

# ZIGBEE HEATING REGULATOR H1

## REFERENCE MANUAL



# Table of Contents

1. Overview	1
2. Features	2
3. Installation	3
4. Initial Device Start-Up	4
5. Man-Machine interface (MMI)	5
5.1. Appearance of H1	5
5.2. Display of H1	5
5.3. Commissioning and operations on H1	5
6. Zigbee Interface	7
6.1. Device Anatomy	7
6.2. Installation Code	7
6.3. Application Endpoint #0 - Zigbee Device Object	8
6.4. Application Endpoint #1 - Thermostat	9
6.4.1. Basic Cluster (Server)	10
6.4.2. Power configuration Cluster (Server)	11
6.4.3. Identify Cluster (Server)	11
6.4.4. Groups Cluster (Server)	11
6.4.5. Scenes Cluster (Server)	12
6.4.6. Time Cluster (Server)	13
6.4.7. Poll Control Cluster (Server)	13
6.4.8. Thermostat Cluster (Server)	14
6.4.9. Works With All Hubs Cluster (Server)	15
6.4.10. Identify Cluster (Client)	18
6.4.11. Time Cluster (Client)	18
6.4.12. Over-the-Air Upgrade Cluster (Client)	19
6.4.13. Thermostat Cluster (Client)	19
6.4.14. Temperature Measurement (Client)	19
6.4.15. Relative Humidity Measurement (Client)	19
6.4.16. Occupancy Sensing (Client)	19
7. Thermostat Heating Setpoints Schedules	21
7.1. Weekdays configuration	21
7.2. Format of setpoint schedules	21
7.3. Number of daily heating setpoint schedules	22
7.4. Schedule Configuration Examples	23
8. Physical Dimension	27
9. Ordering Information	28
10. General Terms & Conditions Of Business	29
11. Declaration Of Conformity	30
12. Revision History	31
13. Contact	32

## 1. Overview

Thank you for purchasing the ubisys Heating Control H1. The H1 heating control is a smart, battery-powered radiator thermostat with integrated temperature sensor for radiator heating systems and is used for efficient room temperature control in private and commercial facilities.

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 3.0 radiator controller
- Advanced PI control loop to determine optimum heating demand
- Supports scheduling heating set-points on a weekly basis
- Supports vacation mode with configurable heating set-point schedule
- Works with remote temperature, occupancy or combined environmental sensors
- Local control even works when the device is not joined to a network (not commissioned yet)
- Man-Machine-Interface (MMI): Display, push-button and rotary dial for calibration, network commissioning, temperature set-point setup, weekly schedule configuration etc.
- Made in Germany using high-quality, enduring parts for many years of life expectancy
- Supports groups, scenes, bindings, and reporting
- On-board MCU: Advanced 32-bit Qorvo QPG6095 micro-controller running at 32MHz with 64KB SRAM, featuring IEEE 802.15.4 compliant PHY and real-time MAC
- On-board inverted-F antenna
- Leverages ubisys 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 sleepy end device
- Three pre-configured Trust Center Link-Keys for joining:
  - Global Default Trust Center Link-Key (“ZigBeeAlliance09”)
  - ZigBee 3.0 Global Distributed Security Link-Key
  - Device-individual link-key derived from installation code - also printed as QR barcode
- 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 packet transmission with automatic network-level retries
- Advanced initial joining algorithm featuring a fast retry mode and options for back-off
- Advanced rejoining algorithm featuring a fast retry mode and options for back-off
- Enhanced retry mechanism to increase the reliability of data polling
- Feature support of the Amazon Works With All Hubs (WWAH) cluster
- Firmware upgradable over-the-air during normal operation using ZigBee OTA Upgrade Cluster

### 3. Installation

Please refer to the installation guide of the product manual, which is included in the product package. It provides detailed installation instructions.

## 4. Initial Device Start-Up

In a factory-new state, the H1 looks for open ZigBee networks to join when being first-time supplied with power.

Open your ZigBee network for joining and then power-up the H1.

After the initialization phase, the H1 starts the calibration process, during which movements of the H1 valve pin take place to adjust to the radiator valve. Following a successful calibration, H1 performs the network startup, discovers existing networks and attempts to commission itself to an open Zigbee network. After initial network commissioning to an open network succeeds, the quality of the Zigbee wireless connection is indicated in the upper left corner of H1's display. the H1 will then prolong the joining window by three minutes by broadcasting a ZDO permit joining request command.

After reboot, given the H1 has been properly commissioned before, it attempts resuming network activity by performing network rejoin to find a suitable router parent. Once completed, H1 will broadcast a device announcement command to announce itself to the entire network. In such a rejoin case, H1 will NOT broadcast a ZDO permit joining request command to start a new joining window.

## 5. Man-Machine interface (MMI)

The application software installed on the H1 provides a user-friendly MMI.

### 5.1. Appearance of H1

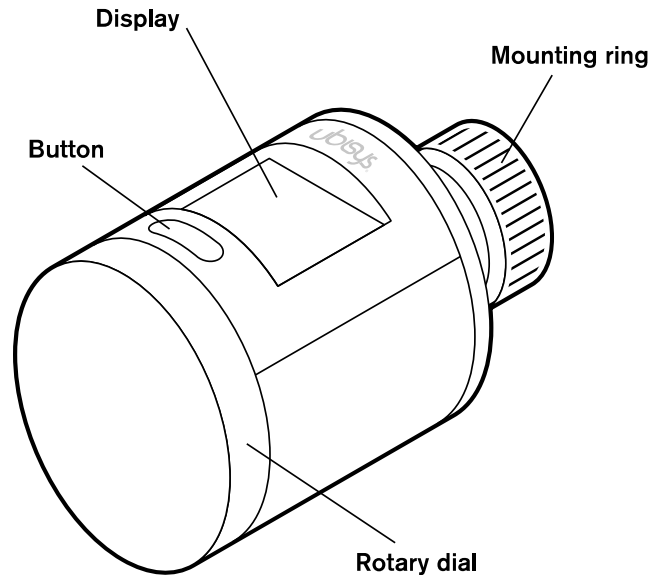


Figure 1. Heating Regulator H1 - Appearance of the device

### 5.2. Display of H1

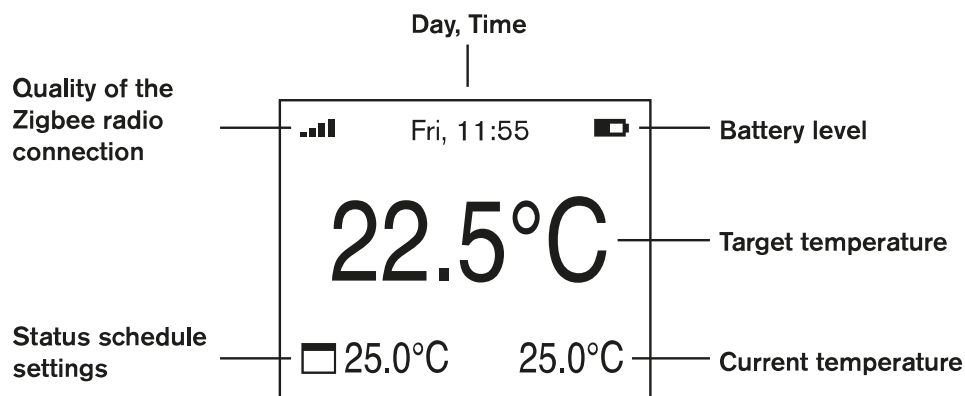


Figure 2. Heating Regulator H1 - Display of the device

### 5.3. Commissioning and operations on H1

Utilizing button press and rotary dial as shown in [Figure 1](#), user can swiftly and effortlessly complete commissioning and operational tasks on the H1. The display succinctly showcases the progress and outcomes of a commissioning process, along with the temperature set-point and current operational status related to heating regulation, see e.g. [Figure 2](#)

Table 1. Supported commissioning and operational tasks on H1 via MMI

Item #	Commissioning and Operational Tasks
1	<b>Operation: Calibration</b> Trigger a calibration process to automatically match the H1's tappet to the radiator valve

Item #	Commissioning and Operational Tasks
2	<p><b>Zigbee Commissioning: Factory Reset</b> Resets all settings to factory defaults and restarts the device</p>
3	<p><b>Zigbee Commissioning: Network Steering</b> Option #1: Attempt to join an open network, if H1 is in a factory-new state Option #2: Broadcast a ZDO Permit Joining command to open a network for 180 seconds, if H1 is operational on this network</p>
4	<p><b>Zigbee Commissioning: Finding &amp; Binding</b> Option #1: As F&amp;B Initiator, query identifying targets and create suitable bindings to the discovered targets Option #2: As F&amp;B Target, enter the identify mode for 180 seconds</p>
5	<p><b>Operation: Unmount</b> Retract the H1's ram completely for facilitating a convenient disassembly</p>
6	<p><b>Operation: Set Date &amp; Time</b> Manually set the current date and time for enabling configuration of weekly schedules on the H1. Should a device in the same network operate as a timer server (e.g. the ubisys Gateway G1), furnishing time information, H1 can autonomously request this information, obtaining wirelessly and syncing its local time accordingly.</p>
7	<p><b>Operation: Set Temperature Offset</b> In specific situations, the temperature measured locally on H1 may differ from the overall temperature at the location. To address this, user can manually set an optional temperature offset within the range of [-10°C, +10°C]. This offset compensates for any discrepancies, meaning that the temperature used for regulation is calculated as the locally measured temperature minus the temperature offset. Note that if a remotely measured temperature value is obtained through over-the-air reporting, it will be directly used for heating regulation without taking the temperature offset into account.</p>
8	<p><b>Operation: Configure Weekly Schedule</b> Create automated heating regulation based on pre-configured weekly set-point schedules.</p> <p>Note that, weekly schedules can be added, modified or removed not only through the MMI on the H1, but also upon receiving over-the-air ZCL commands from a management system.</p> <p>If setting or adding weekly schedules through the MMI, keep in mind that the internally applied schedule transition pattern may differ from the one defined in an over-the-air ZCL <a href="#">Set Weekly Schedule</a> command. Upon reception of such a command, the existing MMI-added weekly schedules will be entirely erased before storing the newly received schedules using the standard ZCL pattern. Furthermore, the MMI for weekly schedule configuration will stay disabled until the schedules using the standard ZCL pattern are later removed totally.</p> <p>Please find the details at <a href="#">Chapter 7</a></p>
9	<p><b>Operation: Switch Off</b> Switch off H1 and stop heating regulation.</p>
10	<p><b>Operation: Enable Vacation Mode</b> Apply heating schedules for the vocation mode, if any, or the pre-defined temperature set-point for absence.</p>

For carrying out each specific commissioning and operational tasks listed above [Table 1](#) on H1, please refer to the [Heating Control H1 - Manual](#), which provides detailed instructions respectively.



## 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 22, Document No. 05-3474-22
- [R3] ZigBee 2015 Layer PICS and Stack Profiles, Revision 6, Document No. 08-0006-06
- [R4] ZigBee Cluster Library Specification, Revision 8, Document No. 07-5123-08
- [R5] ZigBee Base Device Behavior Specification, Revision 12, Document No. 16-02828-12
- [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
- [R9] ZigBee WWAH ZCL Cluster Definition, Version 1., Document No. 17-01067-024

### 6.1. Device Anatomy

The H1 heating control, a battery-powered smart radiator thermostat with a built-in temperature sensor designed for radiator heating systems, ensures precise and effective temperature regulation in both residential and commercial spaces.

Using Zigbee, a globally recognized radio standard, the H1 communicates wirelessly. Temperature control is fully automated, with the ability to set the temperature either through predefined weekly schedules or manually on the H1 itself. Remote adjustments can also be made via a smartphone application, such as the ubisys Smart Home App.

To operate through the ubisys Smart Home App, the ubisys Gateway G1 is necessary. If a Zigbee-capable gateway from another manufacturer is employed, an appropriate app can be utilized. The power source of the H1 consists of two lithium batteries (type AA with 1.5 V, Energizer L91, Ultimate Lithium, or an equivalent).

H1 provides 2 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 enables temperature regulation and valve output control, aligning with either individual temperature set-points or weekly schedules. It also supports group and scene functionalities. The identify cluster facilitates Finding & Binding commissioning, serving as either initiator or target. Moreover, the thermostat client allows remote control through another thermostat, while the time client, the temperature sensor client or the occupancy client facilitate the reception of measurement values from a remote entity for local control purposes.

### 6.2. Installation Code

The Zigbee heating controller H1 comes with a preset link key that is generated from a unique installation code. The label, on which this code is printed as a two-dimensional barcode (QR code), can be found on the inner surface of the housing, after removing the rotary dial of the device H1. The format specified in [R6], section 5.4.8.1.1, is used with a full 128-bit installation code + 16-bit CRC. The QR code contains additional information as illustrated in the following example:

```
Z:001FEE000000998B$I:D4A2486AC24F9723A5C639B26916D505B189%G$M:H1%M:10F2
```

The content string initiates with an EUI-64 address (prefixed by "Z:"). It is followed by a complete 128-bit installation code along with its 16-bit CRC checksum (prefixed by "I:"). Following that are the model identifier string "H1" (prefixed by "M:") and the manufacturer code 0x10f2 (prefixed by "M:").

### 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 H1 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>[1]</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>[2]</sup>	<b>Parent announcement/response</b> This is part of the Zigbee 2015 end-device child management feature.
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>[3]</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>[3]</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 to control the heating unit connected to output.

This is a Finding & Binding initiator 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.
0x0001	Inbound (Server)	<b>Power Configuration</b> Allows to obtain the detailed information about the device's power source and configure under/over voltage alarms.
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.
0x000a	Inbound (Server)	<b>Time</b> Expose the internal time information.
0x0020	Inbound (Server)	<b>Poll Control</b> Provides a mechanism to handle the MAC Data Request rate when functioning as a sleepy end device. The server side encompasses various attributes related to the MAC Data Request rate for the end device. At pre-defined intervals, it sends a check-in query to the client to determine if the client wishes to switch to the fast polling mode for responsive data transaction.
0x0201	Inbound (Server)	<b>Thermostat</b> Provides an interface to the functionality of a thermostat. Allow to obtain basic information like local temperature, Occupancy, HeatingDemand, Heating Set point etc. Allow to configure the device's behavior depending on the specified operating environment and the system mode etc. Allow to configure the device's behavior according to the specified weekly schedules.
0xfc57	Inbound (Server)	<b>Works With All Hubs</b> This is a manufacturer-specific cluster with the manufacturer code <b>0x1217</b> . It provides the proprietary functionality of the Amazon Works With All Hubs (WWAH) server cluster. The goal of this cluster is to enable hubs to provide improved security, interoperability, and customer experience for devices supporting this cluster <a href="#">[R9]</a> .
0x0003	Outbound (Client)	<b>Identify</b> Allows to identify targets during Finding & Binding commissioning
0x000a	Outbound (Client)	<b>Time</b> Synchronizes the internal real-time clock with a Zigbee time server in the network, especially one linked and synchronized with an internet time server.
0x0019	Outbound (Client)	<b>OTA</b> Provides a standard way to upgrade devices in the network via OTA messages. 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.
0x0201	Outbound (Client)	<b>Thermostat</b> Receives attribute reports from other thermostat servers.
0x0402	Outbound (Client)	<b>Temperature Measurement</b> Receives attribute reports from other temperature sensors.
0x0405	Outbound (Client)	<b>Relative Humidity Measurement</b> Receives attribute reports from other humidity sensors.

Cluster	Direction	Description
0x0406	Outbound (Client)	<b>Occupancy Sensing</b> Receives attribute reports from other occupancy sensors.

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.

### 6.4.1. 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> "H1"
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, "20220921-DE-FB1" refers to a device built September 21, 2022 in Germany and programmed with factory block format version 1.
0x0007	enum8, read-only	<b>PowerSource</b> Always set to be battery-powered.
0x0008	enum8, read-only	<b>GenericDevice-Class</b> "Unspecified" by default. Might be set by commissioning software during installation
0x0009	enum8, read-only	<b>GenericDevice-Type</b> "Unspecified" by default. Might be set by commissioning software during installation
0x000a	octstr, read-only	<b>ProductCode</b> Specifies a code for the product.
0x000b	string, read-only	<b>ProductURL</b> Specifies a link to a web page containing specific product information. "www.ubisys.de/products/h1"
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
0x4000	enum8, read-only	<b>SWBuildID</b> Represents a detailed, manufacturer-specific reference to the version of installed software.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that this implementation adheres to.

Cluster commands supported:

None

## 6.4.2. Power configuration Cluster (Server)

The cluster specifies detailed information about H1's power source.

Attributes supported:

Attribute	Type	Description
0x0020	unsigned8 read-only, reportable	<b>BatteryVoltage</b> Specifies the current measured battery voltage, in units of 100mV.
0x0021	unsigned8 read-only, reportable	<b>BatteryPercentageRemaining</b> Specifies the remaining battery life as a half integer percentage of the full battery capacity with a range between 0% and 100%.
0x0031	enum8 persistent	<b>BatterySize</b> Specifies the type of battery being used by H1. "AA"
0x0033	unsigned8 read-only	<b>BatteryQuantity</b> Specifies the number of battery cells used to power the device H1. "2"
0x0034	unsigned8 persistent	<b>BatteryRatedVoltage</b> Specifies the rated voltage of the battery being used in H1, measured in units of 100mV. "1.5V"
0xFFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that this implementation adheres to.

Cluster commands supported:

None

## 6.4.3. Identify Cluster (Server)

This endpoint can be applied either as a Finding & Binding target or an initiator according to different applicable scenarios.

Attributes supported:

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

Cluster commands supported:

Command	Description
0x00	<b>Identify</b> Allows accepting the command and put the device H1 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.

In the identify mode, the symbol as a F&B target will flash on the H1's display for the specified identify duration.

## 6.4.4. 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.
0xFFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that 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.

#### 6.4.5. 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.
0xFFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that 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.

Command	Description
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

### 6.4.6. Time Cluster (Server)

Expose the internal time information.

Attributes supported:

Attribute	Type	Description
0x0000	UTC global-scope	<b>Time</b> Holds the time value of a real-time clock, in UTC time format. It may not be synchronized to UTC, indicated by the value of the TimeStatus attribute.
0x0001	Bitmap8, global-scope	<b>TimeStatus</b> As bitmap attribute indicating the status of the Time attribute.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that this implementation adheres to.

Cluster commands supported:

None

### 6.4.7. Poll Control Cluster (Server)

Attributes supported:

Attribute	Type	Description
0x0000	Unsigned32 persistent	<b>Check-inInterval</b> Specifies the amount of time between check-ins by the server with a remote poll control client, measured in quarterseconds.
0x0001	Unsigned32 read-only	<b>LongPollInterval</b> Specifies the maximum amount of time in quarterseconds between MAC Data Requests from the local end device to its router parent.
0x0002	Unsigned16 read-only	<b>ShortPollInterval</b> Specifies the amount of time in quarterseconds that the local end device waits between sending MAC Data Requests to its router parent when the local end device is expecting data transaction (i.e., in fast poll mode).
0x0003	Unsigned16 persistent	<b>FastPollTimeout</b> Specifies the amount of time in quarterseconds that the local end device will stay in fast poll mode by default.
0x0004	Unsigned32 read-only	<b>Check-inIntervalMin</b> Specifies the minimum value for the check-in interval to protect against the check-in interval being set too low and draining the battery of the local end device.
0x0005	Unsigned32 read-only	<b>LongPollIntervalMin</b> Specifies the minimum value for the long-poll interval to protect against another device setting the value to a too short value resulting in an inadvertent power drain on the local end device.
0x0006	Unsigned16, read-only	<b>FastPollTimeoutMax</b> Specifies the maximum value for the fast-poll timeout to avoid it being set to a too high value resulting in an inadvertent power drain on the local end device.
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Check-in</b> Sends out this command to find out if Poll Control cluster clients, with which it is paired, are interested in having it entering fast polling mode so that the local end device becomes responsive for pending data transactions.
0x01	<b>Fast Poll Stop</b> Receives this command from a paired client and stops the fast poll mode initiated before.
0x02	<b>Set Long Poll Interval</b> Receives this command from a paired client to set the read-only long-poll interval accordingly, if acceptable.
0x03	<b>Set Short Poll Interval</b> Receives this Command from a paired client to set the read-only short-poll interval accordingly, if acceptable.

## 6.4.8. 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 locally or remotely (over the network) measured temperature value in the heating zone.
0x0001	Signed16, read-only, reportable	<b>OutdoorTemperature</b> Indicates the outdoor temperature value in the heating zone. This might be measured using remote zigbee temperature or environmental
0x0002	Bool, read-only, reportable	<b>Occupancy</b> Determines whether the heated 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.
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%.
0x0012	Signed16, reportable	<b>OccupiedHeatingSetpoint</b> Specifies the heating mode setpoint when the room is occupied.
0x0014	Signed16, reportable	<b>UnoccupiedHeatingSetpoint</b> Specifies the heating mode setpoint when the room is unoccupied.
0x001b	enum8, reportable	<b>ControlSequenceOfOperation</b> Specifies the overall operating environment of H1, and thus the possible system modes that H1 can operate in. H1 supports "Heating Only" option.
0x001c	enum8, reportable	<b>SystemMode</b> Specifies the current operating mode of the thermostat. H1 supports "Off", "Auto" and "Heat" the three options. "Heat" by default
0x001e	enum8, reportable	<b>RunningMode</b> Represents the running mode of H1, i.e. the actual mode it is currently in. H1 supports "Off" and "Heat" the two options.
0x0020	enum8, read-only	<b>StartOfWeek</b> Specifies the day of the week that H1 considers to be the start of week for weekly set point scheduling. "Monday"
0x0021	Unsigned8, read-only	<b>NumberOfWeeklyTransitions</b> Specifies the number of weekly schedule transitions H1 is capable of handling. "70"



Attribute	Type	Description
0x0022	Unsigned8, read-only	<b>NumberOfDailyTransitions</b> Specifies the number of daily schedule transitions H1 is capable of handling. "10"
0xFFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that this implementation adheres to.

Manufacturer-specific attributes supported:

Attribute	Type	Description
0x0010	Signed8, persistent	<b>TemperatureOffset</b> <sup>[4]</sup> Specifies the temperature offset for the locally measured temperature value. This is to account for situations where the H1 is mounted in a place with little to no air flow and as a consequence the heat stowing.
0x0011	Signed16, persistent	<b>DefaultOccupiedHeatingSetpoint</b> <sup>[6]</sup> Specifies the default heating setpoint during occupancy, representing the targeted temperature when a recurring weekly schedule ends without a follow-up schedule.
0x0012	Boolean, persistent	<b>VacationMode</b> <sup>[6]</sup> <sup>[7]</sup> Indicates whether vocation mode is activated. If active, the corresponding vocation schedule, if defined, should be applied.
0x0013	Signed16, read-only	<b>RemoteTemperature</b> <sup>[9]</sup> Indicates the remotely measured temperature value, accessible through attribute reports. For heating regulation, a received remote temperature value, as long as valid, takes precedence over the locally measured one.
0x0014	Unsigned8, persistent	<b>RemoteTemperatureValidDuration</b> <sup>[9]</sup> Specifies the duration period in seconds, during which a remotely measured temperature value remains valid since its reception as attribute report.
0x0020	Signed16, persistent	<b>ProportionalGain</b> <sup>[9]</sup> Specifies the proportional gain (Kp) of the applied PI controller on H1
0x0021	Signed8, persistent	<b>ProportionalShift</b> <sup>[10]</sup> Specifies the proportional down-scaling term, noted as shift value (N), of the applied PI controller on H1
0x0022	Signed16, persistent	<b>IntegralFactor</b> <sup>[11]</sup> Specifies the integral factor of the applied PI controller on H1

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.
0x01	<b>Set Weekly Schedule</b> Receives this command from a client to update its weekly set-points schedule. The weekly schedule of set-points, stored in the H1, becomes effective upon its reception.  <ul style="list-style-type: none"> <li>- Receiving this command, the H1 is enabled to create a new schedule if none available, add a new schedule to the existing ones or modify the existing ones on the associated days.</li> <li>- Setting the field <a href="#">NumberOfTransitionsForSequence</a> in the header to be 0, the H1 is enabled to delete the existing schedules on the associated days.</li> <li>- The H1 does not support receiving multiple <a href="#">Set Weekly Schedule</a> commands if the field <a href="#">NumberOfTransitionsForSequence</a> is set to be more than 10.</li> </ul>
0x02	<b>Get Weekly Schedule</b> Receives this command from a client to provide the existing weekly set-point schedule that conforms to the requested mode and days.
0x03	<b>Clear Weekly Schedule</b> Receives this command from a client to clear <b>all</b> existing weekly schedules currently stored in the H1.

#### 6.4.9. Works With All Hubs Cluster (Server)

Works With All Hubs cluster is a manufacturer-specific cluster from Amazon, identifiable through the

cluster ID 0xfc57 and manufacturer ID 0x1217 and can exist at most only once on each device node [R9]. This cluster provides an interface for enabling various features of the Works With All Hubs cluster on H1.

Manufacturer-specific attributes supported:

Attribute	Type	Description
0x0002	Boolean, read-only, persistent	<b>DisableOTADowngrades</b> Set the attribute to TRUE if the OTA client shall prohibit downgrades to an earlier version
0x0003	Boolean, read-only, persistent	<b>MGMTLeaveWithoutRejoinEnabled</b> Set the attribute to FALSE, the node SHALL ignore MGMT Leave Without Rejoin commands that do not include APS encryption using the TC link key.
0x0004	Unsigned8, read-only	<b>NWKRetryCount</b> Number of network-level retries that will be made, in a row, in case that a MAC data request fails. "3"
0x0005	Unsigned8, read-only	<b>MACRetryCount</b> Number of MAC Level Retries the device performs when sending a unicast command, not including APS retries. "3"
0x0006	Boolean, read-only, persistent	<b>RouterCheckInEnabled</b> set to TRUE if the router's check-in algorithm has been enabled through the 'Enable Periodic Router Check-Ins' command. "FALSE"
0x0007	Boolean, read-only, persistent	<b>TouchlinkInterpanEnabled</b> set the attribute to FALSE if support for Touchlink Interpan messages has been disabled On the device. "TRUE"
0x0008	Boolean, read-only, persistent	<b>WWAHParentClassificationEnabled</b> Set the attribute to TRUE, then the device SHALL enable the WWAH Parent Classification Advertisements feature. "FALSE"
0x0009	Boolean, read-only, persistent	<b>WWAHAppEventRetryEnabled</b> Set the attribute to TRUE if the WWAH Application Event Retry Algorithm is enabled. "FALSE"
0x000A	Unsigned8, read-only, persistent	<b>WWAHAppEventRetryQueueSize</b> This attribute is the queue size for re-delivery attempts in the WWAH Application Event Retry Algorithm. "32"
0x000B	Boolean, read-only, persistent	<b>WWAHRejoinEnabled</b> Set this attribute TRUE if the WWAH Rejoin Algorithm (described below) is enabled. "FALSE"
0x000C	Unsigned8, read-only, persistent	<b>MACPollFailureWaitTime</b> This attribute describes the time in seconds the device waits before retrying a data poll when a MAC level data poll fails for any reason. "2"
0x000D	Boolean, read-only, persistent	<b>ConfigurationModeEnabled</b> Set the attribute TRUE, the device will accept all ZDO commands without APS encryption. Set the attribute FALSE, all ZDO commands except those specified in [R9] should be disabled (unless sent encrypted using the Trust Center Link Key)
0x000E	Unsigned8, read-only	<b>CurrentDebugReportID</b> This attribute depicts the ID of the current debug report stored on the node. A value of 0x00 indicates that no debug report is available. "0"
0x000F	Boolean, read-only, persistent	<b>TCSecurityOnNwkKeyRotationEnabled</b> Set the attribute TRUE, the node processes network key rotation commands, specifically APS Transport Key, which are sent via unicast and are encrypted by the Trust Center Link Key.
0x0010	Boolean, read-only, persistent	<b>WWAHBadParentRecoveryEnabled</b> Set the attribute TRUE, the end device enables the WWAH Bad Parent Recovery feature described as in [R9]. "FALSE"

Attribute	Type	Description
0x0011	Unsigned8, read-only, persistent	<b>PendingNetworkUpdateChannel</b> This attribute describes the channel number of the only channel the device accepts in a ZDO Mgmt Network Update command (and/or a Mgmt Network Enhanced Update command).
0x0012	Unsigned16, read-only, persistent	<b>PendingNetworkUpdatePANID</b> This attribute contains the only short PAN ID the device SHALL accept in a NLME Network Update command.
0x0013	Unsigned16, read-only, persistent	<b>OTAMaxOfflineDuration</b> This attribute describes the maximum time in seconds the device may be unavailable after completing its OTA download (while restarting, etc). "60"
0xFFFD	unsigned16	<b>ClusterRevision</b> The version of the cluster specification that this implementation adheres to.

Cluster commands supported:

Command	Description
0x00	<b>Enable APS Link Key Authorization</b> This command enforces that all cluster commands for the cluster have APS level security.
0x01	<b>Disable APS Link Key Authorization</b> This command removes the enforcement of APS level security on cluster commands.
0x02	<b>APS Link Key Authorization Query</b> On receipt of this command, a device generates an appropriate <a href="#">APS Link Key Authorization Query Response command</a> and unicast it to the requestor.
0x03	<b>Request New APS Link Key</b> On receipt of this command, a device requests a new Trust Center Link Key from the Trust Center.
0x04	<b>Enable WWAH App Event Retry Algorithm</b> This command enables using the WWAH App Event Retry Algorithm to increase reliability of these events when the device temporarily loses contact with the hub.
0x05	<b>Disable WWAH App Event Retry Algorithm</b> This command disenables using the WWAH App Event Retry Algorithm.
0x06	<b>Request Time</b> This command enables a device to obtain the current values of Time Cluster attributes from the Time Cluster Server.
0x07	<b>Enable WWAH Rejoin Algorithm</b> This command enables applying WWAH Rejoin Algorithm on end devices to recover and reconnect to the Zigbee network faster when connection is lost.
0x08	<b>Disable WWAH Rejoin Algorithm</b> This command disables applying WWAH Rejoin Algorithm and devices are supposed to go back to the default behavior.
0x09	<b>Set IAS Zone Enrollment Method</b> This command is used to dynamically change among the 3 enrollment methods of an IAS Zone Server specified in ZCL specification.
0x0A	<b>Clear Binding Table</b> On receipt of this command, a device clears its binding table.
0x0B	<b>Enable Periodic Router Check-Ins</b> This command enables the Periodic Router Check-in WWAH functionality with the Check-In Interval specified.
0x0C	<b>Disable Periodic Router Check-Ins</b> This command disables the router periodic check-in behavior (if enabled).
0x0D	<b>Set MAC Poll Failure Wait Time</b> This command enables to update the MACPollFailureWaitTime attribute with the Wait Time value passed.
0x0E	<b>Set Pending Network Update</b> This command enables updating the PendingNetworkUpdateChannel and PendingNetworkUpdatePANID attributes with the Channel and PAN ID values passed, to protect the device from unauthorized Network Update.
0x0F	<b>Require APS ACKs on Unicasts</b> This command enforces that all unicast commands have APS ACKs enabled.
0x10	<b>Remove APS ACKs on Unicasts Requirement</b> This command enables to roll back any changes caused by the <a href="#">Require APS ACKs on Unicasts</a> command. The device can default back to its normal settings.
0x11	<b>APS ACK Requirement Query</b> This command is meant to query the current <a href="#">Require APS ACKs on Unicasts</a> settings.

Command	Description
0x12	<b>Debug Report Query</b> This command enables requesting the debug report based on the specified Debug Report ID.
0x13	<b>Survey Beacons</b> On receipt of this command, the device performs a scan for beacons advertising the device's network (using the default scan duration), generate a Survey Beacons Response command and unicast it to the requestor.
0x14	<b>Disable OTA Downgrades</b> On receipt of this command, a device disallows downgrades of all components of its firmware.
0x15	<b>Disable MGMT Leave Without Rejoin</b> On receipt of this command, a device responds with <code>NOT_AUTHORIZED</code> to MGMT Leave Without Rejoin commands that don't contain APS encryption using the TC Link key.
0x16	<b>Disable Touchlink Interpan Message Support</b> On receipt of this command, a device ignores Touchlink interpan messages.
0x17	<b>Enable WWAH Parent Classification</b> On receipt of this command, a device enables the WWAH Parent Classification Advertisements feature.
0x18	<b>Disable WWAH Parent Classification</b> On receipt of this command, a device disables the WWAH Parent Classification Advertisements feature.
0x19	<b>Enable TC Security On Nwk Key Rotation</b> On receipt of this command, a device processes only network key rotation commands, specifically APS Transport Key, which are sent via unicast and are encrypted by the Trust Center Link Key.
0x1A	<b>Enable WWAH Bad Parent Recovery</b> On receipt of this command, a device enables the WWAH Bad Parent Recovery feature.
0x1B	<b>Disable WWAH Bad Parent Recovery</b> On receipt of this command, a device disables the WWAH Bad Parent Recovery feature.
0x1C	<b>Enable Configuration Mode</b> This command enables to protect the device from unauthorized configuration changes during normal optional, i.e., to put a device back into its default state where configuration changes are allowed.
0x1D	<b>Disable Configuration Mode</b> On receipt of this command, a device disables processing of all ZDO commands that have not been encrypted using the Trust Center Link key.
0x1E	<b>Use Trust Center for Cluster</b> This command forces the device to use ONLY the Trust Center as the cluster server for the set of clusters specified and persists this configuration across reboots.
0x1F	<b>Trust Center for Cluster Server Query</b> On receipt of this command, a device generates an appropriate <a href="#">Trust Center for Cluster Server Query Response</a> command and unicast it to the requestor.

#### 6.4.10. Identify Cluster (Client)

The Identify cluster client takes on the role of the Finding & Binding initiator. Once activated, H1 initiates the identification process by broadcasting identify query request command to detect target(s) operating in identify mode.

Cluster commands supported:

Command	Description
0x01	<b>Query Identify</b> Broadcasts this command to identify suitable binding targets. Once target devices are identified, they will be queried for matching descriptors.

#### 6.4.11. Time Cluster (Client)

The Time cluster client locates Time cluster servers in the same network to synchronize its internal real-time clock, preferable one that is connected to the internet and provides a high-quality time-base. Valid timing information is essential for utilizing pre-defined weekly set-point schedules, whether it's automatically synchronized with a remote time-base or manually set through the local MMI on H1.

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

The current image type for H1 is 0x7B2D <sup>[12]</sup>. 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 H1 together with the ZigBee/Ethernet Gateway ubisys G1, the latest firmware for H1 will always be available automatically and you do not need to request it explicitly.

#### 6.4.13. Thermostat Cluster (Client)

The Thermostat cluster client receives attribute reports for setpoints, local temperature, occupancy, system mode etc. from a leader <sup>[13]</sup> thermostat in a group of thermostats. This allows the leader to align one or more thermostats in the same space.

#### 6.4.14. Temperature Measurement (Client)

The Temperature Measurement cluster client receives attribute reports from remote temperature sensors and interprets the received measured values as remote temperature, internally maintained through the manufacturer-specific attribute RemoteTemperature. H1 is equipped with a temperature sensor internally. Hence it supports temperature acquisition not only via internal mechanisms but also via receiving temperature reports sent by external sensors. For heating regulation, a received remote temperature value, as long as valid, takes precedence over the locally measured one.

#### 6.4.15. Relative Humidity Measurement (Client)

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

#### 6.4.16. Occupancy Sensing (Client)

The Occupancy Sensing cluster client receives attribute reports from occupancy sensors and interprets the measured values as occupancy in the heating space. This will bring different heating 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.

- [1] Available in ZigBee stack version 1.60 and above. Legacy ZCP requirement – do not use in applications
- [2] Available in Zigbee stack version 1.56 and above.
- [3] Available in Zigbee stack version 1.61 and above.
- [4] Attribute ID changed since application version 1.5.0. Prior, it used to be 0x0003.
- [5] Attribute ID changed since application version 1.5.0. Prior, it used to be 0x0004.
- [6] Since application version 1.4.0. Prior, the attribute was unsupported.
- [7] Attribute ID changed since application version 1.5.0. Prior, it used to be 0x0005.
- [8] Since application version 1.5.0. Prior, the attribute was unsupported.
- [9] Attribute ID changed since application version 1.5.0. Prior, it used to be 0x0013.
- [10] Attribute ID changed since application version 1.5.0. Prior, it used to be 0x0014.
- [11] Attribute ID changed since application version 1.5.0. Prior, it used to be 0x0015.
- [12] Since application version 1.5.0. Prior, the image type used to be 0x7B0D.
- [13] An example of a suitable leader is the Legrand in-wall thermostat UI Model No. 0 663 40

## 7. Thermostat Heating Setpoints Schedules

H1 supports automated heating regulation based on pre-defined temperature setpoint schedules on a weekly basis. Currently, it supports two approaches for creating, adjusting, removing, or viewing one or more weekly heating setpoint schedules on associated days:

- Approach 1: Utilize the MMI on H1

Operations such as addition, modification or removal of weekly setpoint schedules can be conveniently and efficiently carried out by directly utilizing the MMI on H1. For information on how to set and configure heating schedules using MMI, please refer to *Chapter 4. Schedules* of the [Heating Control H1 - Manual](#).

- Approach 2: From a management entity on a ZigBee network, utilize the radio interface to transmit the Zigbee ZCL Thermostat cluster commands such like [Set Weekly Schedule](#), [Get Weekly Schedule](#), or [Clear Weekly Schedule](#) to H1.

An example of such a management entity is the ubisys Gateway G1, in collaboration with the ubisys smart facility application.

Both approaches are basically compatible with each other, but it's important to highlight the major differences when employing them to set heating setpoints for automated heating regulation through H1.

### 7.1. Weekdays configuration

The heating setpoint schedules on H1 can be configured for days on a weekly basis.

Approach 1: MMI	Approach 2: Over-the-air ZCL commands
<p>If no schedule has been set up yet, you can choose from the following combinations of weekdays to create unified schedules for the respective days:</p> <ul style="list-style-type: none"><li>- <i>Option 1</i>: all weekdays (Monday - Sunday)</li><li>- <i>Option 2</i>: all workdays &amp; weekend (Monday-Friday &amp; Saturday-Sunday)</li><li>- <i>Option 3</i>: vacation days</li></ul> <p>The <i>Option 1</i> and <i>Option 2</i> will continue to be accessible until schedule differentiation is added for at least one of the respective days. Subsequently, configuration of setpoint schedules must be carried out for each weekday individually.</p> <p>The <i>Option 3</i> is always accessible.</p>	<p>The ZCL Thermostat cluster commands <a href="#">Set Weekly Schedule</a> and <a href="#">Get Weekly Schedule</a> include a field that specifies the weekdays or vacation days at which the heating setpoint schedules are associated to. This field is in bitmap format, allowing scheduled setpoints to potentially overlap across multiple days.</p> <p>H1 supports receiving the <a href="#">Set Weekly Schedule</a> and <a href="#">Get Weekly Schedule</a> commands with the combination of any weekday specified and vacation days by the user.</p>

### 7.2. Format of setpoint schedules

If you create a setpoint schedule using the MMI on H1, the internally applied and stored schedule format, the so-called schedule slot format, is implemented. It differs sometimes slightly from the one created when receiving a ZCL Thermostat cluster [Set Weekly Schedule](#) command that utilizes the ZCL standard schedule transition format. H1 can handle both formats individually, but it doesn't support a combination of them for the same weekly schedules. In other words, you must define weekly schedules in one of the two formats, but not a mix of them.

---

**Format variant 1 :  
Schedule Slot**

---

Each schedule slot actually consists of two transitions:

- **Start transition**, which contains two fields:
  - (a). time of day, in minutes since midnight, which specifies the start time of a heating duration period
  - (b). a heating setpoint to be applied at the associated start transition time
- **End transition**, which contains two fields:
  - (a). time of day, in minutes since midnight, which specifies the end time of a heating duration period
  - (b). an end-mark with the hexadecimal integer value 0x8000 indicating the end of applying the heating setpoint at this associated end transition time

This is the default format used for adding and further modifying weekly schedules through the MMI on H1.

Consider a sequence of schedule slots:

- Since each schedule slot has a defined duration with a specified start and end time, there may be instances during the day when H1 does not have a heating setpoint schedule. In such scenarios, the target heating temperature setpoint is determined based on the state of the [Occupancy](#) attribute. It applies either the [Default Occupied Heating Setpoint](#) or [Unoccupied Heating Setpoint](#) attribute value until the start time of the next available schedule slot.
- On H1 through MMI, this type of schedule slot is only applicable to a specified combination of week days.

When transmitting a ZCL Thermostat cluster [Set Weekly Schedule](#) command to H1, you can also submit weekly schedule definitions in the slot format:

- The count of schedule transitions included must be an even number  $2m$ , so that  $m$  pairs of transition slots can be generated in fact.
- The heating setpoint value of the second transition in each transition pair should be set to the hexadecimal end-mark value 0x8000.

---

**Format variant 2 :  
ZCL Standard Schedule Transition**

---

The definition of ZCL standard schedule transition for heating setpoint, applied for the ZCL Thermostat cluster command [Set Weekly Schedule](#) and [Get Weekly Schedule](#), can be found in the ZCL specification [R4].

Each standard schedule transition includes two fields:

- (a). time of day, in minutes since midnight, which specifies the start time of a transition
- (b). a heating setpoint to be applied since the transition start time

The MMI on H1 does not support adding and further modifying weekly schedules in the ZCL standard schedule transition format.

Consider a sequence of schedule transitions:

- The start time of a schedule transition is inherently the end time of the proceeding transition.
- H1 never has duration periods without scheduled heating temperature setpoint once at least one schedule transition is defined.
- A heating setpoint schedule created in this manner may extend across several days until the commencement of any subsequent schedule transition.

Consider H1 is already equipped with weekly schedules set up via the MMI, it receives a ZCL Thermostat cluster [Set Weekly Schedule](#) command containing schedule definitions in the ZCL standard schedule transition format,

- it will result in the removal of the whole stored weekly schedules on H1.
  - The newly received schedules will then be applied.
  - The configuration of weekly schedules through MMI will be in this case deactivated until all schedules using the standard ZCL pattern are entirely deleted at a later time.
- 

Regardless of the schedule format currently in use, H1 always shows the current scheduled temperature setpoint in the bottom left corner of the display, if available.

While the formats differ, the fundamental data structure, encompassing the fields of time of day and temperature setpoint, remains identical in both format definitions. This means you can always employ a ZCL Thermostat cluster [Set Weekly Schedule](#) command via over-the-air transmission to set up weekly schedules, whether using the ubisys schedule slot format or the ZCL standard schedule transition format.

### 7.3. Number of daily heating setpoint schedules

By default, the attribute [NumberOfDailyTransitions](#) is set to 10, indicating the H1's capacity to handle daily schedule transitions. The variations in the format of setpoint schedules lead to different maximum numbers of daily heating setpoint schedules:

---

**Number of heating setpoints for Sequence  
based upon Schedule Slot : 5**

---

You can define up to five schedule duration periods per day.

---

**Number of heating setpoints for Sequence  
based upon ZCL Standard Schedule Transition : 10**

---

You can define up to ten schedule transition changes per day.

---



## 7.4. Schedule Configuration Examples

Here are a few ZCL Thermostat cluster command frame examples highlighting that the *Approach 2* is utilized to configure temperature setpoint schedules. These over-the-air ZCL Thermostat cluster commands should be always sent to the endpoint #1 and cluster 0x0201 of a target H1. For creating the ZCL Thermostat cluster `SetWeeklySchedule` command, both the schedule slot and the ZCL standard schedule transition formats for schedule configuration are showcased.

Examples of applying the *Approach 1* to enable temperature setpoint schedules can be found in [Chapter 4. Schedules](#) of the [Heating Control H1 - Manual](#).

- Example 1. Clear all existing heating setpoint schedules

*ZCL thermostat cluster command Clear Weekly Schedule*

```
0x03
```

Item	Description
0x03	Identifier of the command <code>ClearWeeklySchedule</code>

- Example 2. Clear the heating setpoint schedules for specific days
  - Clear the setpoint schedules on Tuesday:

*ZCL thermostat cluster command Set Weekly Schedule*

```
0x01 0x00 0x04 0x01
```

Item	Description
0x01	Identifier of the command <code>SetWeeklySchedule</code>
0x00	Field <code>NumberOfTransitionsForSequence</code> set to 0
0x04	Field <code>DayOfWeekForSequence</code> set to 0x04 (Tuesday)
0x01	Field <code>ModeForSequence</code> set 0x01 (Heat)

- Clear the heating setpoint schedules from Monday to Friday:

*ZCL thermostat cluster command Set Weekly Schedule*

```
0x01 0x00 0x3E 0x01
```

Item	Description
0x01	Identifier of the command <code>SetWeeklySchedule</code>
0x00	Field <code>NumberOfTransitionsForSequence</code> set to 0
0x3E	Field <code>DayOfWeekForSequence</code> set to 0x3E (Monday-Friday)
0x01	Field <code>ModeForSequence</code> set 0x01 (Heat)

- Clear the heating setpoint schedules for vacation days:

*ZCL thermostat cluster command Set Weekly Schedule*

```
0x01 0x00 0x80 0x01
```

Item	Description
0x01	Identifier of the command <a href="#">SetWeeklySchedule</a>
0x00	Field <a href="#">NumberOfTransitionsForSequence</a> set to 0
0x80	Field <a href="#">DayOfWeekForSequence</a> set to 0x80 (Away or Vacation)
0x01	Field <a href="#">ModeForSequence</a> set 0x01 (Heat)

- Example 3. Query the heating setpoint schedules for specific weekdays

- Query the existing temperature setpoint schedules for weekend:

*ZCL thermostat cluster command Get Weekly Schedule*

```
0x02 0x41 0x01
```

Item	Description
0x02	Identifier of the command <a href="#">GetWeeklySchedule</a>
0x41	Field <a href="#">DaysToReturn</a> set to 0x41 (Saturday and Sunday)
0x01	Field <a href="#">ModeToReturn</a> set 0x01 (Heat)

- Query the existing temperature setpoint schedules for vacation days:

*ZCL thermostat cluster command Get Weekly Schedule*

```
0x02 0x80 0x01
```

Item	Description
0x02	Identifier of the command <a href="#">Get Weekly Schedule</a>
0x80	Field <a href="#">DaysToReturn</a> set to 0x80 (Vacation days)
0x01	Field <a href="#">ModeToReturn</a> set 0x01 (Heat)

- Example 4. Set the heating setpoint schedules for specific weekdays

- Set three heating setpoints on all workdays utilizing the ubisys schedule slot format:

Following settings are intended on H1:

Heating Period	Parameter	Value
1	Start Time	00:00 (0x0000)
1	End Time	06:00 (0x0168)
1	Target Temperature	15°C (0x05DC)
2	Start Time	06:00 (0x0168)
2	End Time	22:00 (0x0528)
2	Target Temperature	24°C (0x0960)
3	Start Time	22:00 (0x0528)
3	End Time	24:00 (0x05A0)
3	Target Temperature	15°C (0x05DC)

*ZCL thermostat cluster command SetWeeklySchedule*

```
0x01 0x06 0x3E 0x01 0x00 0x00 0xDC 0x05 0x68 0x01 0x00 0x80 0x68 0x01 0x60
0x09 0x28 0x05 0x00 0x80 0x28 0x05 0xDC 0x05 0xA0 0x05 0x00 0x80
```

Item	Description
0x01	Identifier of the command <a href="#">SetWeeklySchedule</a>

Item	Description
0x06	Field <code>NumberOfTransitionsForSequence</code> set to 0x06 indicating 3 slots, i.e., 6 transitions
0x3E	Field <code>DayOfWeekForSequence</code> set 0x3E (From Monday to Friday)
0x01	Field <code>ModeForSequence</code> set 0x01 (Heat)
0x00 0x00	Field <code>TransitionTime 1</code> set 0x0000 (StartTime 00:00)
0xDC 0x05	Field <code>HeatSetPoint 1</code> set 0x05DC (15°C)
0x68 0x01	Field <code>TransitionTime 2</code> set 0x0168 (EndTime 06:00)
0x00 0x80	Field <code>HeatSetPoint 2</code> set 0x8000 (EndMark)
0x68 0x01	Field <code>TransitionTime 3</code> set 0x0168 (StartTime 06:00)
0x60 0x09	Field <code>HeatSetPoint 3</code> set 0x0960 (24°C)
0x28 0x05	Field <code>TransitionTime 4</code> set 0x0528 (EndTime 22:00)
0x00 0x80	Field <code>HeatSetPoint 4</code> set 0x8000 (EndMark)
0x28 0x05	Field <code>TransitionTime 5</code> set 0x0528 (StartTime 22:00)
0xDC 0x05	Field <code>HeatSetPoint 5</code> set 0x05DC (15°C)
0xA0 0x05	Field <code>TransitionTime 6</code> set 0x05A0 (EndTime 24:00)
0x00 0x80	Field <code>HeatSetPoint 6</code> set 0x8000 (EndMark)

Upon receipt of this over-the-air ZCL Thermostat cluster `SetWeeklySchedule` command in the ubisys schedule slot format,

- **all** currently stored schedules will be removed, if they are defined in the ZCL standard schedule transition format, if any
  - the currently stored schedules for Monday to Friday will be replaced by the ones given in the received command frame, if any and applying the same slot format
  - the new schedules will be applied
  - the just added schedules can be modified either via MMI on H1 or upon receipt of further over-the-air ZCL thermostat schedule commands
  - the heating regulation on the specified weekdays occurs according to these new schedules
  - the heating regulation on other weekdays, in this case on Saturday and Sunday, remains unchanged
- Set three heating setpoints on all weekdays utilizing the ZCL standard schedule transition format:

Following settings are intended:

Setpoint Transition	Parameter	Value
1	Start Time	00:00 (0x0000)
1	Target Temperature	15°C (0x05DC)
2	Start Time	06:00 (0x0168)
2	Target Temperature	24°C (0x0960)
3	Start Time	22:00 (0x0528)
3	Target Temperature	15°C (0x05DC)

*ZCL thermostat cluster command SetWeeklySchedule*

```
0x01 0x03 0x3E 0x01 0x00 0x00 0xDC 0x05 0x68 0x01 0x60 0x09 0x28 0x05 0xDC
0x05
```

Item	Description
0x01	Identifier of the command <code>SetWeeklySchedule</code>
0x03	Field <code>NumberOfTransitionsForSequence</code> set to 0x03 indicating 3 transitions
0x3E	Field <code>DayOfWeekForSequence</code> set 0x3E (From Monday to Friday)
0x01	Field <code>ModeForSequence</code> set 0x01 (Heat)
0x00 0x00	Field <code>TransitionTime 1</code> set 0x0000 (StartTime 00:00)
0xDC 0x05	Field <code>HeatSetPoint 1</code> set 0x05DC (15°C)
0x68 0x01	Field <code>TransitionTime 2</code> set 0x0168 (StartTime 06:00)
0x60 0x09	Field <code>HeatSetPoint 2</code> set 0x0960 (24°C)
0x28 0x05	Field <code>TransitionTime 3</code> set 0x0528 (StartTime 22:00)
0xDC 0x05	Field <code>HeatSetPoint 3</code> set 0x05DC (15°C)

Upon receipt of this over-the-air ZCL Thermostat cluster `SetWeeklySchedule` command in the ZCL standard schedule transition format,

- **all** currently stored schedules will be removed, if they are defined in the ubisys schedule slot format, if any
- the currently stored schedules for Monday to Friday will be replaced by the ones given in the received command frame, if any and applying the same standard transition format
- the new schedules will be applied
- the just added schedules can be modified only upon receipt of further over-the-air ZCL commands that utilize the ZCL standard schedule transition format
- the heating regulation on the specified weekdays occurs according to these new schedules
- the heating regulation on other weekdays, in this case on Saturday and Sunday, may be affected as well, i.e., the last heating setpoint (15°C) on Friday could take into effect beyond the midnight to Saturday, till the next scheduled setpoint

## 8. Physical Dimension

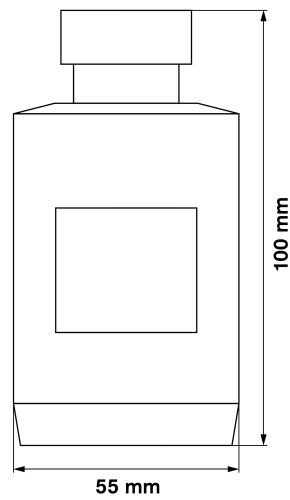


Figure 3. Heating Control H1 – Dimensions



If the ubisys heating control H1 does not fit on the valve of your radiator, you will probably need an adapter. The valve adapter is mounted between the H1 and your radiator valve. You can refer to the [Valve Adapter List](#) provided by ubisys. The list does not claim to be complete. For other valve types, other adapters may be required.

## 9. Ordering Information

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

Orders for products sold under different brand names need to be placed with the respective vendor.

For any customized versions, please contact ubisys sales. OEM customers may order these products with the dimming mode pre-configured to either “automatic”, “leading edge” or “trailing edge”.

<b>Vendor/Brand</b>	<b>Order Code</b>	<b>Description</b>
ubisys	1267	Heating Regulator H1

## 10. 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)

## 11. Declaration Of Conformity



We – ubisys technologies GmbH, Neumannstraße 10, 40235 Düsseldorf, Germany – declare under our sole responsibility that the ubisys Heating Regulator H1 with order codes as detailed in [Chapter 9](#) 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

Signature

Seal



## 12. Revision History

<b>Revision</b>	<b>Date</b>	<b>Remarks</b>
0.1	17/08/2022	Initial draft
1.0	04/12/2023	Initial Public Version

## 13. 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)