HF-Interfaces V1.4.0 (2017-12-18)

ULE Alliance Standard

Digital Enhanced Cordless Telecommunications (DECT); Ultra Low Energy (ULE);

Home Area Network Functional (HAN-FUN) Interfaces

Keywords

DECT, ULE, HAN, HAN-FUN

ULE ALLIANCE
Secretariat

Wabernstrasse 40
3007 Berne
Switzerland

T: +49 89 5166 2456 M: +49 160 9667 96966
Email: secretariat@ulealliance.org
http://www.ulealliance.org

Important notice

Individual copies of the present document can be downloaded from:
http://www.ulealliance.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ULE Alliance printers of the PDF version kept on a specific network drive within ULE Alliance Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status without further notice. Information on the current status of this and other ULE Alliance documents is available at
http://www.ulealliance.org

If you find errors in the present document, please send your comment to:
http://www.dect.org/content.aspx?id=35

Contents

Intellectual Property Rights

The Intellectual Property Rights (IPR) regulation is binding on all members and adopters participating in the ULE Alliance. Its purpose is to ensure the widest possible dissemination of the specifications adopted by the ULE Alliance while giving due weight and respect to the IPR of its members.

The IPR regulation can be found at http://www.ulealliance.org/downloads.aspx?c=w (Miscellaneous)

Foreword

This document has been produced by the ULE Alliance Technical Working Group.

The information in the present document is believed to be correct at the time of publication. However, Home Area Network Functional (HAN-FUN) may rapidly evolve, and consequently, it is possible that some of the information contained in the present document may become incomplete.

The present document is part of a multi-part deliverable covering the HAN-FUN protocol as identified below:

HF-Overview [REF 1]: Overview

HF-Protocol [REF 2]: Protocol Specification

HF-Service [REF 3]: Core Services & Interfaces

HF-Interfaces [REF 4]: Interface Library

HF-Profile [REF 5]: Profiles

HF-ULE-Interworking [REF 6]: HF &ULE Interworking

1 Scope

The present document specifies the functional HAN-FUN (HF) interfaces that can be implemented by HF units.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents:

[1] ETSI EN 300 175-1: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 1: Overview"

http://www.etsi.org/deliver/etsi_en/300100_300199/30017501/02.02.01_60/en_30017501v020201p.pdf

[2] Unicode® 10.0.0: "The Unicode® Standard Version 10.0 – Core Specification"

http://www.unicode.org/versions/Unicode10.0.0/UnicodeStandard-10.0.pdf

3 Definitions, Symbols and Abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

- *Italic* is used to indicate the name designation of attributes and commands.
- Device addresses are indicated as D'xxxx, where xxxx is a hexadecimal number up to four digits. This provides a compact notation for a HF device address. Usually appears as D'0 referring to the network main device, which has the address 0x0000.
- Group addresses are indicated as G'xxxx and have the same notation definitions as a device address.
- Unit IDs are indicated as U'xx, where xx is a hexadecimal number up to two digits. This provides a compact notation for a unit's ID. Usually appears as U'0 referring to the network management unit, with ID 0x00, that every HF device implements.
- Fully qualified HF network addresses are indicated as D'xxxx:U'xx. This compact notation is a combination of the previous two definitions.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

M	Provision Mandatory
O	Provision Optional
N/A	Non-applicable

The symbols here defined are applied to interfaces, attributes and commands and their fields in the present document if not explicitly otherwise stated. The interpretation of these status indications is as follows:

- Provision mandatory, means that the indicated interface, attributes, command or command field shall be implemented as described in the present document, and may be subject to testing.
- Provision optional, means that the indicated interface, attribute, command or command field may be implemented, and if implemented, the interface, attribute or command shall be implemented as described in the present document, and may be subject to testing.
- Non-applicable, means that the indicated interface, attribute, command or command field is not applicable in the defined context. As such, is should not be implemented and consequently not subject to testing.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

DECT	Digital Enhanced Cordless Telecommunications
DFS	DECT Forum Standard
HAN	Home Area Network
HAN-FUN (HF)	Home Area Network Functional
HF-IFL	HAN-FUN Interface Library
HF-PRF	HAN-FUN Profiles
ID	Identifier
LED	Light Emitting Diode
UID	Unique Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
WG	Working Group
XML	Extensible Mark-up Language

4 Introduction

The HAN-FUN Interface Library (HF-IFL) provides the functional building blocks for units defined in, and in compliance with the HF-Profile [REF 5] document.

Each interface is a collection of commands and attributes usable in units as either mandatory or optional. Each interface has also one of two possible roles associated with it – server or client. As an example, figure 1 depicts two different devices each with one unit implementing the same interface: unit 1 implements the server role while unit 2 implements the client role. In this setup both units can interact in a well-defined manner according to a client-server model easily identifying the controller device (client role) and the sensor/actuator device (server role).

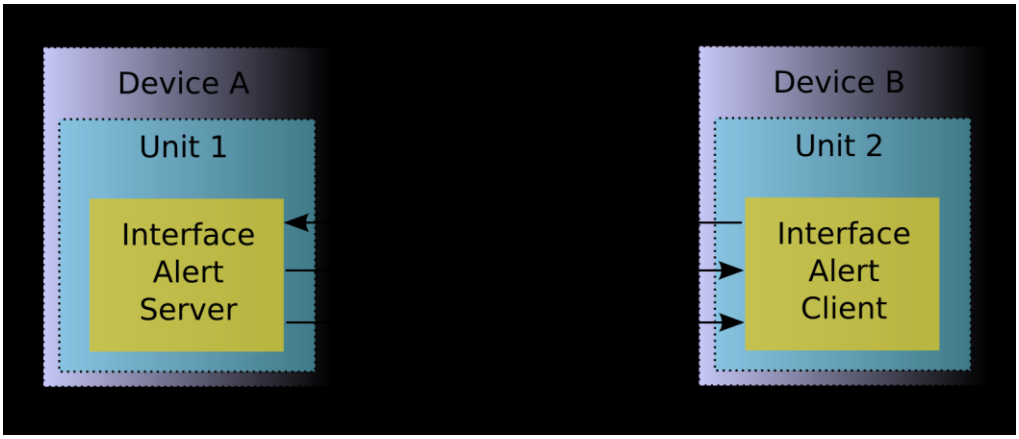

Figure 1 - Example of a Communication with Alert Interface

All available HF interfaces are listed in Table 1 in chapter 5. An interface is identifiable by a unique identification number and a unique human readable name. Table 1 also contains the range of interface IDs that can be used for proprietary extensions. It should be noted that since handling proprietary IDs is out of the scope of this document, collisions between proprietary interfaces are expected to occur.

4.1 HF-IFL Versioning

The HF-IFL will be extended to support more features, but once a HF-IFL document is released, the interfaces it contains will not be changed in a later revision. If an existing interface needs to be extended in functionality, a new interface needs to be created. This ensures on-going support for an interface and the continued operation, without requiring any update, of any system implementing a specific interface.

4.2 Recommendations

If an optional feature is implemented it must be implemented according the standard and not in a proprietary way.

5 HF Interfaces

For clarification, an HF interface is not required to define commands for get or set actions over its attributes, but may define such commands. This feature is already provided at the protocol level, as explained in the HF-Protocol document [REF 2]. . Therefore, commands defined in an interface should serve specific needs of that interface.

For a complete listing of the available HF interfaces with their unique identifiers (UIDs), see Table 1.

Table 1 - List of the Available HF Functional Interfaces

Functional Interfaces		
UID	**Name**	**Description**
Functional		
0x0100	Alert	Use when device wants to indicate an alert
0x0200	On-Off	Use to turn some device feature On or Off (you may also toggle it)
0x0201	Level Control	Use to set some device feature to a defined level
0x0202	Colour Control	Use to control the colour function of device
0x0203	Simple Keypad	Use to send key events from a keypad.
0x0300	Simple Power Metering	Use when device requires doing or providing measurements over electric quantities.
0x0301	Simple Temperature	Interface for simple temperature sensor
0x0302	Simple Humidity	Interface for simple humidity sensor
0x0303	Simple Thermostat	Interface for simple thermostat controller
0x0304	Simple Button	Use this interface to receive several button press related notifications
0x0305	Simple Visual Control	Use this interface to control some visual effects on some visual indicator (display, LED, lamp, etc)
0x0306	Air Pressure	Interface for an air pressure sensor
0x0307	Simple Light Sensor	Interface for simple light sensor
Reserved		
0x7F00 - 0x7FFE	Reserved	Use for proprietary features (e.g. technical , manufacture, etc)
0x7FFF	Reserved	Special UID.

5.1 Alert Interface

The Alert interface can be used by any device that requires sending/receiving an alert. The alert type will be specified by the profile where the interface is implemented, for example for a smoke detector the alert will indicate the presence of smoke, but for a motion detector the same alert will indicate that movement exists on the area covered by it.

This interface can support multiple alerts, from 1 up to 32.

5.1.1 Server Attributes

Table 2 - Alert Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	State	U32 (bitmask)	0x00000000 - 0xFFFFFFFF	Read Only	M
0x02	Enable	U32 (bitmask)	0x00000000 - 0xFFFFFFFF	Read / Write	M

5.1.1.1 State

State attribute indicates the state of an alert. A bit set to ”1” indicates an active alert, a bit set to ”0” indicates idle.

For example:

- a value of 0x00000000 means all alerts are idle;
- a value of 0x00000001 means Alert 0 is active while all others remain idle;
- a value of 0x00000021 means both Alert 0 and Alert 5 are active while all others remain idle;

5.1.1.2 Enable

Enable attribute indicates if an alert is enabled or disabled. A disabled alert MUST NOT trigger sending the *Status* command. A bit set to ”1” indicates alert is enabled, a bit set to ”0” indicates alert is disabled.

For example:

- a value of 0x00000000 means all alerts are disabled;
- a value of 0x00000002 means Alert 1 is enabled while all others remain disabled;
- a value of 0x00000104 means both Alert 2 and Alert 8 are enable while all others remain disabled;

5.1.2 Client Attributes

None.

5.1.3 Server to Client Commands

Table 3 - Implementation status of Alert Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Status	5.1.3.1	M	M	N/A

5.1.3.1 Status

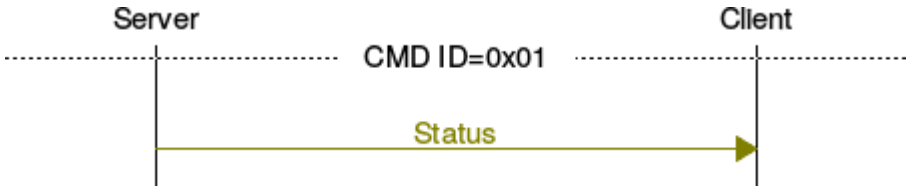

Figure 2 - Alert Interface, *Status* Command

This command, sent to a client implementation of the Alert interface, indicates the current state of all alerts. It is up to the profile to specify when this command should be sent.

The command must provide the information described in Table 4, organized according to Table 5.

Table 4 - Data in Payload of a *Status* Command

Field Name	Field Description	Type	Value	M/O
Profile UID	The unique identifier of the profile implementing this interface.	U16	0x0000 - 0xFFFF	M
State Attribute	The value of the *State* attribute, as defined in 5.1.1.	U32	0x00000000 - 0xFFFFFFFF	M

Table 5 - Data Ordering of Payload of a *Status* Command

8	7	6	5	4	3	2	1	Octet
Profile UID (MSB)								**1**
Profile UID (LSB)								**2**
State Attribute (MSB)								**3**
State Attribute								**4**
State Attribute								**5**
State Attribute (LSB)								**6**

5.1.4 Client to Server Commands

None.

5.2 On/Off Interface

This interface enables a device to be turned on/off (server role) or to turn on/turn off/toggle (client role) some feature. It allows, for example, turning on or off a siren, an LED or a relay.

5.2.1 Server Attributes

Table 6 - On-Off Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	State	U8	0x00 - 0x01	Read Only	M

5.2.1.1 State

State attribute indicates the current on/off state. It can take one of two values:

- The value 0x00 indicates it is off.
- The value 0x01 indicates it is on.

5.2.2 Client Attributes

None.

5.2.3 Server to Client Commands

None.

5.2.4 Client to Server Commands

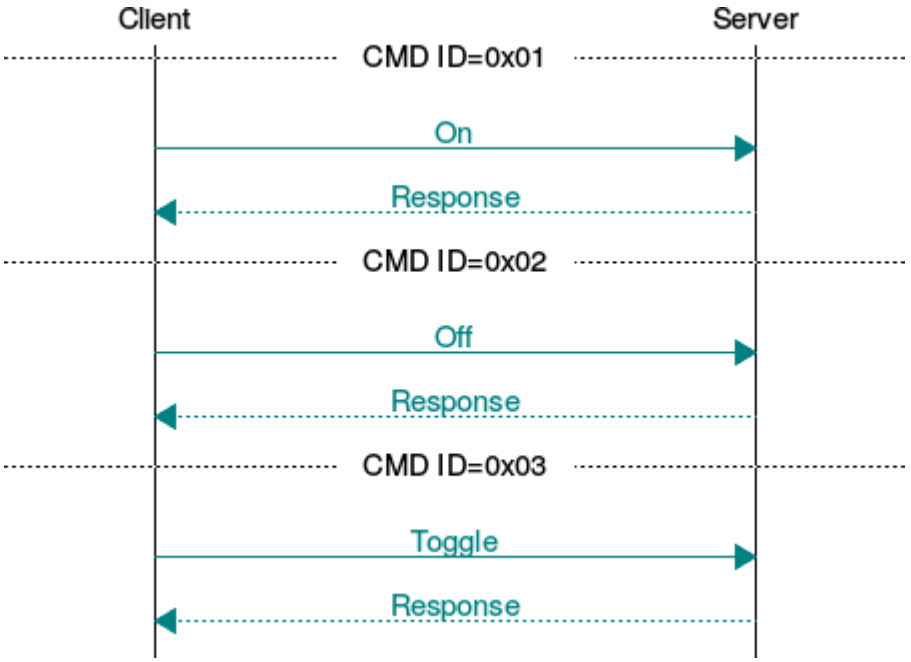

Figure 3 - On-Off Interface, Client Commands

Table 7 - Implementation status of On-Off Interface Client commands.

Command	Reference	Client Role	Server Role	Response
On	5.2.4.1	M*	M	O
Off	5.2.4.2	M*	M	O
Toggle	5.2.4.3	M**	M	O

* These commands are only mandatory when the *T*oggle command is NOT implemented.

** This command is only mandatory when the *On* and *Off* commands are NOT implemented

5.2.4.1 On

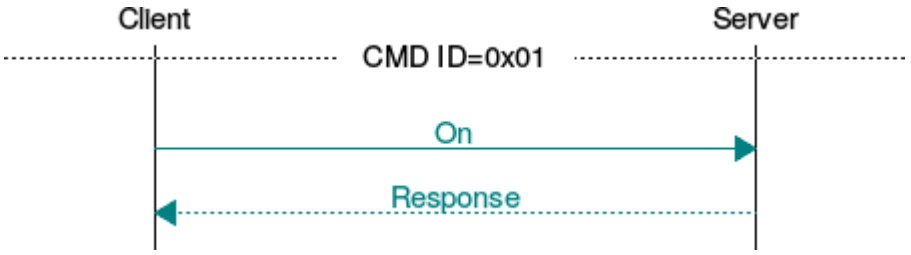

Figure 4 - On-Off Interface, *On* Command

This command, sent to a server implementation of the On-Off interface, turns some device feature on.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.2.4.2 Off

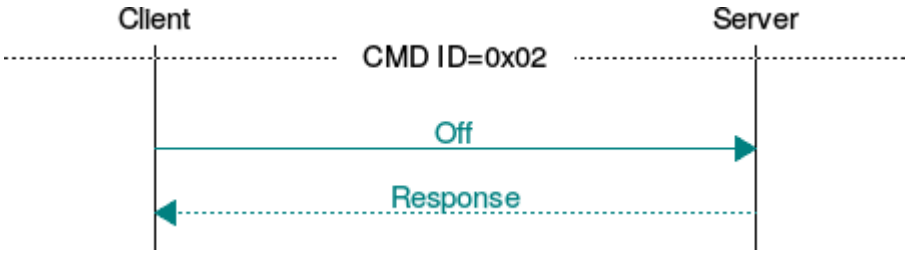

Figure 5 - On-Off Interface, *Off* Command

This command, sent to a server implementation of the On-Off interface, turns some device feature off.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.2.4.3 Toggle

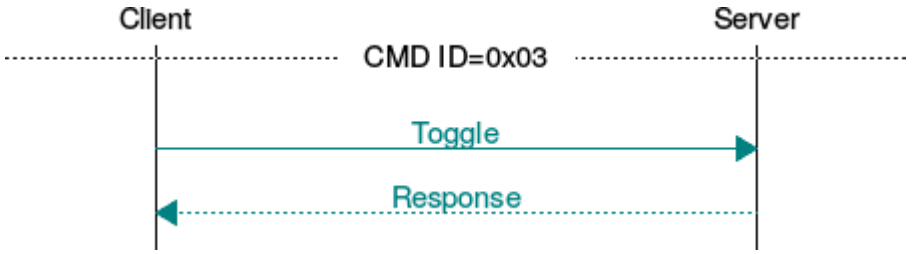

Figure 6 - On-Off Interface, *Toggle* Command

This command, sent to a server implementation of the On-Off interface, toggles the state of some device feature. If the feature was set to On, it will be turned Off and vice-versa.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.3 Level Control Interface

This interface enables a device to be controlled (server role) or to control (client role) some characteristic that can be set to a level. It allows, for example, changing the brightness of a light or the speed at which a window blind opens/closes, or even the height to which it opens/closes.

5.3.1 Server Attributes

Table 8 - Level Control Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Current Level	U8	0x00 - 0xFF	Read / Write	M

5.3.1.1 Current Level

Current Level attribute indicates the current value is a percentage of the maximum value allowed. The maximum value is device dependent, but *Current Level* is not. For example:

- a value of 0xFF (255) indicates 100% of maximum value.
- a value of 0x80 (128) indicates 50% of maximum value.

5.3.2 Client Attributes

None.

5.3.3 Server to Client Commands

None.

5.3.4 Client to Server Commands

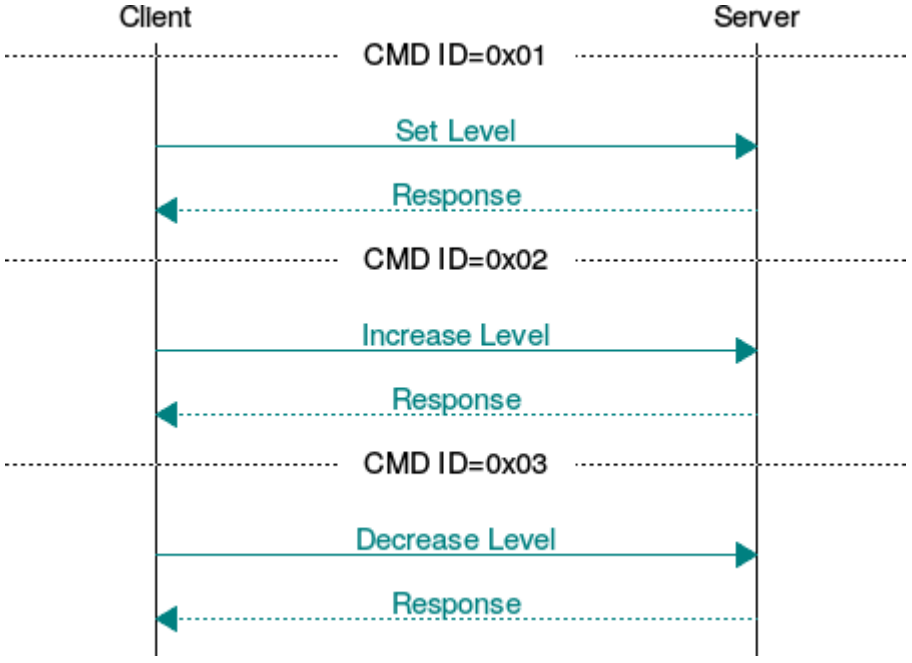

Figure 7 – Level Control Interface, Client Commands

Table 9 - Implementation status of Level Control Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Set Level	5.3.4.1	O	O	O
Increase Level	5.3.4.2	O	O	O
Decrease Level	5.3.4.3	O	O	O

5.3.4.1 Set Level

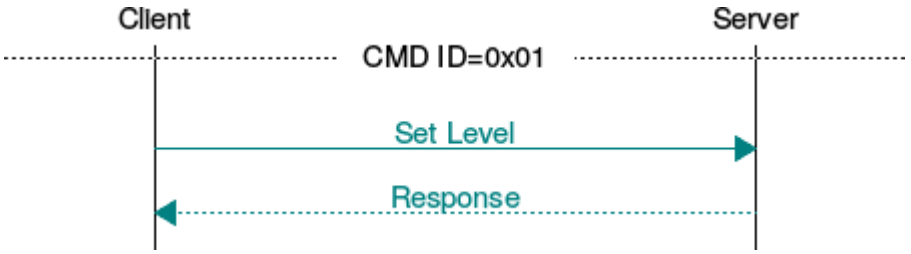

Figure 8 - Level Control Interface, *Set Level* Command

This command, sent to a server implementation of the Level Control interface, sets the current level of some device feature.

The command carries a single byte with a value with the same characteristics as defined for *Current Level* in 5.3.1.1.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.3.4.2 Increase Level

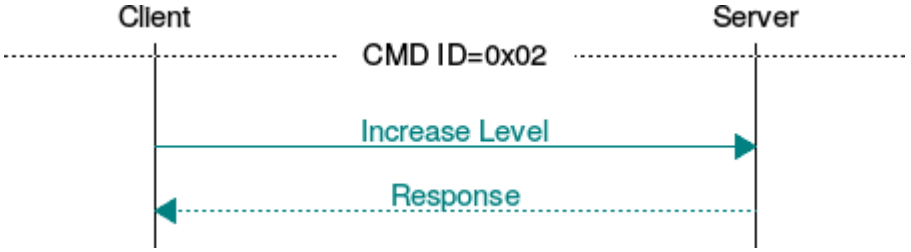

Figure 9 - Level Control Interface, *Increase Level* Command

This command, sent to a server implementation of the Level Control interface, increases the current level.

The command carries a single byte which is the percentage of change, where the value 255 corresponds to 100%.

Example:

If current level is 0x80, an increase by 10% will make the new level be at 0x99.

The increase value sent should be 26 (0x1A), as this is 10% from maximum of 255 (0xFF).

If the increase will cause an overflow the level will be 100% (0xFF) and farther increases will not change the level.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.3.4.3 Decrease Level

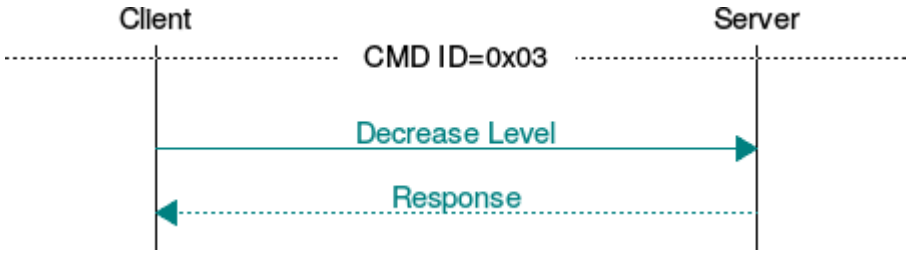

Figure 10 - Level Control Interface, *Decrease Level* Command

This command, sent to a server implementation of the Level Control interface, decreases the current level.

The command carries a single byte which is the percentage of change, where the value 255 corresponds to 100%.

Example:

If current level is 0x80, a decrease by 10% will make the new level be at 0x66.

The decrease value sent should be 26 (0x1A), as this is 10% from maximum of 255 (0xFF).

If the decrease will cause an underflow the level will be 0 (0x00) and farther decreases will not change the level.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4 Colour Control Interface

This interface is intended to provide a way to control the Colour of bulb devices.

5.4.1 Server Attributes

Table 10 – Colour Control Server: Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Supported Colour Modes	U8	Mask of following modes: • H/S; • CIE 1931 XY; • Colour Temperature;	Read Only	M
0x02	Current Colour Mode	U8	0x01 - H/S; 0x02 - CIE 1931 XY; 0x04 - Colour Temperature;	Read Only	M
0x03	Current Hue Saturation	U16+U8	0 - 359 + 0x00 - 0xFF	Read Only	O
0x04	Current XY	U16+U16	0x0000 - 0xFFFF + 0x0000 - 0xFFFF	Read Only	O
0x05	Current Colour Temperature	U16	0x0001 - 0xFFFF	Read Only	O

5.4.1.1 Supported Modes

This attribute holds the supported colour modes.

The following modes are supported:

- H/S – Hue and Saturation (This mode is optional)
- CIE 1931 XY standard – using x and y values (This mode is optional)
- Colour temperature (This mode is optional)

NOTE: At least one of the modes MUST to be supported.

Table 11- Supported Colour Modes

Mode	Mode Value Mask	M/O
H/S	0x01	O
CIE 1931	0x02	O
Colour Temperature	0x04	O

5.4.1.2 Current Colour Mode

This attribute indicates which mode the implementing unit is currently operating on.

Depending on the active mode, the colour value is given by a different attribute, as indicated in the table below.

Table 12 - Current Colour Mode

Mode	Mode Value Mask	Colour Attribute
H/S	0x01	Attribute ID 0x03 – current Hue and Saturation
CIE 1931	0x02	Attribute ID 0x04 – current XY
Colour Temperature	0x04	Attribute ID 0x05 – current colour temperature

5.4.1.3 Current Hue and Saturation

The type of the attribute is U16+U8

- U16 holds the hue – values are from 0 to 359 degrees
- U8 holds the saturation – values are 0x00 to 0xFF where 0xFF is fully saturated

This attribute MUST be implemented if HS mode is supported.

5.4.1.4 Current XY

The type of the attribute is U16+U16:

- First U16 holds the current X – values are from 0 to 65535 where x=Current X/65535
- Second U16 holds the current Y – values are from 0 to 65535 where y=Current Y/65535

This attribute MUST be implemented if CIE 1931 mode is supported.

5.4.1.5 Colour Temperature

The type of the attribute is U16.

The units used are Mired (Temperature in Kelvin = 1,000,000/Mired value)

The valid Range is 1– 65535:

- The value 1 corresponds to 1,000,000 Kelvins.
- The value 65535 corresponds to ~15.26 Kelvins

This attribute MUST be implemented if Colour Temperature mode is supported.

5.4.2 Client Attributes

None.

5.4.3 Server to Client Commands

None.

5.4.4 Client to Server Commands

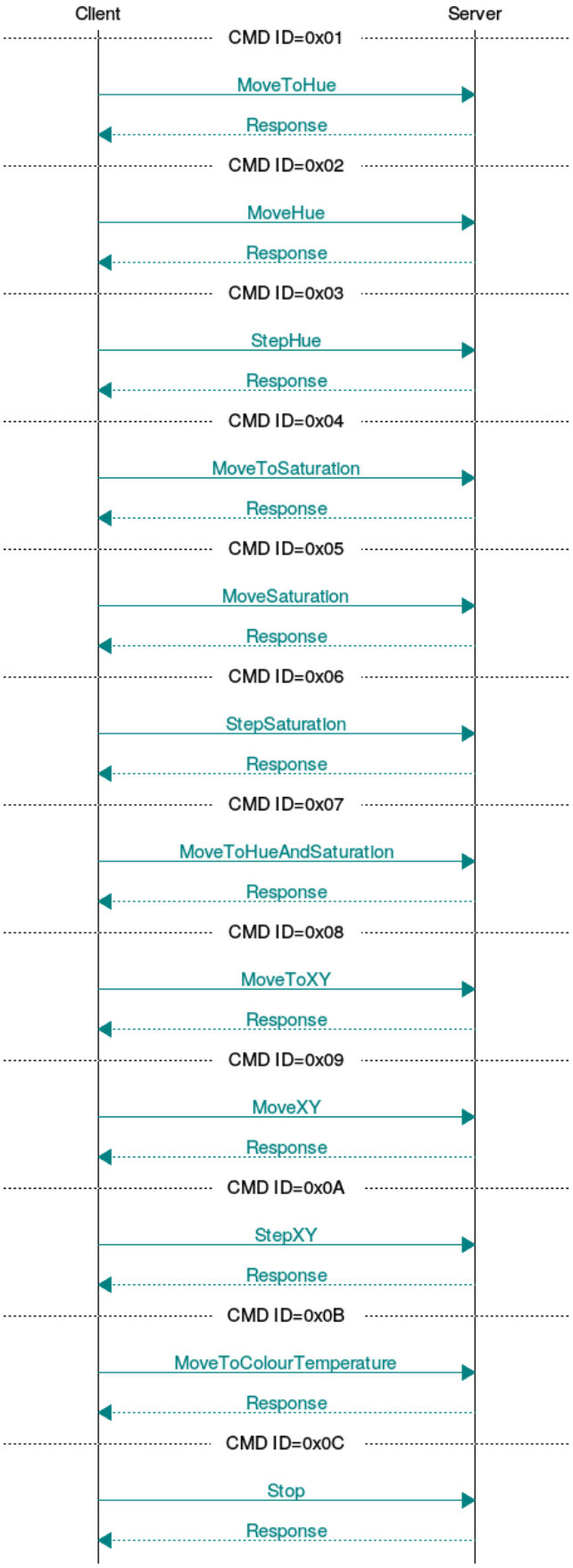

Figure 11 – Colour Control Interface, Client Commands

Table 13 - Implementation status of Colour Control Interface Client commands..

Command ID	Command Name	Reference	Client Role	Server Role	Response
0x01	Move To Hue	5.4.4.1	M[1]	M[1]	O
0x02	Move Hue	5.4.4.2	M[1]	M[1]	O
0x03	Step Hue	5.4.4.3	M[1]	M[1]	O
0x04	Move To Saturation	5.4.4.4	M[1]	M[1]	O
0x05	Move Saturation	5.4.4.5	M[1]	M[1]	O
0x06	*Step Saturation*	5.4.4.6	M[1]	M[1]	O
0x07	Move To Hue And Saturation	5.4.4.7	M[1]	M[1]	O
0x08	Move To XY	5.4.4.8	M[2]	M[2]	O
0x09	Move XY	5.4.4.9	M[2]	M[2]	O
0x0A	Step XY	5.4.4.10	M[2]	M[2]	O
0x0B	Move To Colour Temperature	5.4.4.11	M[3]	M[3]	O
0x0C	Stop	5.4.4.12	O	M	O

1 - These commands are only mandatory when the H/S mode is supported.

2 - These commands are only mandatory when the CIE 1931 mode is supported.

3 - These commands are only mandatory when the Colour Temperature mode is supported.

5.4.4.1 Move To Hue

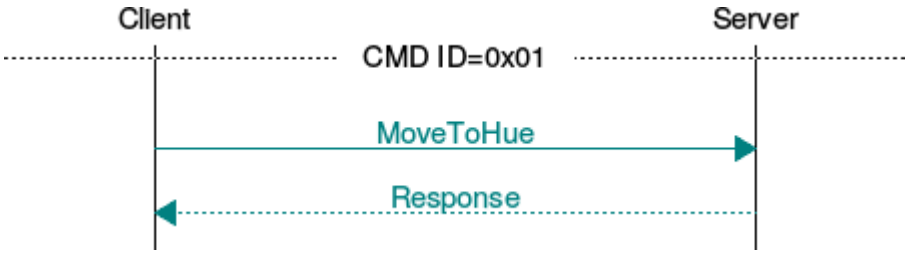

Figure 12 – Colour Control, *Move To Hue* Command.

This command shall move the colour from its current Hue to the specified Hue.

The time taken to move to the new hue shall be equal to the Transition time field. A Transition time of 0 means an immediate change to the new Hue, in this case the direction value is not applicable and can be ignored.

The current colour mode should be changed to H/S.

Table 14 - Data in the payload of a *Move To Hue* command

Field Name	Field Description	Type	Value	M/O
Hue	The value of new Hue	U16	0 - 359	M
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down 0x03 - Shortest distance 0x04 - Longest distance	M
Transition Time	Time of transition in units of 100msec	U16	0x0000 - 0xFFFF	M

Table 15 - Data ordering of the payload of a *Move To Hue* command.

8	7	6	5	4	3	2	1	Octet
Hue (MSB)								**1**
Hue(LSB)								**2**
Direction								**3**
Transition Time(MSB)								**4**
Transition Time(LSB)								**5**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.2 Move Hue

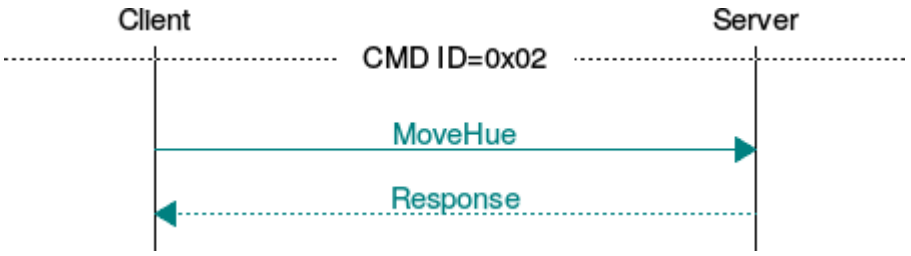

Figure 13 – Colour Control Interface, *Move Hue* Command.

This command shall cause the bulb to move from its current Hue in the rate specified in the rate field and in the direction specified in the direction field.

The movement of the Hue can be stopped by calling the Stop command.

The current colour mode should be changed to H/S.

Table 16 - Data in the payload of a *Move Hue* command

Field Name	Field Description	Type	Value	M/O
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down	M
Rate	The rate of change in degrees per seconds	U16	0 - 359	M

Table 17 - Data ordering of the payload of a *Move Hue* command.

8	7	6	5	4	3	2	1	Octet
Direction								1
Rate(MSB)								2
Rate(LSB)								3

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.3 Step Hue

Figure 14 – Colour Control Interface, *Step Hue* Command.

This command shall move the colour from its current Hue by one step. The step size is specified in the Step Size field.

The time taken to move this step shall be equal to the Transition time field. A Transition time of 0 means an immediate change of the step.

The current colour mode should be changed to H/S.

Table 18 - Data in the payload of a *Step Hue* command

Field Name	Field Description	Type	Value	M/O
Step Size	Step size in degrees	U8	0x00 - 0xFF	M
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down	M
Transition Time	Time to perform a single step. Units of 100msec	U8	0x00 - 0xFF	M

Table 19 - Data ordering of the payload of a *Step Hue* command.

8	7	6	5	4	3	2	1	Octet
Step Size								1
Direction								2
Transition Time								3

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.4 Move To Saturation

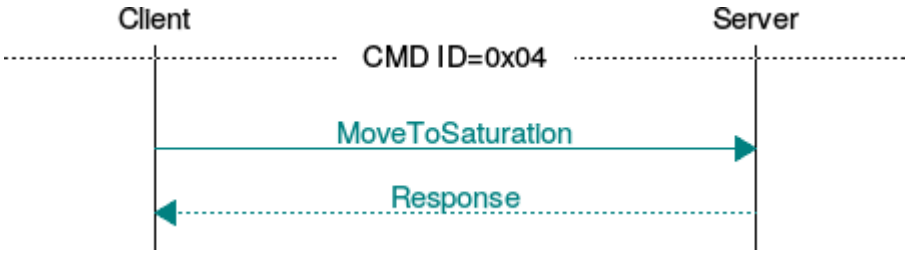

Figure 15 – Colour Control Interface, *Move To Saturation* Command.

This command shall move the colour from its current Saturation to the specified Saturation.

The time taken to move to the new saturation shall be equal to the Transition time field. A Transition time of 0 means an immediate change to the new Saturation, in this case the direction value is not applicable and should be ignored.

The current colour mode should be changed to H/S.

Table 20 - Data in the payload of a *Move To Saturation* command

Field Name	Field Description	Type	Value	M/O
Saturation	The value of new Saturation	U8	0x00 – 0xFF	M
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down	M
Transition Time	Time of transition in units of 100msec	U16	0x0000 – 0xFFFF	M

Table 21 - Data ordering of the payload of a *Move To Saturation* command.

8	7	6	5	4	3	2	1	Octet
Saturation								**1**
Direction								**2**
Transition Time(MSB)								**3**
Transition Time(LSB)								**4**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.5 Move Saturation

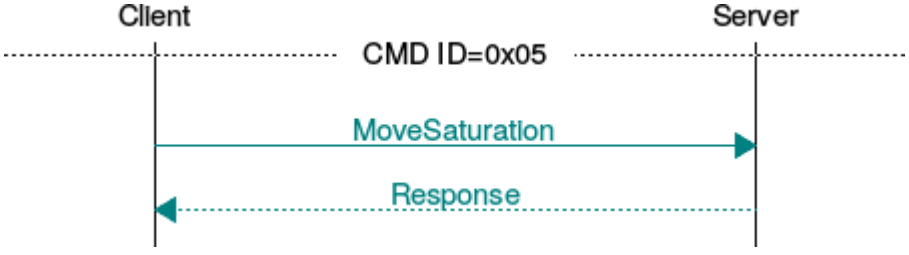

Figure 16 – Colour Control Interface, *Move Saturation* Command.

This command shall cause the colour to move from its current Saturation in the rate specified in the rate field and in the direction specified in the direction field.

The movement of the Saturation can be stopped by calling the Stop command.

The current colour mode should be changed to H/S.

Table 22 - Data in the payload of a *Move* Saturation command

Field Name	Field Description	Type	Value	M/O
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down	M
Rate	The rate of change in steps per seconds	U8	0x00 - 0xFF	M

Table 23 - Data ordering of the payload of a *Move* Saturation command.

8	7	6	5	4	3	2	1	Octet
Direction								**1**
Rate								**2**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2]

5.4.4.6 Step Saturation

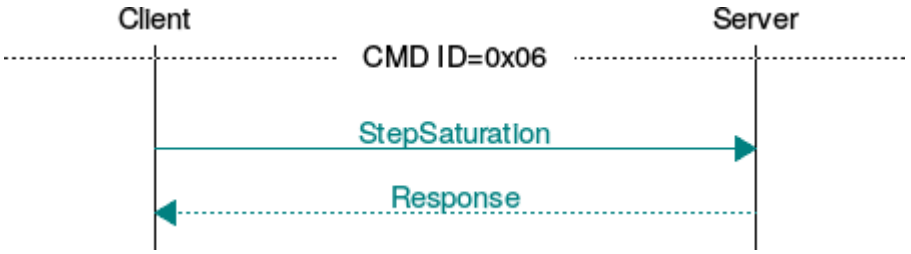

Figure 17 – Colour Control Interface, *Step Saturation* Command.

This command shall move the bulb from its current Saturation by one step.

The step size is specified in the Step Size filed.

The time taken to move this step shall be equal to the Transition time field. A Transition time of 0 means an immediate change of the step.

The current colour mode should be changed to H/S.

Table 24 - Data in the payload of a *Step Saturation* command

Field Name	Field Description	Type	Value	M/O
Step Size	Step size in units	U8	0x00 - 0xFF	M
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down	M
Transition Time	Time to perform a single step. Units of 100msec	U8	0x00 - 0xFF	M

Table 25 - Data ordering of the payload of a *Step Saturation* command.

8	7	6	5	4	3	2	1	Octet
Step Size								**1**
Direction								**2**
Transition Time								**3**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.7 Move To Hue And Saturation

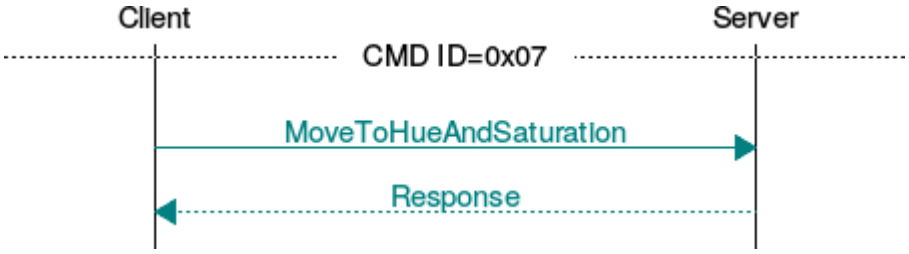

Figure 18 – Colour Control Interface, *Move To Hue And Saturation* Command.

This command shall move the colour from its current Hue and Saturation to the specified Hue and Saturation.

The time taken to move to the new hue and saturation shall be equal to the Transition time field. A Transition time of 0 means an immediate change to the new Hue and Saturation, in this case the direction value is not applicable and should be ignored.

The current colour mode should be changed to H/S.

Table 26 - Data in the payload of a *Move To Hue And Saturation* command

Field Name	Field Description	Type	Value	M/O
Hue	The value of new Hue	U16	0 - 359	M
Saturation	The value of new Saturation	U8	0x00 - 0xFF	M
Direction	Direction of movement	U8	0x01 - Up 0x02 - Down 0x03 - Shortest distance 0x04 - Longest distance	M
Transition Time	Time of transition in units of 100msec	U16	0x0000 - 0xFFFF	M

Table 27 - Data ordering of the payload of a *Move To Hue And Saturation* command.

8	7	6	5	4	3	2	1	Octet
Hue (MSB)								1
Hue(LSB)								2
Saturation								3
Direction								4
Transition Time(MSB)								5
Transition Time(LSB)								6

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.8 Move To XY

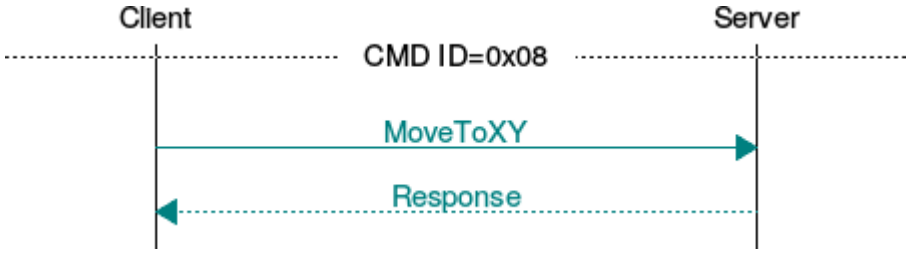

Figure 19 – Colour Control Interface, *Move To XY* Command.

This command shall move the colour from its current XY to the specified XY.

The time taken to move to the new XY shall be equal to the Transition time field. A Transition time of 0 means an immediate change to the new XY.

The direction is straight line from current XY to new XY.

The current colour mode should be changed to XY.

Table 28 - Data in the payload of a *Move To XY* command

Field Name	Field Description	Type	Value	M/O
XY	The value of new XY	U16+U16	0x0000 - 0xFFFF (X) + 0x0000 - 0xFFFF (Y)	M
Transition Time	Time of transition in units of 100msec	U16	0x0000 - 0xFFFF	M

Table 29 - Data ordering of the payload of a *Move To XY* command.

8	7	6	5	4	3	2	1	Octet
XY (MSB)								1
XY								2
XY								3
XY(LSB)								4
Transition Time(MSB)								5
Transition Time(LSB)								6

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.9 Move XY

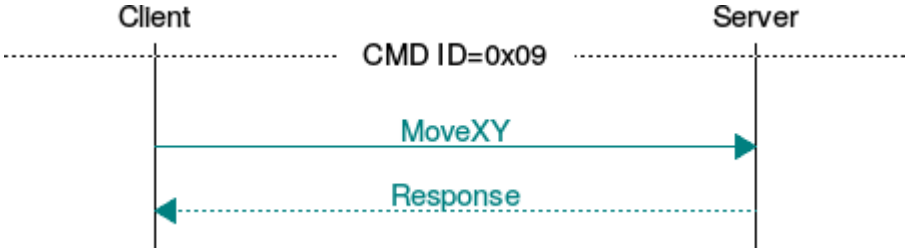

Figure 20 – Colour Control Interface, *Move XY* Command.

This command shall cause the colour to move from its current XY in the rate specified in the rate field.

The movement of the XY can be stopped by calling the Stop command.

The current colour mode should be changed to XY.

Table 30 - Data in the payload of a *Move XY* command

Field Name	Field Description	Type	Value	M/O
Rate of X	The rate of change in units per seconds	S16	(-32,768) – (32,767)	M
Rate of Y	The rate of change in units per seconds	S16	(-32,768) – (32,767)	M

Table 31 - Data ordering of the payload of a *Move XY* command.

8	7	6	5	4	3	2	1	Octet
Rate of X(MSB)								**1**
Rate of X (LSB)								**2**
Rate of Y(MSB)								**3**
Rate of Y (LSB)								**4**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.10 Step XY

Figure 21 – Colour Control Interface, *Step XY* Command.

This command shall move the bulb from its current XY by one step.

The step size of X and Y are specified in the Step Size fields.

The time taken to move this step shall be equal to the Transition time field.

A Transition time of 0 means an immediate change of the step.

The current colour mode should be changed to XY.

Table 32 - Data in the payload of a *Step XY* command

Field Name	Field Description	Type	Value	M/O
X Step Size	Step size of X in units	S16	(-32,768) – (32,767)	M
Y Step Size	Step size of Y in units	S16	(-32,768) – (32,767)	M
Transition Time	Time to perform a single step. Units of 100msec	U8	0x00 – 0xFF	M

Table 33 - Data ordering of the payload of a *Step XY* command.

8	7	6	5	4	3	2	1	Octet
X Step Size (MSB)								**1**
X Step Size (LSB)								**2**
Y Step Size (MSB)								**3**
Y Step Size (LSB)								**4**
Transition Time								**5**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.11 Move To Colour Temperature

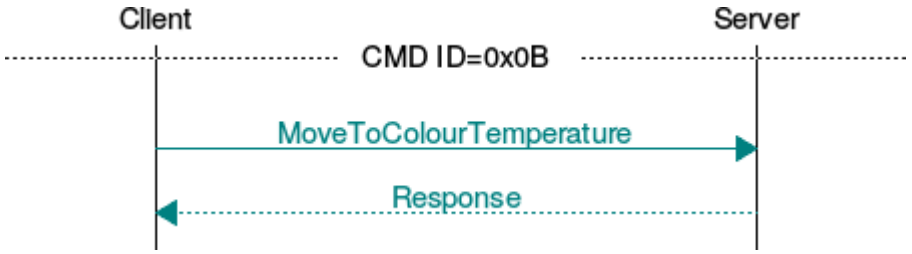

Figure 22 – Colour Control Interface, *Move To Colour Temperature* Command.

This command shall move the colour from its current colour temperature to the specified colour temperature.

The time taken to move to the colour temperature shall be equal to the Transition time field.

A Transition time of 0 means an immediate change to the new colour temperature, in this case the direction value is not applicable and should be ignored.

The current colour mode should be changed to Colour Temperature.

Table 34 - Data in the payload of a *Move To Colour Temperature* command

Field Name	Field Description	Type	Value	M/O
Colour Temperature	The value of new Colour Temperature (unit is Mired)	U16	0x0001 – 0xFFFF	M
Transition Time	Time of transition in units of 100msec	U16	0x0000 – 0xFFFF	M

Table 35 - Data ordering of the payload of a *Move To Colour Temperature* command.

8	7	6	5	4	3	2	1	Octet
Colour Temperature (MSB)								**1**
Colour Temperature (LSB)								**2**
Transition Time(MSB)								**3**
Transition Time(LSB)								**4**

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.4.12 Stop

Figure 23 – Colour Control Interface, Stop Command.

This command shall stop any prior Move commands (Hue, Saturation, XY).

This command doesn't have any payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.4.5 Attribute reporting

The attribute reporting interface should be implemented to report the changes of current XY, current HS and current Colour temperature.

5.5 Simple Keypad Interface

This interface is intended to provide a way for the device to convey the key that was pressed.

The UTF-32 encoding is used to provide universal character encoding, e.g. "*Home*" is 0x00002302, "*Open Lock*" is 0x0001F513, "*Lock*" is 0x0001F512, etc.

When selecting a character code that a particular key sends, the following procedure is recommended. Symbols and pictographs should be preferred to language specific ideographs. When multiple symbols/pictographs can be associated with the key is meant to convey then the lower character value should be used.

5.5.1 Server Attributes

None.

5.5.2 Client Attributes

None.

5.5.3 Server to Client Commands

This interface defines the following client to server commands.

- Key Pressed

Table 36 – Simple Keypad commands.

Command ID	Command Name	Reference	Client Role	Server Role	Response
0x01	KeyPressed	5.5.3.1	M	M	O

5.5.3.1 KeyPressed

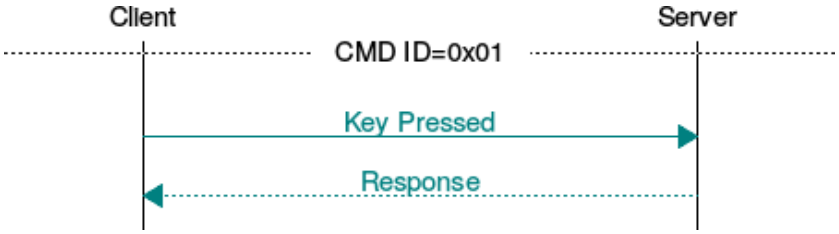

Figure 24 – Simple Keypad. KeyPressed Command.

This command should be sent from the device when one of the keys has been pressed. The command will pass the key id as a parameter. Payload is UTF32 representation of the Key that was pressed

Table 37 - Data in the payload of a KeyPressed command

Field Name	Field Description	Type	Value	M/O
Key Id	The id of the key	U32	0x00000000-0xFFFFFFFF	M

Table 38 - Data ordering of the payload of a KeyPressed command.

8	7	6	5	4	3	2	1	Octet
Key Id (MSB)								1
Key Id								2
Key Id								3
Key Id (LSB)								4

5.6 Simple Power Metering Interface

This interface enables a device to realize measurements of electric quantities that are made available for other devices to read. Having several optional attributes makes this interface able to cover a wide range of applications, from the basic outlet that reports energy consumption to full-featured electric meters.

This interface also offers flexibility regarding measurement precision. Several measurement values are preceded by a byte (Table 39) indicating the metric prefix that affects its basic unit of measurement. Time-related measurements are also preceded by a byte (Table 40) that indicates the original source from which time is measured/referenced.

Table 39 - Measurements Precision Codes and Meaning

Precision Code	Designation	Decimal Multiplier
0x00	-	10^{0}
0x10	milli	10^{-3}
0x11	micro	10^{-6}
0x12	nano	10^{-9}
0x13	pico	10^{-12}
0x20	kilo	10^{3}
0x21	mega	10^{6}
0x22	giga	10^{9}
0x23	tera	10^{12}

Table 40 - Time Measurement Codes and Meaning

Time Code	Designation	Description
0x00	Uptime	Indicates time is device referenced, it started from 0 (zero) when the device was powered up.
0x01	UTC	Indicates time is UTC referenced. Usually implies Time Service is used (see HF-Service document, [REF 3] for more details about this service).

5.6.1 Server Attributes

Table 41 - Simple Power Metering Server: Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Energy	U8+U32	See Table 39 + 0x00000000 - 0xFFFFFFFF	Read Only	M*
0x02	Energy at Last Reset	U8+U32	See Table 39 + 0x00000000 - 0xFFFFFFFF	Read Only	M*
0x03	Time at Last Reset	U8+U32	See Table 40 + 0x00000000 - 0xFFFFFFFF	Read Only	M*
0x04	Instantaneous Power	U8+U32	See Table 39 + 0x00000000 - 0xFFFFFFFF	Read Only	O
0x05	Average Power	U8+U32	See Table 39 + 0x00000000 - 0xFFFFFFFF	Read Only	O
0x06	Average Power Interval	U16	0x0000 - 0xFFFF	Read / Write	O
0x07	Voltage	U8+U32	See Table 39 + 0x00000000 - 0xFFFFFFFF	Read Only	O
0x08	Current	U8+U32	See Table 39 + 0x00000000 - 0xFFFFFFFF	Read Only	O
0x09	Frequency	U8+U32	See Table 39 + 0x0000 - 0xFFFF	Read Only	O
0x0A	Power Factor	U8	0x00 - 0xFF	Read Only	O
0x0B	Report Interval	U16	0x0000 - 0xFFFF	Read / Write	M**

* These attributes are only mandatory when the *Measurement Reset* command is implemented.

** This attribute is only mandatory when the *Report* command is implemented.

5.6.1.1 Energy

Energy attribute stores energy consumption from device power up or from a measurement reset (see *Measurement Reset* command in 5.6.4.1). This attribute is only mandatory if the referred command is implemented.

The stored value has the basic unit of Watts/Hour that may be affected by a precision code from Table 39.

5.6.1.2 Energy at Last Reset

Energy at Last Reset attribute stores the *Energy* attribute value at the instant a measurement reset occurred (see *Measurement Reset* command in 5.6.4.1). This attribute is only mandatory if the referred command is implemented.

The stored value has the basic unit of Watts/Hour that may be affected by a precision code from Table 39.

5.6.1.3 Time at Last Reset

Time at Last Reset attribute stores the time value (from device uptime or UTC) at the instant a measurement reset occurred (see *Measurement Reset* command in 5.6.4.1). This attribute is only mandatory if the referred command is implemented.

The stored value has the basic unit of Seconds with the first byte indicating its reference according to Table 40.

5.6.1.4 Instantaneous Power

Instantaneous Power attribute allows reading the presently instantaneous power value.

The stored value has the basic unit of Watts that may be affected by a precision code from Table 39.

5.6.1.5 Average Power

Average Power attribute stores the power measured over a period of time specified by *Average Power Interval*.

The stored value has the basic unit of Watts that may be affected by a precision code from Table 39.

5.6.1.6 Average Power Interval

Average Power Interval attribute specifies the time period over which power should be averaged and stored into *Average Power*.

The stored value has the basic unit of Seconds.

5.6.1.7 Voltage

Voltage attribute allows reading the presently instantaneous voltage value.

The stored value has the basic unit of Volts that may be affected by a precision code from Table 39.

5.6.1.8 Current

Current attribute allows reading the presently instantaneous current value.

The stored value has the basic unit of Amperes that may be affected by a precision code from Table 39.

5.6.1.9 Frequency

Frequency attribute allows reading the presently instantaneous frequency value.

The stored value has the basic unit of Hertz that may be affected by a precision code from Table 39.

5.6.1.10 Power Factor

Power Factor attribute stores the ratio between real and apparent power.

The stored value is a dimensionless fraction with minimum value 0 (zero - represented by 0x00) and maximum 1 (one - represented by 0xFF). Therefore, each stored value must be multiplied by the fraction – 1/255 – to obtain the real value.

5.6.1.11 Report Interval

Report Interval attribute stores the periodic time interval, in seconds, at which the *Report* command (5.6.3.1) should be sent. If this attribute is set to 0 (zero) the command will never be sent. This attribute is only mandatory if the referred command is implemented.

5.6.2 Client Attributes

None.

5.6.3 Server to Client Commands

Table 42 - Implementation status of Simple Power Metering Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Report	5.6.3.1	O	O	N/A

5.6.3.1 Report

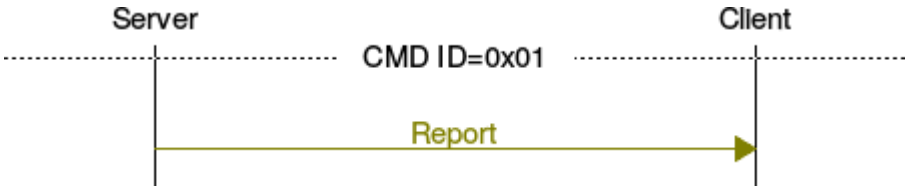

Figure 25 - Simple Power Metering Interface, *Report* Command

This optional command, sent to a client implementation of the Simple Power Metering interface, will send the value of all implemented attributes with the periodicity defined by *Report Interval* attribute.

If this command is implemented then the *Report Interval* attribute must also be.

This command must provide the information described in Table 43, organized according to Table 44.

Table 43 - Data in Payload of a *Report* Command

Field Name	Field Description	Type	Value	M/O
Number of Attributes	Number of attributes, present in the command, for which its value is sent.	U8	0x00 - 0xFF	M
Attribute ID	Identifier of an attribute whose value is sent.	U8	0x00 - 0xFF	M
Attribute Value	Value currently stored in the specified attribute.	-	-	M

Table 44 - Data Ordering of Payload of a *Report* Command

8	7	6	5	4	3	2	1	Octet
Number of Attributes								**1**
Attribute ID i								**2**
Attribute Value (MSB)								**3**
…								⋮
Attribute Value (LSB)								**4 + n**
…								⋮
Attribute ID i+n								**5 + n**
Attribute Value (MSB)								**6 + n**
…								⋮
Attribute Value (LSB)								**7 + n**

5.6.4 Client to Server Commands

Table 45 - Implementation status of Simple Power Metering Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Measurement Reset	5.6.4.1	O	O	O

5.6.4.1 Measurement Reset

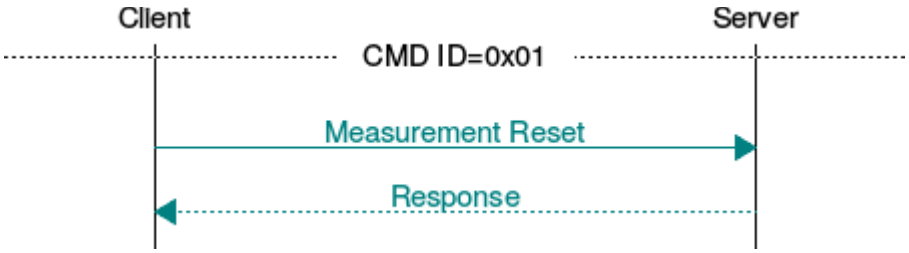

Figure 26 - Simple Power Metering Interface, *Measurement Reset* Command

This optional command sent to a server implementation of the Simple Power Metering interface, performs the following operations:

- Copy *Energy* attribute present value into *Energy at Last Reset* attribute;
- Set *Energy* attribute to 0 (zero);
- Store device time into *Time at Last Reset* attribute.

Due to the nature of these operations, if this command is implemented then the attributes *Energy at Last Reset* and *Time at Last Reset* must also be.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.7 Simple Temperature Interface

This interface enables a device to be able to provide temperature readings.

5.7.1 Server Attributes

Table 46 – Simple Temperature Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Measured Temperature	S16	-32768 to +32767	Read Only	M
0x02	Minimum Measureable Temperature	S16	-32768 to +32767	Read Only	M
0x03	Maximum Measureable Temperature	S16	-32768 to +32767	Read Only	M
0x04	Tolerance	U16	0x0000 - 0xFFFF	Read Only	M

5.7.1.1 Measured Temperature

This attribute holds the current measured temperature, in one hundredth (1/100) of Celsius degrees (ºC).

5.7.1.2 Minimum Measureable Temperature

This attribute holds the minimum temperature, in one hundredth (1/100) of Celsius degrees (ºC), that can be measured by the device.

5.7.1.3 Maximum Measureable Temperature

This attribute holds the maximum temperature, in one hundredth (1/100) of Celsius degrees (ºC), that can be measured by the device.

5.7.1.4 Tolerance

This attribute holds the magnitude, in one hundredth (1/100) of Celsius degrees (ºC), of the error associated with *Measured Temperature***,** which means the actual value is between (*Measured Temperature – Tolerance*) and (*Measured Temperature + Tolerance*).

5.7.2 Client Attributes

None.

5.7.3 Server to Client Commands

None.

5.7.4 Client to Server Commands

None.

5.8 Simple Humidity Interface

This interface enables a device to be able to provide humidity readings.

5.8.1 Server Attributes

Table 47 – Simple Humidity Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Measured Humidity	U16	0x0000 - 0xFFFF	Read Only	M
0x02	Tolerance	U16	0x0000 - 0xFFFF	Read Only	M

5.8.1.1 Measured Humidity

This attribute holds the current measured Relative Humidity percentage, in one hundredth (1/100) of Relative Humidity percentage.

5.8.1.2 Tolerance

This attribute holds the magnitude, in one hundredth (1/100) of Relative Humidity percentage, of the error associated with *Measured Humidity***,** which means the actual value is between (*Measured Humidity – Tolerance*) and (*Measured Humidity + Tolerance*).

5.8.2 Client Attributes

None.

5.8.3 Server to Client Commands

None.

5.8.4 Client to Server Commands

None.

5.9 Simple Thermostat Interface

This interface enables a device to be able to control the temperature of an indoor area, either by cooling or heating it.

5.9.1 Server Attributes

Table 48 – Simple Thermostat Interface: Server attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Supported Modes	U8 (bitmask)[1]	0x01 - Heating only 0x02 - Cooling only 0x04 - Heating and Cooling 0x10 - Fan not supported 0x20 - Fan supported 0x40 - Fan supports automatic mode	Read Only	M
0x02	Operating Mode	U8	0x01 - Heating mode 0x02 - Cooling mode 0x04 - Automatic mode	Read / Write	M
0x03	Fan Mode	U8	0x10 - Fan is OFF 0x20 - Fan is ON 0x40 - Fan is in AUTO mode	Read / Write	O
0x04	Heating Mode Temperature	S16	-32768 to +32767	Read / Write	O[2]
0x05	Cooling Mode Temperature	S16	-32768 to +32767	Read / Write	O[3]
0x06	Automatic Mode Heating Temperature	S16	-32768 to +32767	Read / Write	O[4]
0x07	Automatic Mode Cooling Temperature	S16	-32768 to +32767	Read / Write	O[4]
0x08	Heating Mode Temperature Offset	S16	-32768 to +32767	Read / Write	O
0x09	Cooling Mode Temperature Offset	S16	-32768 to +32767	Read / Write	O
0x0A	Boost Duration	U8	0x00 - 0xFF	Read / Write	O[5]

5.9.1.1 Supported Modes

Supported Modes attribute indicates all the supported thermostat modes both in terms of temperature control and fan control.

5.9.1.2 Operating Mode

Operating Mode attribute is used to read and/or set the current mode of operation of the thermostat. It can take one of three values:

---

[1] The values of *Supported Modes* attribute can be combined since they are interpreted as a bitmask.

[2] This attribute must be implemented if *Heating Mode* is supported.

[3] This attribute must be implemented if *Cooling Mode* is supported.

[4] This attribute must be implemented if *Automatic Mode* is supported.

[5] This attribute must be implemented if *Boost Start* command (5.9.4.1) is implemented.

- The value 0x01 indicates it is in Heating Mode, meaning it will try to maintain the temperature equal or above the value specified in the *Heating Mode Temperature* attribute plus, if implemented, the value specified in the *Heating Mode Temperature Offset* attribute.

- The value 0x02 indicates it is in Cooling Mode, meaning it will try to maintain the temperature equal or below the value specified in the *Cooling Mode Temperature* attribute plus, if implemented, the value specified in the *Cooling Mode Temperature Offset* attribute.

- The value 0x04 indicates it is in Automatic Mode, meaning it will try maintaining the temperature between the values specified by *Automatic Mode Heating Temperature* and *Automatic Mode Cooling Temperature* attributes. To maintain the temperature range, the thermostat will start heating when temperature is below the value specified in *Automatic Mode Heating Temperature* and start cooling when temperature is above the *Automatic Mode Cooling Temperature*.

5.9.1.3 Fan Mode

Fan Mode attribute is used to read and/or set the current mode of operation of the thermostat's fan. It can take one of three values:

- The value 0x10 indicates the fan is disabled.

- The value 0x20 indicates the fan is enabled and always fanning.

- The value 0x40 indicates the fan is in automatic mode, meaning it is only enabled (and fanning) when the thermostat is either heating or cooling, otherwise the fan is disabled.

5.9.1.4 Heating Mode Temperature

Heating Mode Temperature attribute holds the threshold, in one hundredth (1/100) of Celsius degrees (ºC), below which a thermostat in Heating Mode (see 5.9.1.2) will start heating. Heating should stop when the temperature reaches this threshold or a few degrees higher to avoid oscillations.

5.9.1.5 Cooling Mode Temperature

Cooling Mode Temperature attribute holds the threshold, in one hundredth (1/100) of Celsius degrees (ºC), above which a thermostat in Cooling Mode (see 5.9.1.2) will start cooling. Cooling should stop when the temperature reaches this threshold or a few degrees lower to avoid oscillations.

5.9.1.6 Automatic Mode Heating Temperature

Automatic Mode Heating Temperature attribute holds the threshold, in one hundredth (1/100) of Celsius degrees (ºC), below which a thermostat in Automatic Mode (see 5.9.1.2) will start heating. Heating should stop when the temperature reaches this threshold or a few degrees higher to avoid oscillations.

5.9.1.7 Automatic Mode Cooling Temperature

Automatic Mode Cooling Temperature attribute holds the threshold, in one hundredth (1/100) of Celsius degrees (ºC), above which a thermostat in Automatic Mode (see 5.9.1.2) will start cooling. Cooling should stop when the temperature reaches this threshold or a few degrees lower to avoid oscillations.

5.9.1.8 Heating Mode Temperature Offset

Heating Mode Temperature Offset attribute holds a temperature offset value, in one hundredth (1/100) of Celsius degrees (ºC), that can be used to adjust the real temperature target of the thermostat without directly changing the *Heating Mode Temperature* attribute.

5.9.1.9 Cooling Mode Temperature Offset

Cooling Mode Temperature Offset attribute holds a temperature offset value, in one hundredth (1/100) of Celsius degrees (ºC), that can be used to adjust the real temperature target of the thermostat without directly changing the *Cooling Mode Temperature* attribute.

5.9.1.10 Boost Duration

Boost Duration attribute holds the duration, in minutes, of a boost and is used when the *Boost Start* command not specify a duration.

A boost is started by *Boost Start* command (5.9.4.1) and provides a means to temporarily override the normal thermostat behaviour making it heat or cool more intensely with the intention of reaching the target temperature more rapidly.

To avoid damage to the device or to people the server implementation can discard or lower the value set by a client implementation. It is therefore not guaranteed that the boost will have the duration set by a *Set Attribute Request* command (see [REF2] for details regarding attribute requests).

5.9.2 Client Attributes

None.

5.9.3 Server to Client Commands

None.

5.9.4 Client to Server Commands

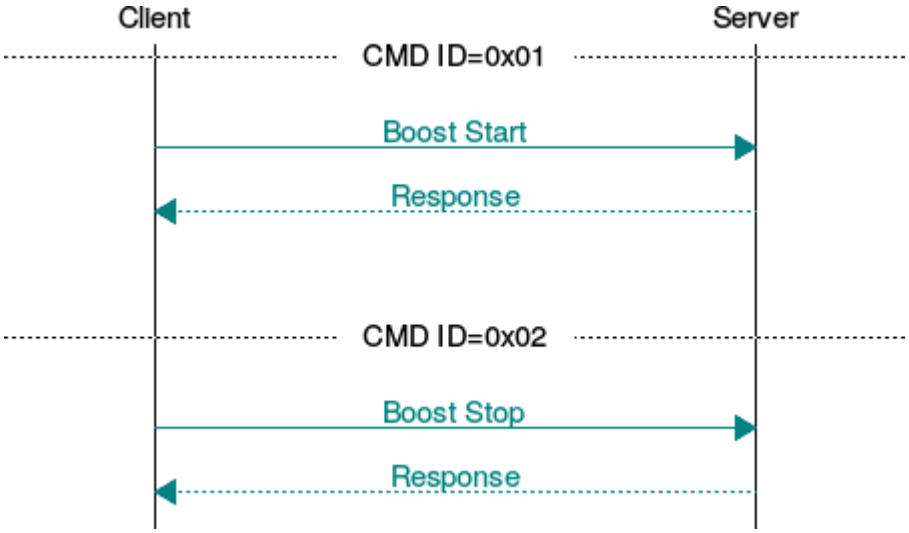

Figure 27 - Simple Thermostat Interface: Client commands.

Table 49 - Implementation status of Simple Thermostat Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Boost Start	5.9.4.1	O	O	O
Boost Stop	5.9.4.2	O	O[6]	O

[6] *Boost Stop* command must be implemented as a server role if the *Boost Start* command is implemented.

5.9.4.1 Boost Start

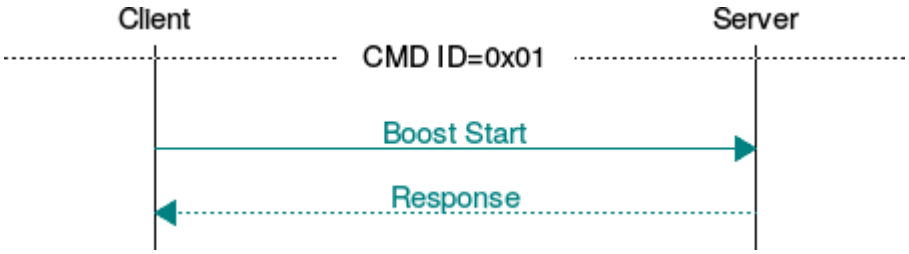

Figure 28 - Simple Thermostat Interface: *Boost Start* command.

This command allows a device to make a thermostat temporarily override its normal heating/cooling behaviour, effectively making it heat or cool more intensely with the intention of reaching a target temperature more rapidly. This procedure is called a boost and has always a duration associated with it; such duration is either specified in the command's payload or the server uses the value specified in the *Boost Duration* attribute (5.9.1.10).

This command may be sent without any payload or provide the optional single byte with the information described in Table 50.

The optional response, if requested, is a single byte – Response Code – with one of the values described in Table 51.

Table 50 - Data in the payload of a *Boost Start* command.

Field Name	Field Description	Type	Value	M/O
Duration	Number of minutes to sustain the boost mode for. To avoid damage to the device or to people the server implementation can discard or lower this value. It is therefore not guaranteed that the boost will have the specified duration.	U8	0x00 - 0xFF	O

Table 51 - Data in the payload of a Response to a *Boost Start* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M

5.9.4.2 Boost Stop

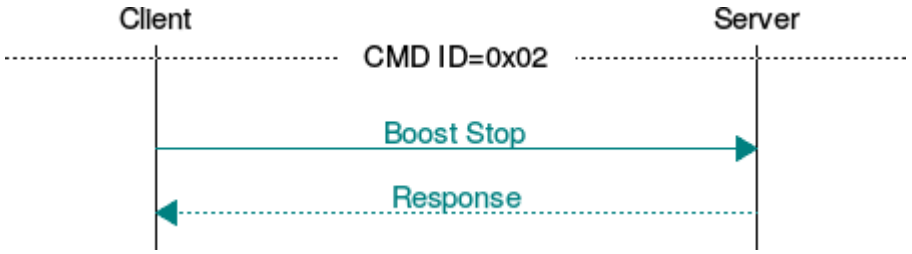

Figure 29 - Simple Thermostat Interface: *Boost Stop* command.

This command allows a device to prematurely end a boost cycle that was previously initiated by a *Boost Start* command.

This command has no payload.

The optional response, if requested, is a single byte – Response Code – with one of the values described in Table 52.

Table 52 - Data in the payload of a Response to a *Boost Stop* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x03 - Fail: Not supported 0xFF - Fail: Unknown reason	M

5.10 Simple Button Interface

This interface enables a device to be able to send button press related notifications, such as when the press occurred and the type of press, short, long, extra-long or “double click”.

A button press is defined as the downward compression of a button or surface followed by its release. By taking into account the duration of this procedure as well the duration between consecutive such procedures, several different types of button presses can be identified.

5.10.1 Server Attributes

Table 53 – Simple Button: Server attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Short Press Maximum Duration	U16	0x0000 - 0xFFFF	Read / Write	M
0x02	Extra Long Press Minimum Duration	U16	0x0000 - 0xFFFF	Read / Write	M
0x03	Double Press Gap Duration	U16	0x0064 - 0xFFFF	Read / Write	O

5.10.1.1 Short Press Maximum Duration

Short Press Maximum Duration attribute defines the maximum press duration, in milliseconds, that is still considered a short press. This value is used as a threshold to trigger a *Short Press* command (5.10.3.1) and also influences the triggering of a *Long Press* command (5.10.3.2).

If this attribute is set to the value – 0x0000 – then a *Short Press* command (5.10.3.1) is never sent.

5.10.1.2 Extra Long Press Minimum Duration

Extra Long Press Minimum Duration attribute defines the minimum press duration, in milliseconds, required for it to be considered an extra-long press. This value is used as a threshold to trigger an *Extra-Long Press* command (5.10.3.3) and also influences the triggering of a *Long Press* command (5.10.3.2).

If this attribute is set to the value – 0x0000 – then an *Extra-Long Press* command (5.10.3.3) is never sent.

If this attribute is set to a value lower than *Short Press Maximum Duration* attribute then a *Long Press* (5.10.3.2) and an Extra-*Long Press* (5.10.3.3) commands are never sent.

5.10.1.3 Double Press Gap Duration

Double Press Gap Duration attribute defines the maximum gap duration, in milliseconds, between the release of a short press (first press[7]) and the detection of a downward compression of a subsequent press (second press[8]) so it is still considered a double press. This value is used as a threshold to trigger a *Double Press* command (5.10.3.4).

This attribute is restricted to a minimum of 100ms (0x0064). The special value of 0x0000 can be used to indicate that a *Double Press* command is never sent.

5.10.2 Client Attributes

None.

[7] The first press must always be a short press (5.10.1.1).

[8] Since the decision is taken as soon as the downward compression of the second press occurs, its duration is irrelevant.

5.10.3 Server to Client Commands

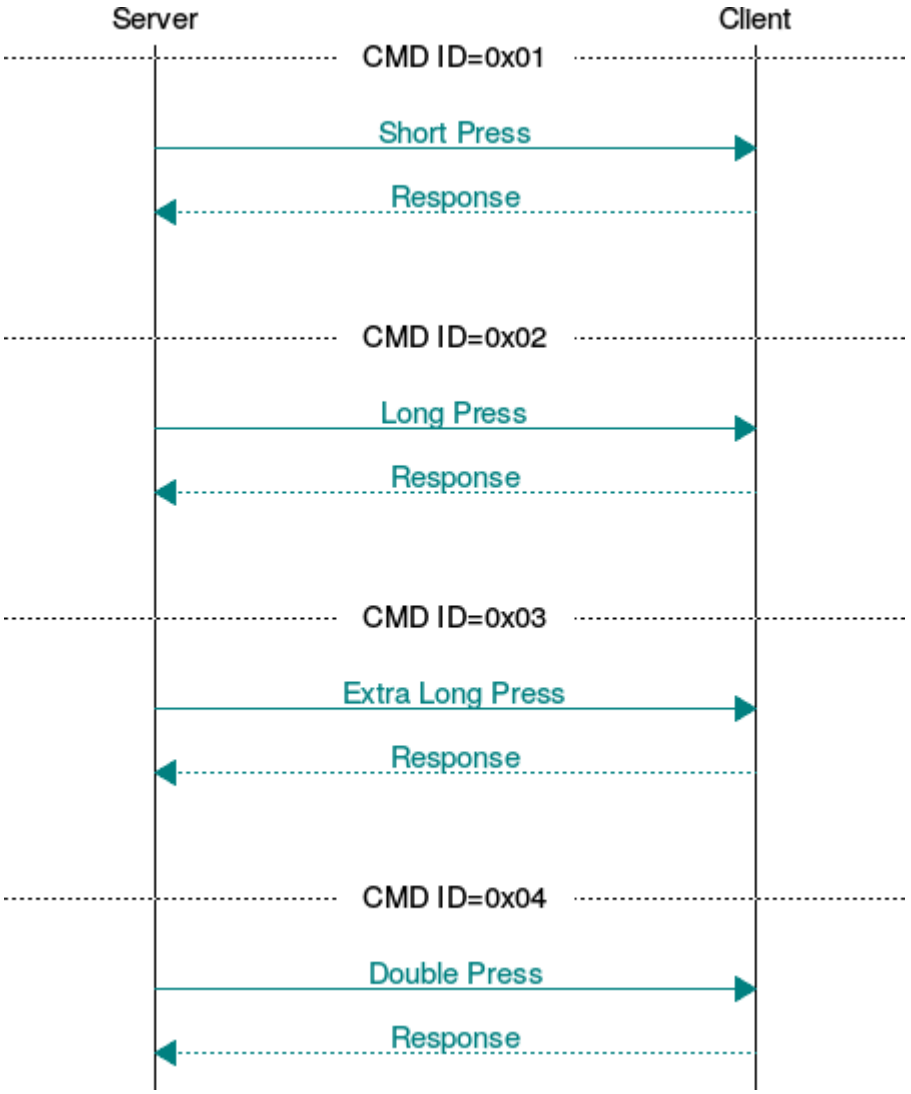

Figure 30 - Simple Button Interface: Server commands.

Table 54 - Implementation status of Simple Button Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Short Press	5.10.3.1	O	M	O
Long Press	5.10.3.2	O	M	O
Extra-Long Press	5.10.3.3	O	O	O
Double Press	5.10.3.4	O	O	O

5.10.3.1 Short Press

Figure 31 - Simple Button Interface: *Short Press* command.

This command, sent to a client implementation of Simple Button Interface, informs that a short press was detected. It must be sent anytime a press with duration less than *Short Press Maximum Duration* (5.10.1.1) is detected.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.10.3.2 Long Press

Figure 32 - Simple Button Interface: *Long Press* command.

This command, sent to a client implementation of Simple Button Interface, informs that a long press was detected. It must be sent anytime a press with duration greater than *Short Press Maximum Duration* (5.10.1.1) but less than *Extra Long Press Minimum Duration* (5.10.1.2) is detected.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.10.3.3 Extra-Long Press

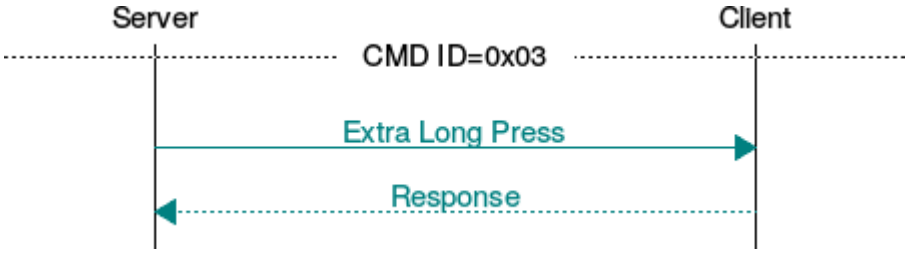

Figure 33 - Simple Button Interface: *Extra-Long Press* command.

This command, sent to a client implementation of Simple Button Interface, informs that an extra-long press was detected. It must be sent anytime a press with duration greater than *Extra Long Press Minimum Duration* (5.10.1.2) is detected.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.10.3.4 Double Press

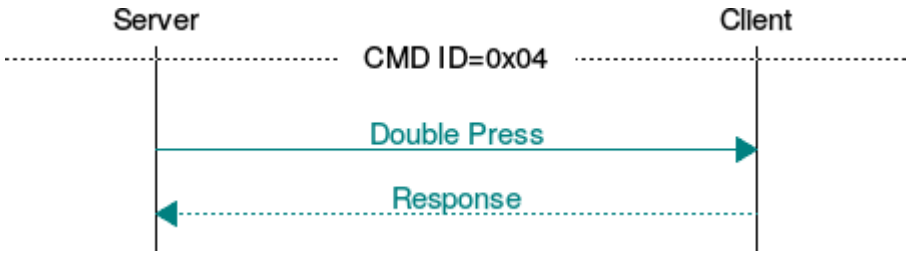

Figure 34 - Simple Button Interface: *Double Press* command.

This command, sent to a client implementation of Simple Button Interface, informs that a double press was detected. It must be sent anytime two consecutive presses are detected within the duration specified by *Double Press Gap Duration* (5.10.1.3).However, the first press must be a short press and the command is to be sent as soon as the “start” of the second press is detected as described in 5.10.1.3.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.10.4 Client to Server Commands

None.

5.11 Simple Visual Control Interface

This interface enables a device to be able to control several visual effects (blink, fade and breathe) as well as to turn OFF and ON some visual indicator (display, LED, lamp, etc).

5.11.1 Server Attributes

None.

5.11.2 Client Attributes

None.

5.11.3 Server to Client Commands

None.

5.11.4 Client to Server Commands

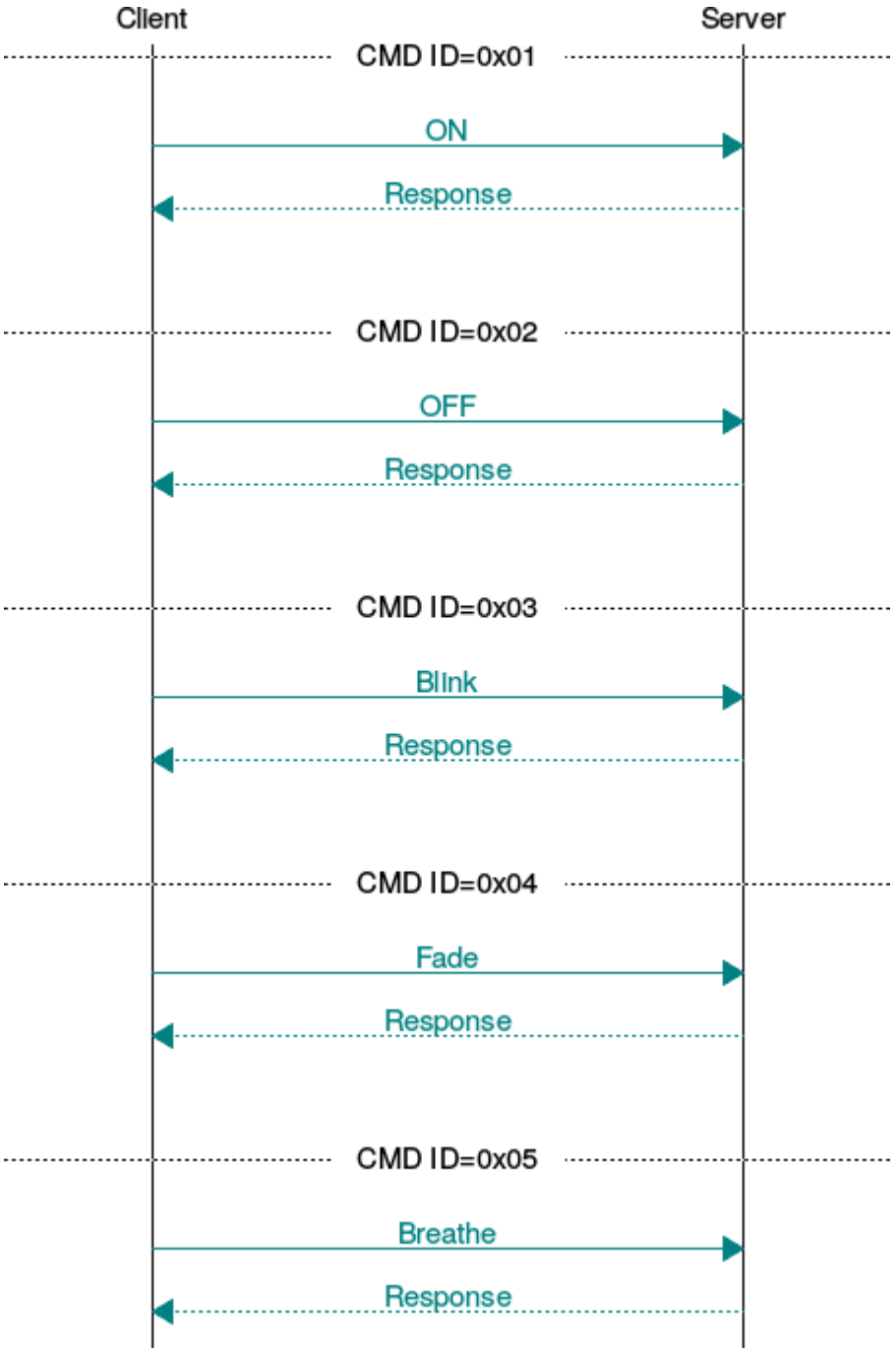

Figure 35 - Simple Visual Control: Client commands.

Table 55 - Implementation status of Simple Visual Control Client commands.

Command	Reference	Client Role	Server Role	Response
ON	5.11.4.1	O	M	O
OFF	5.11.4.2	O	M	O
Blink	5.11.4.3	O	O	O
Fade	5.11.4.4	O	O	O
Breath	5.11.4.5	O	O	O

5.11.4.1 ON

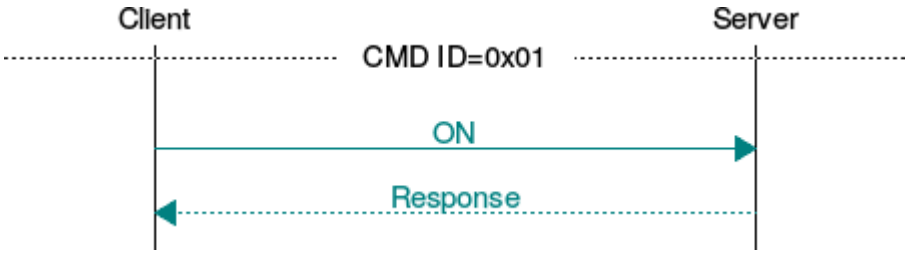

Figure 36 - Simple Visual Control: *ON* command.

This command allows a device to turn ON some visual indicator for a specified amount of time.

This command must provide the information described in Table 56, organized according to Table 57.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

Table 56 - Data in the payload of an *ON* command.

Field Name	Field Description	Type	Value	M/O
Duration	Number of milliseconds to maintain the visual indicator ON.	U16	0x0000 - 0xFFFF	M

Table 57 - Data Ordering of Payload of an *ON* command

8	7	6	5	4	3	2	1	Octet
Duration (MSB)								**1**
Duration (LSB)								**2**

5.11.4.2 OFF

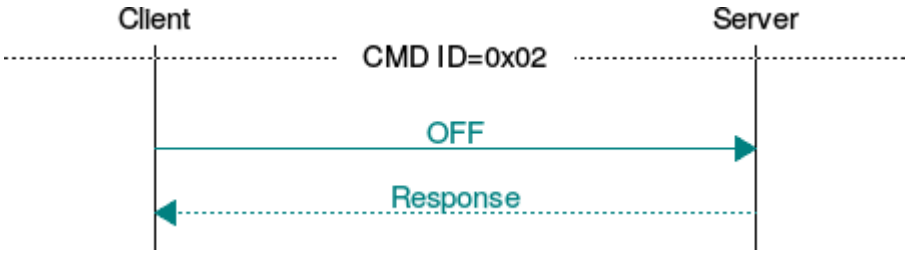

Figure 37 - Simple Visual Control: *OFF* command.

This command allows a device to turn OFF some visual indicator, prematurely ending any effect that might be active.

This command has no payload.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

5.11.4.3 Blink

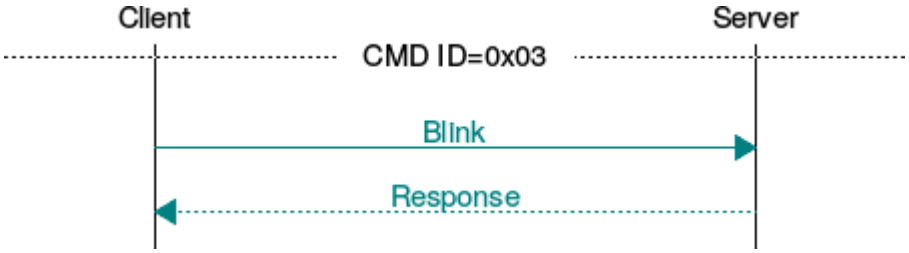

Figure 38 - Simple Visual Control: *Blink* command.

This command allows a device to make some visual indicator blink for a specified amount of time.

This command must provide the information described in Table 58, organized according to Table 59.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

Table 58 - Data in the payload of a *Blink* command.

Field Name	Field Description	Type	Value	M/O
On Duty Cycle	Number of milliseconds to maintain the visual indicator ON.	U16	0x0000 - 0xFFFF	M
Off Duty Cycle	Number of milliseconds to maintain the visual indicator OFF.	U16	0x0000 - 0xFFFF	M
Number of Cycles	Number of times to repeat the blinking cycle.	U16	0x0001 - 0xFFFF	M

Table 59 - Data Ordering of Payload of a *Blink* command

8	7	6	5	4	3	2	1	Octet
On Duty Cycle (MSB)								1
On Duty Cycle (LSB)								2
Off Duty Cycle (MSB)								3
Off Duty Cycle (LSB)								4
Number of Cycles (MSB)								5
Number of Cycles (LSB)								6

5.11.4.4 Fade

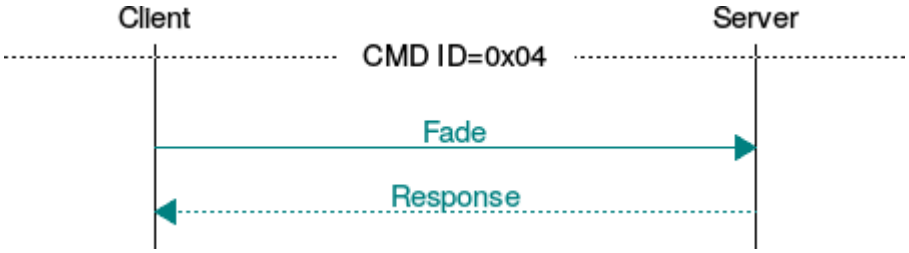

Figure 39 - Simple Visual Control: *Fade* command.

This command allows a device to make some visual indicator decrease its brightness between two specified values taking a specified amount of time.

This command must provide the information described in Table 60, organized according to Table 61.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

Table 60 - Data in the payload of a *Fade* command.

Field Name	Field Description	Type	Value	M/O
Starting Brightness	Brightness starting value expressed a percentage of the maximum value allowed.	U8	0x00 - 0xFF	M
Final Brightness	Brightness final value expressed a percentage of the maximum value allowed.	U8	0x00 - 0xFF	M
Fade Duration	Number of milliseconds to take when decreasing the brightness between *Starting Brightness* and *Final Brightness*.	U16	0x0000 - 0xFFFF	M

Table 61 - Data Ordering of Payload of a *Fade* command

8	7	6	5	4	3	2	1	Octet
Starting Brightness								1
Final Brightness								2
Fade Duration (MSB)								3
Fade Duration (LSB)								4

5.11.4.5 Breath

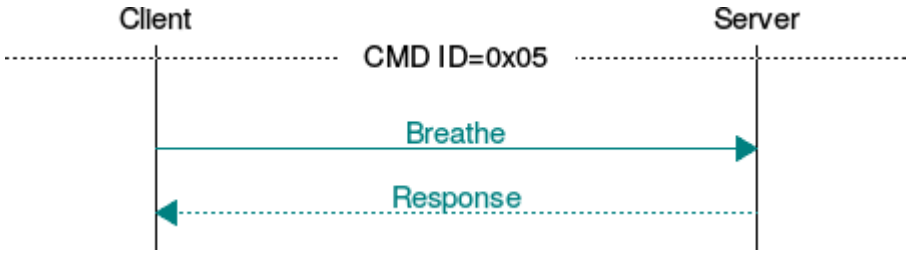

Figure 40 - Simple Visual Control: *Breath* command.

This command allows a device to make some visual indicator change its brightness between two specified values with breathing like effect (brightness is held at the extremes for some amount of time).

This command must provide the information described in Table 62, organized according to Table 63.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

Table 62 - Data in the payload of a *Breath* command.

Field Name	Field Description	Type	Value	M/O
First Brightness	Brightness first value expressed a percentage of the maximum value allowed.	U8	0x00 - 0xFF	M
First Brightness Duration	Number of milliseconds to maintain the visual indicator with *First Brightness* intensity.	U16	0x0000 - 0xFFFF	M
First to Second Fade Duration	Number of milliseconds to take when changing brightness between *First Brightness* and *Second Brightness*.	U16	0x0000 - 0xFFFF	M
Second Brightness	Brightness second value expressed a percentage of the maximum value allowed.	U8	0x00 - 0xFF	M
Second Brightness Duration	Number of milliseconds to maintain the visual indicator with *Second Brightness* intensity.	U16	0x0000 - 0xFFFF	M
Second to First Fade Duration	Number of milliseconds to take when changing brightness between *Second Brightness* and *First Brightness*.	U16	0x0000 - 0xFFFF	M
Number of Cycles	Number of times to repeat the breathing cycle.	U16	0x0001 - 0xFFFF	M

Table 63 - Data Ordering of Payload of a *Breath* command

8	7	6	5	4	3	2	1	Octet
First Brightness								1
First Brightness Duration (MSB)								2
First Brightness Duration (LSB)								3
First to Second Fade Duration (MSB)								4
First to Second Fade Duration (LSB)								5
Second Brightness								6
Second Brightness Duration (MSB)								7
Second Brightness Duration (LSB)								8
Second to First Fade Duration (MSB)								9
Second to First Fade Duration (LSB)								10
Number of Cycles (MSB)								11
Number of Cycles (LSB)								12

5.12 Air Pressure Interface

This interface enables a device to be able to provide air pressure readings.

Air pressure readings are given in hectopascals (hPa), a multiple of pascal (Pa) (1hPa = 100Pa) which is an International System (SI) unit of measurement. The average pressure at sea-level is approximately 101325 Pa or 1013.25 hPa.

5.12.1 Server Attributes

Table 64 – Simple Temperature Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Measured Air Pressure	U16	0x0000 - 0xFFFF	Read Only	M
0x02	Minimum Measureable Air Pressure	U16	0x0000 - 0xFFFF	Read Only	M
0x03	Maximum Measureable Air Pressure	U16	0x0000 - 0xFFFF	Read Only	M
0x04	Tolerance	U16	0x0000 - 0xFFFF	Read Only	M

5.12.1.1 Measured Air Pressure

This attribute holds the current measured air pressure, in hectopascals (hPa) units.

5.12.1.2 Minimum Measureable Air Pressure

This attribute holds the minimum air pressure, in hectopascals (hPa), that can be measured by the device.

5.12.1.3 Maximum Measureable Air Pressure

This attribute holds the maximum air pressure, in hectopascals (hPa), that can be measured by the device.

5.12.1.4 Tolerance

This attribute holds the magnitude, in hectopascals (hPa), of the error associated with *Measured Air Pressure***,** which means the actual value is between (*Measured Air Pressure – Tolerance*) and (*Measured Air Pressure + Tolerance*).

5.12.2 Client Attributes

None.

5.12.3 Server to Client Commands

None.

5.12.4 Client to Server Commands

None.

5.13 Simple Light Sensor Interface

The Simple Light Sensor interface enables a device reporting about Light readings.

The lux (symbol: lx) is the SI unit of illuminance and luminous emittance, measuring luminous flux per unit area. It is equal to one lumen per square meter. In photometry, this is used as a measure of the intensity, as perceived by the human eye, of light that hits or passes through a surface.

The possible values are varying from “total darkness” (zero - 0 lux) to direct sunlight (150,000 lux).

5.13.1 Server Attributes

Table 65 – Server Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Measured Lux	U32	0x00000000 - 0xFFFFFFFF	Read	M
0x02	Minimum Lux	U32	0x00000000 - 0xFFFFFFFF	Read	M
0x03	Maximum Lux	U32	0x00000000 - 0xFFFFFFFF	Read	M
0x04	Tolerance Lux	U32	0x00000000 - 0xFFFFFFFF	Read	M

5.13.1.1 Measured Lux

The *Measured Lux* attribute holds the current measured Lux, in one hundredth (1/100) of Lux units.

5.13.1.2 Minimum Lux

The *Minimum Lux* attribute holds the minimum value of *Measured Lux* that can be measured in one hundredth (1/100) of Lux units.

5.13.1.3 Maximum Lux

The *Maximum Lux* attribute holds the maximum value of *Measured Lux* that can be measured in one hundredth (1/100) of Lux units.

The *Maximum Lux* attribute value shall always be greater than *Minimum Lux* attribute value.

The *Minimum Lux* & *Maximum Lux* attributes define the range of the sensor.

5.13.1.4 Tolerance Lux

The Tolerance Lux attribute holds the magnitude, in one hundredth (1/100) of Lux units, of the error associated with *Measured Lux,* which means the actual value is between (*Measured Lux – Tolerance*) and (*Measured Lux + Tolerance*).

5.13.2 Client Attributes

None.

5.13.3 Server to Client Commands

None.

5.13.4 Client to Server Commands

None.

5.14 Custom Manufacturer Interfaces

Although HF will continue to expand existing interfaces and to define new ones, it also permits manufacturers to build their own specific interfaces.

Any device implementing manufacturer specific interfaces is required to have an Equipment Manufacturer Code (EMC) stored in the appropriate attribute of Device Information service (see HF-Service document, [REF 3]) and provide the EMC during HF registration in the *Discriminator Value* message field. Consult the *Register Device* command section in the HF-Service document [REF 3] for more details, especially the Discriminator Type section.

Any interface must be identifiable by a UID. The HF protocol saves IDs from `0x7F00` to `0x7FFE` (see Table 1 in chapter 5) for custom manufacturer interfaces. However, their uniqueness is not guaranteed so is up to the manufacturer to provide a means to uniquely identify its custom interface and/or handle possible ID collisions.

Any manufacturer-built interface must conform to the general structure in which HAN-FUN interfaces are defined.

Annex:

Terms and Conditions of Use

Version: 23 January 2014

The copyrights in this Document and the specifications contained herein are owned by the ULE Alliance and its Members (hereinafter, the 'ULE Alliance'). Use of this Document and the specifications and any related intellectual property (collectively, the "Specification"), is governed by this Notice and the IPR Regulation of the ULE Alliance, where appropriate.

Use of the Specification by anyone who is not a member of the ULE Alliance or an Adopter is prohibited. However, such parties shall be permitted to view this Document. Requests for permission to reprint this Document, in whole or in part, or requests for a license to reproduce and/or distribute this Document, in any form, must be submitted via email to secretariat@ulealliance.org or in writing to: ULE Alliance Secretariat, Wabernstr. 40, 3007 Bern, Switzerland.

Elements of this Document and Specification may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a Member of ULE Alliance). Nothing in this Document or Specification may be construed as a license to use such intellectual property, and such license must be separately sought with the right holder. The ULE Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights

THE DOCUMENT AND SPECIFICATION ARE PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFI- CATION OR SAMPLE.

Each Member and Adopter hereby acknowledges that products equipped with the Specification (hereinafter, 'Products') may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Each Member and Adopter is solely responsible for the compliance by Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Products related to such regulations within the applicable jurisdictions. Each Member and Adopter acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

IN NO EVENT WILL THE ULE ALLIANCE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The ULE Alliance reserves the right to adopt any changes or alterations to the Document or Specification as it deems necessary or appropriate without notice.