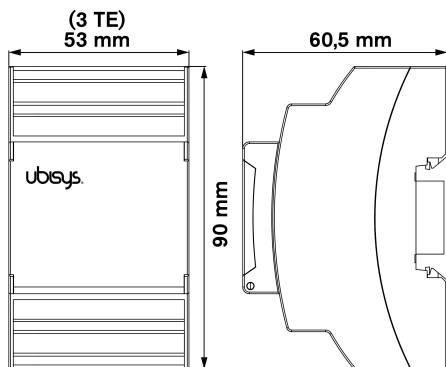
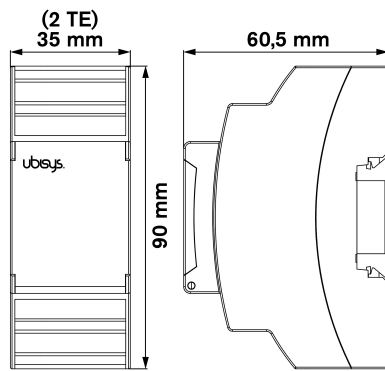


Figure 2. H10/230-XI Wired 230V~/50Hz, H10/24-XI Wired 24V Thermostat Interface Extension Modules – Dimensions



*Figure 3. H10-XP Circulation Pump Control, H10-XS Heating/Cooling Demand, H10-XW 1-Wire® Bridge Extension Modules – Dimensions*

## 13. Ordering Information

The following tables list the product variants available. Use the specified order code for your orders.

Vendor/Brand	Order Code	Description
ubisys	1205	H10/230-B 10-Channel Zigbee Thermostat for 230V~/50Hz Base Module
ubisys	1199	H10/24-B 10-Channel Zigbee Thermostat for 24V= Base Module
ubisys	1250	H10-XP Circulation Pump Control Extension Module
ubisys	1243	H10-XS Heating/Cooling Demand Extension Module
ubisys	1236	H10-XW 1-Wire® Bridge Extension Module
ubisys	1229	H10/230-XI Wired 230V~/50Hz Thermostat Interface Extension Module
ubisys	1212	H10/24-XI Wired 24V= Thermostat Interface Extension Module

The housings are flame retardant (V-0), Color: Black, RAL 9005.☒

## 14. General Terms & Conditions Of Business

When placing your order you agree to be bound by our General Terms & Conditions of Business, "Allgemeine Geschäftsbedingungen", which are available for download here:

[www.ubisys.de/en/smarthome/terms.html](http://www.ubisys.de/en/smarthome/terms.html)



## 15. Declaration Of Conformity



We – ubisys technologies GmbH, Neumannstraße 10, 40235 Düsseldorf, Germany – declare under our sole responsibility that the ubisys Heating Regulator H10 with order codes as detailed in [Chapter 13](#) under the trade name “ubisys” to which this declaration relates are in conformity with the following directives and standards:

Directive/Standard	Description/Scope
2014/53/EU	Radio Equipment Directive (RED)
2014/30/EU	Electromagnetic Compatibility Directive (EMC)
2014/35/EU	Low Voltage Directive (LVD)
2012/19/EU	Waste Electrical and Electronic Equipment Directive (WEEE)
2011/65/EU	Restriction of Hazardous Substances Directive (RoHS)
EN 300 328 V2.2.2	ERM; Wideband transmission systems; 2.4 GHz ISM band
EN 300 440 V2.2.1	ERM; Radio equipment to be used in the 1 GHz to 40 GHz frequency range
EN 301 489-1 V2.1.1	EMC
IEEE 802.15.4:2020	IEEE Standard 802 – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)
ZigBee 3.0	ZigBee 2017 with Green Power

Düsseldorf, Germany

September 16, 2021

Place of issue

Date of issue

Dr.-Ing. Arasch Honarbacht

Managing Director, Head of Research & Development

Full name of Authorized Signatory

Title of Authorized Signatory

**ubisys**  
 ubisys technologies GmbH  
 Neumannstr. 10  
 40235 Düsseldorf  
 Germany

Signature

Seal

## 16. Revision History

<b>Revision</b>	<b>Date</b>	<b>Remarks</b>
1.0	11/05/2018	Initial Public Version
1.1	09/06/2021	Update Declaration of Conformity
1.2	04/12/2023	Update supported commands of Thermostat Cluster Server and other typo fix

## 17. Contact

ubisys technologies GmbH  
Neumannstr. 10  
40235 Düsseldorf  
Germany

T: +49. 211. 54 21 55 - 19  
F: +49. 211. 54 21 55 - 99

[www.ubisys.de](http://www.ubisys.de)  
[info@ubisys.de](mailto:info@ubisys.de)