OEMan Communications Specification

version 1.0

1. Version

Version	Date	Details	Editer
1.0	31-January-2015	First version	Adam Tyler

2. Table of contents

- Version
- Table of contents
- Summary
- Standard summary
- Terminology
- Variations from Emoncms protocol
- Communication examples
    - Registration
        - Registration example
    - Normal operation
        - Dumb node example
        - Smart node example
- Variable standards
- OEMan attribute library
- Functional Domain Groups
        - General
        - Closures
        - HVAC
        - Lighting
        - Measurement and Sensing
        - Security and Safety
        - Protocol Interfaces
    - Attribute Sets
        - General Functional Domain Attribute Sets
            - Basic group
            - Power configuration group
            - Device Temperature Configuration group
            - Identify group
            - Groups group
            - Scenes group
            - On/off group
            - On/off Switch Configuration group
            - Level Control group
            - Alarms group
            - Time group
            - RSSI Location group
            - Analog Input (Basic) group
            - Analog Output (Basic) group

- Analog Value (Basic) group
- Binary Input (Basic) group
- Binary Output (Basic) group
- Binary Value (Basic) group
- Multistate Input (Basic) group
- Multistate Output (Basic) group
- Multistate Value (Basic) group
- Commissioning group

Closures Functional Domain Attribute Sets
- Shade Configuration Group
- Door Lock Group

HVAC Functional Domain Attribute Sets
- Pump Configuration and Control Group
- Thermostat Group
  - Thermostat Information Attribute Set
  - Thermostat Settings Attribute Set
- Fan Control Group
- Dehumidification Control Group
- Thermostat User Interface Configuration Group

Lighting Functional Domain Attribute Sets
- Colour control Group
- Ballast Configuration Group

Measurement and sensing Functional Domain Attribute Sets
- Illuminance Measurement Group
- Illuminance Level Sensing Group
- Temperature Measurement Group
- Pressure Measurement Group
  - Pressure Measurement Information Attribute Set
  - Extended Pressure Measurement Information Attribute Set
- Flow Measurement Group
- Relative Humidity Measurement Group
  - Relative Humidity Measurement Information Attribute Set
- Occupancy Sensing Group

Security and safety Functional Domain Attribute Sets
- IAS Zone Group
- IAS ACE Group
- IAS WD Group

Protocol interfaces Functional Domain Attribute Sets
- Generic Tunnel Group
- BACnet Protocol Tunnel Group
- Analog Input (BACnet Regular) Group
- Analog Input (BACnet Extended) Group
- Analog Output (BACnet Regular) Group
- Analog Output (BACnet Extended) Group
- Analog Value (BACnet Regular) Group
- Analog Value (BACnet Extended) Group
- Binary Input (BACnet Regular) Group
- Binary Input (BACnet Extended) Group

    - Binary Output (BACnet Regular) Group
    - Binary Output (BACnet Extended) Group
    - Binary Value (BACnet Regular) Group
    - Binary Value (BACnet Extended) Group
    - Multistate Input (BACnet Regular) Group
    - Multistate Input (BACnet Extended) Group
    - Multistate Output (BACnet Regular) Group
    - Multistate Output (BACnet Extended) Group
    - Multistate Value (BACnet Regular) Group
    - Multistate Value (BACnet Extended) Group
- OEMan Status codes

3. Summary

The following is the first published draft of the communications standard for the Open Energy Management (OEMan) system. This standard has been through various iterations so far, but has currently settled in this current form mainly due to its compatibility with the Open Energy Monitoring OEMon) projects Emoncms web-app, which is being modified to allow the desired control functionality of the OEMan project.

There are a number of areas where the standard has had to be moulded away from the ideal to allow it to work with Emoncms, these are compromises that are felt necessary yet allow for the majority of the required functionality and can be revisited and revised at a later date.

4. Standard summary

Within this standard three types of communication are defined, these are:

- Registration - on first startup a new device or node, for example a new temperature sensor, will make contact with Emoncms, which will then store its details and assign it a node id.
- Dumb node input - nodes which are not powerful enough to process poll requests, need to be in sleep mode and uncontactable for long periods of time to conserve power, or the event which is being monitored is infrequent or random enough that it is not worth polling for data, these will send out data to Emoncms as and when they desire or are able to.
- Smart node request - nodes which are able to respond to poll requests from Emoncms then return the requested data.

As discussed, this standard is based on the communications protocol currently used by Emoncms. It is therefore recommended that the equivalent documents for the Emoncms are read alongside this one, as this document intend only to highlight the new requirements OEMan has placed on the existing protocol; it is intended that all existing actions will continue to work.

For Emoncms inputcomands see http://emoncms.org/input/api

For Emoncms feed requests see http://emoncms.org/feed/api

5. Terminology

- Node - a node is a device within the system that inputs or outputs data
- Input - a input is a raw, unsaved reading. Inputs are overwritten next time a reading is taken
- Feed - a feed is a saved and potentially processed reading
- Attributes - the functionality of a node, i.e. a temperature setpoint, a high humidity alarm point or a motor speed; note each node may have multiple attributes.
- Precooling - Cooling a building in the early (cooler) part of the day, so that the thermal mass of the building decreases cooling needs in the later (hotter) part of the day.
- 4-pipes: In a 4-pipe HVAC fan coil system, heated and chilled water each have their own supply and return pipes, while in a 2 pipe system they share the same supply and return. With a 4-pipes system, heating and cooling can take place at the same time in different locations of a building. With a 2-pipes system, only heating or cooling can take place in the whole building.

6. Variations from Emoncms protocol

There are four main variations within this standard as to how elements are used. these are:

- APIkey - in the Emoncms protocol, the APIkey is used to direct input to the correct user on the server, and to differentiate between read and write permissions. It is intended to use the APIkey in OEMan as a to address, or rather the port number to the Emoncms ip address; it is intended at a later date to consider a change within Emoncms to correct this naming discrepancy. Direction of data to users and read/write permissions, will be taken over by a modified user module.
- Node id - Similarly to the APIkey, in OEMan the node id is used as a from address, or rather the port number to the nodes ip address; this is not much of a change in mindset from its current use
- Input key - The key for each input will be used to encode the type of input that is being sent; the format of this is is defined below in section 8.
- Feed id - The id for each feed will match the input key it is related to, but with the addition of the associated node id.

7. Communication examples

7.1. Registration

On first startup a new node will have already been configured with the servers ip address and API key, it will then send out its own ip address to Emoncms. Emoncms will then log the ip address, assign a node ip, and then send that node id back to the node. The node will then store the assigned node id; node id’s are intended to be retained for the life of the node.

Once a node has been assigned a node id, it then must cycle through all of its attributes, registering them all with Emoncms.

Below is an example of the communications between a node and Emoncms at first startup:

7.1.1. Registration example

Note all items in square brackets [] are saved variables.

(node to Emoncms) - Registration of node and application for node id

http://[server ip address]/emoncms/register/register.json?apikey=[Emoncms APIkey]&nodeip=[ip address of node]&timeout=[specified timeout]

(response to request)

[assigned node id]
or

[error code see section 10]

(node to Emoncms) - Registration of attribute (should be repeated for each of the nodes attributes)

http://[server ip address]/emoncms/register/setup.json?apikey=[Emoncms APIkey]&node=[assigned node id]&json={[group id][attribute id][attribute number],[attribute default value]}&timeout=[specified timeout]

(response to request)

[ok]
or

[error code see section 10]

7.2. Normal operation

In normal operation, either the node is dumb and sends in inputs as and when it is programed to, or is a smart node and responds to poll requests from Emoncms.

Below are two examples of communications, dumb and smart. These communications could be between a node and Emoncms, between two nodes, between Emoncms and a node or between two Emoncms's.

7.2.1. Dumb node example

Note all items in square brackets [] are saved variables.

(node to Emoncms) - sending input to sever at pre programmed interval or on an event http://[server ip address]/emoncms/input/post.json?apikey=[Emoncms APIkey]&node=[assigned node id]&json={[group id][attribute id][attribute number],[attribute value]}&timeout=[specified timeout]

(response to request)

[ok]
or

[error code see section 10]

7.2.2. Smart node example

Note all items in square brackets [] are saved variables.

(Emoncms to node) - Emoncms requesting a specific attribute value from a node

http://[node ip address]/emoncms/feed/value.json?apikey=[Emoncms APIkey]&node=[assigned node id]&id=[group id][attribute id][attribute number][assigned node id]&timeout=[specified timeout]

(response to request)

[requested value]
or

[error code see section 10]

8. Variable standards

Note that the order of variables outside the json packet does not matter.

Identifier	**apikey**	**nodeip**	**node**	**timeout**
Mandatory/ Optional	M	M	M	O
Description	id for Emoncms	Ip address of the node	identifier of the node. Integer value starting at 1	specifies the maximum wait time before the request is aborted (in seconds)
Length	32 bytes	45 bytes	variable	variable
example	b742b99d9880b27ff50ce73b17c2e224	ABCD:ABCD:ABCD:ABCD:ABCD:ABCD:192.168.158.190	23	60

Table 6.1: OEMan communication variables

Note that order of variables within the json packet does matter.

Identifier	**Group ID**	**Attribute ID**	**Attribute Number**	**Assigned Node ID**
Mandatory/ Optional	M	M	M	M
Description	identifier for group that the attribute is under (see section 9)	identifier for the attribute (see section 9)	unique identifier for the attribute within the node (multiple attributes of the same type are allowed within one node)	As for node in table 1. Identifier of the node. Integer value starting at 1
Length	2 bytes (hexadecimal)	2 bytes (hexadecimal)	12 bits (hexadecimal)	variable
example	0x0201	0x0001	0x0Fa	23

Table 6.2: OEMan communications variables within json packet

9. OEMan attribute library

The OEMan attribute library is divided into a number of functional domains, each domain addressing groups relating to specific functionality. The functional domains defined in the OEMan attribute library are listed in Table 7.1.

Currently all the attributes and attribute sets have not been defined; those that are not currently complete will be be updated over time.

Functional Domain	**Cluster ID Range**
General	0x0000 – 0x00ff
Closures	0x0100 – 0x01ff
HVAC	0x0200 – 0x02ff
Lighting	0x0300 – 0x03ff
Measurement and sensing	0x0400 – 0x04ff
Security and safety	0x0500 – 0x05ff
Protocol interfaces	0x0600 – 0x06ff

Table 7.1: Functional domains defined in the OEMan attribute library

The structure of each of these functional groups is described below

9.1. Functional Domain Groups

9.1.1. General

The general functional domain contains groups and information that provides generally applicable functions and attributes that are not specific to other functional domains.

This functional domain specifies the groups listed in Table 7.1.1.1.

Group ID	**Group Name**	**Description**
0x0000	Basic	Attributes for determining basic information about a device, setting user device information such as location, and enabling a device.

0x0001	Power configuration	Attributes for determining more detailed information about a device's power source(s), and for configuring under/over voltage alarms.
0x0002	Device Temperature Configuration	Attributes for determining information about a device's internal temperature, and for configuring under/over temperature alarms.
0x0003	Identify	Attributes and commands for putting a device into Identification mode (e.g. flashing a light)
0x0004	Groups	Attributes and commands for group configuration and manipulation.
0x0005	Scenes	Attributes and commands for scene configuration and manipulation.
0x0006	On/off	Attributes and commands for switching devices between 'On' and 'Off' states.
0x0007	On/off Switch Configuration	Attributes and commands for configuring On/Off switching devices
0x0008	Level Control	Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off'.
0x0009	Alarms	Attributes and commands for sending notifications and configuring alarm functionality.
0x000a	Time	Attributes and commands that provide a basic interface to a real-time clock.
0x000b	RSSI Location	Attributes and commands that provide a means for exchanging location information and channel parameters among devices.
0x000c	Analog Input (Basic)	An interface for reading the value of an analog measurement and accessing various characteristics of that measurement.
0x000d	Analog Output (Basic)	An interface for setting the value of an analog output (typically to the environment) and accessing various characteristics of that value.
0x000e	Analog Value (Basic)	An interface for setting an analog value, typically used as a control system parameter, and accessing various characteristics of that value.
0x000f	Binary Input (Basic)	An interface for reading the value of a binary measurement and accessing various characteristics of

		that measurement.
0x0010	Binary Output (Basic)	An interface for setting the value of a binary output (typically to the environment) and accessing various characteristics of that value.
0x0011	Binary Value (Basic)	An interface for setting a binary value, typically used as a control system parameter, and accessing various characteristics of that value.
0x0012	Multistate Input (Basic)	An interface for reading the value of a multistate measurement and accessing various characteristics of that measurement.
0x0013	Multistate Output (Basic)	An interface for setting the value of a multistate output (typically to the environment) and accessing various characteristics of that value.
0x0014	Multistate Value (Basic)	An interface for setting a multistate value, typically used as a control system parameter, and accessing various characteristics of that value.
0x0015	Commissioning	Attributes and commands for commissioning and managing a ZigBee device.
0x0016 – 0x00ff	-	Reserved.

Table 7.1.1.1: Groups of the General Functional Domain

9.1.2. Closures

The closures functional domain contains groups and information to build devices in the closure domain, e.g. shade controllers.

This functional domain specifies the groups listed in Table 7.1.2.1.

Group ID	**Group Name**	**Description**
0x0100	Shade Configuration	Attributes and commands for configuring a shade.
0x0101	Door Lock	An interface for controlling a door lock.
0x0102 – 0x01ff	-	Reserved.

Table 7.1.2.1: Groups of the Closures Functional Domain

9.1.3. HVAC

The HVAC functional domain contains groups and information to build devices in the HVAC domain, e.g. pumps.

This functional domain specifies the groups listed in Table 7.1.3.1.

Group ID	Group Name	Description
0x0200	Pump Configuration and Control	An interface for configuring and controlling pumps.
0x0201	Thermostat	An interface for configuring and controlling the functionality of a thermostat.
0x0202	Fan Control	An interface for controlling a fan in a heating / cooling system.
0x0203	Dehumidification Control	An interface for controlling dehumidification.
0x0204	Thermostat User Interface Configuration	An interface for configuring the user interface of a thermostat (which may be remote from the thermostat).
0x0205 – 0x02ff	-	Reserved

Table 7.1.3.1: Clusters of the HVAC Functional Domain

9.1.4. Lighting

The lighting functional domain contains groups and information to build devices in the lighting domain, e.g. ballast units.

This functional domain specifies the groups listed in Table 7.1.4.1.

Group ID	Group Name	Description
0x0300	Color control	Attributes and commands for controlling the color properties of a color-capable light
0x0301	Ballast Configuration	Attributes and commands for configuring a lighting ballast
0x0302 – 0x03ff	-	Reserved.

Table 7.1.4.1: Groups of the Lighting Functional Domain

9.1.5. Measurement and Sensing

The measurement and sensing functional domain contains groups and information to build devices in the measurement and sensing domain, e.g. a temperature sensor or an occupancy sensor.

This functional domain specifies the clusters listed in Table 7.1.5.1.

Group ID	**Group Name**	**Description**
0x0400	Illuminance measurement	Attributes and commands for configuring the measurement of illuminance, and reporting illuminance measurements.
0x0401	Illuminance level sensing	Attributes and commands for configuring the sensing of illuminance levels, and reporting whether illuminance is above, below, or on target.
0x0402	Temperature measurement	Attributes and commands for configuring the measurement of temperature, and reporting temperature measurements.
0x0403	Pressure measurement	Attributes and commands for configuring the measurement of pressure, and reporting pressure measurements.
0x0404	Flow measurement	Attributes and commands for configuring the measurement of flow, and reporting flow rates.
0x0405	Relative humidity measurement	Attributes and commands for configuring the measurement of relative humidity, and reporting relative humidity measurements.
0x0406	Occupancy sensing	Attributes and commands for configuring occupancy sensing, and reporting occupancy status.
0x0407 – 0x04ff	-	Reserved.

Table 7.1.5.1: Groups of the Measurement and Sensing Functional Domain

9.1.6. Security and Safety

The security and safety functional domain contains groups and information to build devices in the security and safety domain, e.g. alarm units.

This functional domain specifies the groups listed in Table 7.1.6.1.

Group ID	Group Name	Description
0x0500	IAS Zone	Attributes and commands for IAS security zone devices.
0x0501	IAS ACE	Attributes and commands for IAS Ancillary Control Equipment.
0x0502	IAS WD	Attributes and commands for IAS Warning Devices.
0x0503 – 0x05ff	-	Reserved.

Table 7.1.6.1: Groups of the Security and Safety Functional Domain

9.1.7. Protocol Interfaces

The protocol interfaces functional domain contains groups and information to build devices to interface to other protocols, e.g. BACnet.

This functional domain specifies the groups listed in Table 7.1.7.1.

Group ID	Group Name	Description
0x0600	Generic Tunnel	The minimum common commands and attributes required to tunnel any protocol.
0x0601	BACnet Protocol Tunnel	Commands and attributes required to tunnel the BACnet protocol.
0x0602	Analog Input (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of an analog measurement.
0x0603	Analog Input (BACnet Extended)	An interface for accessing a number of BACnet based attributes of an analog measurement.
0x0604	Analog Output (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of an analog output.
0x0605	Analog Output (BACnet Extended)	An interface for accessing a number of BACnet based attributes of an analog output.
0x0606	Analog Value (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of an analog value, typically used as a control system parameter.

0x0607	Analog Value (BACnet Extended)	An interface for accessing a number of BACnet based attributes of an analog value, typically used as a control system parameter.
0x0608	Binary Input (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of a binary measurement.
0x0609	Binary Input (BACnet Extended)	An interface for accessing a number of BACnet based attributes of a binary measurement.
0x060a	Binary Output (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of a binary output.
0x060b	Binary Output (BACnet Extended)	An interface for accessing a number of BACnet based attributes of a binary output.
0x060c	Binary Value (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of a binary value, typically used as a control system parameter.
0x060d	Binary Value (BACnet Extended)	An interface for accessing a number of BACnet based attributes of a binary value, typically used as a control system parameter.
0x060e	Multistate Input (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of a multistate measurement.
0x060f	Multistate Input (BACnet Extended)	An interface for accessing a number of BACnet based attributes of a multistate measurement.
0x0610	Multistate Output (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of a multistate output.
0x0611	Multistate Output (BACnet Extended)	An interface for accessing a number of BACnet based attributes of a multistate output.
0x0612	Multistate Value (BACnet Regular)	An interface for accessing a number of commonly used BACnet based attributes of a multistate value, typically used as a control system parameter.
0x0613	Multistate Value (BACnet Extended)	An interface for accessing a number of BACnet based attributes of a multistate value, typically used as a control system parameter.
0x0614 – 0x06ff	-	Reserved.

Table 7.1.7.1: Groups of the Protocol Interfaces Functional Domain

9.2. Attribute Sets

9.2.1. General Functional Domain Attribute Sets

9.2.1.1. Basic group

To be completed

9.2.1.2. Power configuration group

To be completed

9.2.1.3. Device Temperature Configuration group

To be completed

9.2.1.4. Identify group

To be completed

9.2.1.5. Groups group

To be completed

9.2.1.6. Scenes group

To be completed

9.2.1.7. On/off group

To be completed

9.2.1.8. On/off Switch Configuration group

To be completed

9.2.1.9. Level Control group

To be completed

9.2.1.10. Alarms group

To be completed

9.2.1.11. Time group

To be completed

9.2.1.12. RSSI Location group

To be completed

9.2.1.13. Analog Input (Basic) group

To be completed

9.2.1.14. Analog Output (Basic) group

To be completed

9.2.1.15. Analog Value (Basic) group

To be completed

9.2.1.16. Binary Input (Basic) group

To be completed

9.2.1.17. Binary Output (Basic) group

To be completed

9.2.1.18. Binary Value (Basic) group

To be completed

9.2.1.19. Multistate Input (Basic) group

To be completed

9.2.1.20. Multistate Output (Basic) group

To be completed

9.2.1.21. Multistate Value (Basic) group

To be completed

9.2.1.22. Commissioning group

To be completed

9.2.2. Closures Functional Domain Attribute Sets

9.2.2.1. Shade Configuration Group

To be completed

9.2.2.2. Door Lock Group

To be completed

9.2.3. HVAC Functional Domain Attribute Sets

9.2.3.1. Pump Configuration and Control Group

To be completed

9.2.3.2. Thermostat Group

This cluster provides an interface to the functionality of a thermostat.

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets for Thermostat are listed in Table 7.2.3.2.

Attribute Set Identifier	**Description**
0x000	Thermostat Information

0x001	Thermostat Settings
0x002 – 0xfff	Reserved

Table 7.2.3.2: Currently Defined Thermostat Attribute Sets

9.2.3.2.1. Thermostat Information Attribute Set

The Thermostat Information attribute set contains the attributes summarized in Table 7.2.3.2.1.1.

Identifier	**Name**	**Type**	**Range**	**Default**	**Mandatory/ Optional**
0x0000	LocalTemperature	Signed 16-bit integer	0x954d – 0x7fff	-	M
0x0001	OutdoorTemperature	Signed 16-bit integer	0x954d – 0x7fff	-	O
0x0002	Ocupancy	8-bit bitmap	0000000x	0	O
0x0003	AbsMinHeatSetpointLimit	Signed 16-bit integer	0x954d – 0x7fff	0x02bc (7°C)	O
0x0004	AbsMaxHeatSetpointLimit	Signed 16-bit integer	0x954d – 0x7fff	0x0bb8 (30°C)	O
0x0005	AbsMinCoolSetpointLimit	Signed 16-bit integer	0x954d – 0x7fff	0x0640 (16°C)	O
0x0006	AbsMaxCoolSetpointLimit	Signed 16-bit integer	0x954d – 0x7fff	0x0c80 (32°C)	O
0x0007	PICoolingDemand	Unsigned 8-bit integer	0x00 – 0x64	-	O
0x0008	PIHeatingDemand	Unsigned 8-bit integer	0x00 – 0x64	-	O

Table 7.2.3.2.1.1: Attributes of the Thermostat Information Attribute Set

LocalTemperature Attribute

LocalTemperature represents the temperature in degrees Celsius, as measured locally or remotely as follows:

LocalTemperature = 100 x temperature in degrees Celsius.

Where -273.15°C <= temperature <= 327.67 ºC, corresponding to a LocalTemperature in the range 0x954d to 0x7fff.

The maximum resolution this format allows is 0.01 ºC.

A LocalTemperature of 0x8000 indicates that the temperature measurement is invalid.

OutdoorTemperature Attribute

OutdoorTemperature represents the outdoor temperature in degrees Celsius, as measured locally or remotely. It is measured as described for LocalTemperature.

Occupancy Attribute

Occupancy specifies whether the heated/cooled space is occupied or not, as measured locally or remotely. If bit 0 = 1, the space is occupied, else it is unoccupied. All other bits are reserved.

AbsMinHeatSetpointLimit Attribute

The MinHeatSetpointLimit attribute specifies the absolute minimum level that the heating setpoint may be set to. The value is calculated as described in the LocalTemperature attribute.

AbsMaxHeatSetpointLimit Attribute

The MaxHeatSetpointLimit attribute specifies the absolute maximum level that the heating setpoint may be set to. The value is calculated as described in the LocalTemperature attribute.

AbsMinCoolSetpointLimit Attribute

The MinCoolSetpointLimit attribute specifies the absolute minimum level that the cooling setpoint may be set to. The value is calculated as described in the LocalTemperature attribute.

AbsMaxCoolSetpointLimit Attribute

The MaxCoolSetpointLimit attribute specifies the absolute maximum level that the cooling setpoint may be set to. The value is calculated as described in the LocalTemperature attribute.

PICoolingDemand Attribute

The PICoolingDemand attribute is 8 bits in length and specifies the level of cooling demanded by the PI (proportional integral) control loop in use by the thermostat (if any), in percent. This value is 0 when the thermostat is in “off” or “heating” mode.

PIHeatingDemand Attribute

The PIHeatingDemand attribute is 8 bits in length and specifies the level of heating demanded by the PI loop in percent. This value is 0 when the thermostat is in “off” or “cooling” mode.

9.2.3.2.2. Thermostat Settings Attribute Set

The Thermostat settings attribute set contains the attributes summarized in Table 7.2.3.2.2.1.

Identifier	**Name**	**Type**	**Range**	**Default**	**Mandatory/ Optional**
0x0010	LocalTemp erature Calibration	Signed 8-bit integer	0xE7 – 0x19	0x00 (0°C)	O
0x0011	OccupiedC ooling Setpoint	Signed 16-bit integer	MinCoolSet point Limit – MaxCoolSe tpoint Limit	0x0a28 (26°C)	M
0x0012	OccupiedH eating Setpoint	Signed 16-bit integer	MinHeatSet point Limit – MaxHeatSe tpoint Limit	0x07d0 (20°C)	M
0x0013	Unoccupied Cooling Setpoint	Signed 16-bit integer	MinCoolSet point Limit – MaxCoolSe tpoint Limit	0x0a28 (26°C)	O
0x0014	Unoccupied Heating	Signed 16-bit	MinHeatSet point Limit –	0x07d0 (20°C)	O

	Setpoint	integer	MaxHeatSetpoint Limit		
0x0015	MinHeatSetpoint Limit	Signed 16-bit integer	0x954d – 0x7fff	0x02bc (7°C)	O
0x0016	MaxHeatSetpoint Limit	Signed 16-bit integer	0x954d – 0x7fff	0x0bb8 (30°C)	O
0x0017	MinCoolSetpoint Limit	Signed 16-bit integer	0x954d – 0x7fff	0x02bc(7°C)	O
0x0018	MaxCoolSetpoint Limit	Signed 16-bit integer	0x954d – 0x7fff	0x0bb8 (30°C)	O
0x0019	MinSetpoint Dead Band	Signed 8-bit integer	0x0a – 0x19	0x19 (2.5°C)	O
0x001a	RemoteSensing	8-bit bitmap	00000xxx	0	O
0x001b	ControlSequenceOf Operation	8-bit enumeration	0x00 – 0x05	0x04	M
0x001c	SystemMode	8-bit enumeration	See Table 6.15	0x01	M
0x001d	AlarmMask	8-bit bitmap	00000xxx	0	O

Table 7.2.3.2.2.1: Attributes of the Thermostat Settings Attribute Set

LocalTemperatureCalibration Attribute

The LocalTemperatureCalibration attribute specifies the offset that can be added/ subtracted to the actual displayed room temperature, in steps of 0.1°C. The range of this offset is –2.5 °C to +2.5 °C).

OccupiedCoolingSetpoint Attribute

The OccupiedCoolingSetpoint attribute is 16 bits in length and specifies the cooling mode setpoint when the room is occupied. It shall be set to a value in the range defined by the MinCoolSetpointLimit and MaxCoolSetpointLimit attributes. The value is calculated as described in the LocalTemperature attribute.

The OccupiedHeatingSetpoint attribute shall always be below the value specified in the OccupiedCoolingSetpoint by at least SetpointDeadband. If an attempt is made to set it such that this condition is violated, a default response command with the status code INVALID_VALUE (see section 10) shall be returned. This shall apply to all attempts to set values of attributes which violate similar conditions.

If it is unknown if the room is occupied or not, this attribute shall be used as the cooling mode setpoint.

OccupiedHeatingSetpoint Attribute

The OccupiedHeatingSetpoint attribute is 16 bits in length and specifies the heating mode setpoint when the room is occupied. It shall be set to a value in the range defined by the MinHeatSetpointLimit and MaxHeatSetpointLimit attributes. The value is calculated as described in the LocalTemperature attribute. The OccupiedCoolingSetpoint attribute shall always be above the value specified in the OccupiedHeatingSetpoint by at least SetpointDeadband.

If it is unknown if the room is occupied or not, this attribute shall be used as the cooling mode setpoint.

UnoccupiedCoolingSetpoint Attribute

The UnoccupiedCoolingSetpoint attribute is 16 bits in length and specifies the cooling mode setpoint when the room is unoccupied. It shall be set to a value in the range defined by the MinCoolSetpointLimit and MaxCoolSetpointLimit attributes. The value is calculated as described in the LocalTemperature attribute. The UnoccupiedHeatingSetpoint attribute shall always be below the value specified in the UnoccupiedCoolingSetpoint by at least SetpointDeadband.

If it is unknown if the room is occupied or not, this attribute shall not be used.

UnoccupiedHeatingSetpoint Attribute

The UnoccupiedHeatingSetpoint attribute is 16 bits in length and specifies the heating mode setpoint when the room is unoccupied. It shall be set to a value in the range defined by the MinHeatSetpointLimit and MaxHeatSetpointLimit attributes. The value is calculated as described in the LocalTemperature attribute. The UnoccupiedCoolingSetpoint attribute shall always be below the value specified in the UnoccupiedHeatingSetpoint by at least SetpointDeadband.

If it is unknown if the room is occupied or not, this attribute shall not be used.

MinHeatSetpointLimit Attribute

The MinHeatSetpointLimit attribute specifies the minimum level that the heating setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be greater than or equal to AbsMinHeatSetpointLimit. If this attribute is not present, it shall be taken as equal to AbsMinHeatSetpointLimit.

This attribute, and the following three attributes, allow the user to define setpoint limits more constrictive than the manufacturer imposed ones. Limiting users (e.g., in a commercial building) to such setpoint limits can help conserve power.

MaxHeatSetpointLimit Attribute

The MaxHeatSetpointLimit attribute specifies the maximum level that the heating setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be less than or equal to AbsMaxHeatSetpointLimit. If this attribute is not present, it shall be taken as equal to AbsMaxHeatSetpointLimit.

MinCoolSetpointLimit Attribute

The MinCoolSetpointLimit attribute specifies the minimum level that the cooling setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be greater than or equal to AbsMinCoolSetpointLimit. If this attribute is not present, it shall be taken as equal to AbsMinCoolSetpointLimit.

MaxCoolSetpointLimit Attribute

The MaxCoolSetpointLimit attribute specifies the maximum level that the cooling setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be less than or equal to AbsMaxCoolSetpointLimit. If this attribute is not present, it shall be taken as equal to AbsMaxCoolSetpointLimit.

MinSetpointDeadBand Attribute

The MinSetpointDeadBand attribute specifies the minimum difference between the Heat Setpoint and the Cool SetPoint, in steps of 0.1°C. Its range is 0x0a to 0x19 (1°C to 2.5°C).

RemoteSensing Attribute

The RemoteSensing attribute is an 8-bit bitmap that specifies whether the local temperature, outdoor temperature and occupancy are being sensed by internal sensors or remote networked sensors. The meanings of individual bits are detailed in Table 7.2.3.2.2.1.

Bit Number	Description	
0	0	– local temperature sensed internally
	1	– local temperature sensed remotely
1	0	– outdoor temperature sensed internally
	1	– outdoor temperature sensed remotely
2	0	– occupancy sensed internally
	1	– occupancy sensed remotely
3 - 7	Reserved	

Table 7.2.3.2.2.1: RemoteSensing Attribute Bit Values

ControlSequenceOfOperation Attribute

The ControlSequenceOfOperation attribute specifies the overall operating environment of the thermostat, and thus the possible system modes that the thermostat can operate in. It shall be set to one of the non-reserved values in Table 7.2.3.2.2.2. (Note - it is not mandatory to support all values).

Attribute Value	Description	Possible Values of SystemMode
0x00	Cooling Only	Heat and Emergency are not possible
0x01	Cooling With Reheat	Heat and Emergency are not possible
0x02	Heating Only	Cool and precooling $^{\text{see section 3}}$ are not possible
0x03	Heating With Reheat	Cool and precooling are not possible
0x04	Cooling and Heating	All modes are possible

	4-pipes see section 3	
0x05	Cooling and Heating 4-pipes with Reheat	All modes are possible
0x06 – 0xfe	Reserved	-

Table 7.2.3.2.2.2: ControlSequenceOfOperation Attribute Values

SystemMode Attribute

The SystemMode attribute specifies the current operating mode of the thermostat,. It shall be set to one of the non-reserved values in Table 7.2.3.2.2.3, as limited by Table 7.2.3.2.2.4. (Note - it is not mandatory to support all values).

Attribute Value	**Description**
0x00	Off
0x01	Auto
0x03	Cool
0x04	Heat
0x05	Emergency heating
0x06	Precooling see section 3
0x07	Fan only
0x02, 0x08 – 0xfe	Reserved

Table 7.2.3.2.2.3: SystemMode Attribute Values

The interpretation of the Heat, Cool and Auto values of SystemMode is shown in Table 7.2.3.2.2.4.

Attribute Values	**Temperature Below Heat Setpoint**	**Temperature Between Heat Setpoint and Cool Setpoint**	**Temperature Above Cool Setpoint**

Heat	Temperature below target	Temperature on target	Temperature on target
Cool	Temperature on target	Temperature on target	Temperature above target
Auto	Temperature below target	Temperature on target	Temperature above target

Table 7.2.3.2.2.4: Interpretation of SystemMode Values

AlarmMask Attribute

The AlarmMask attribute specifies whether each of the alarms listed in Table 7.2.3.2.2.5 is enabled. When the bit number corresponding to the alarm code is set to 1, the alarm is enabled, else it is disabled. Bits not corresponding to a code in the table are reserved.

When the Alarms cluster is implemented on a device, and one of the alarm conditions included in this table occurs, an alarm notification is generated, with the alarm code field set as listed in the table.

Alarm Code	**Alarm Condition**
0	Initialization failure. The device failed to complete initialization at power-up.
1	Hardware failure
2	Self-calibration failure

Table 7.2.3.2.2.5: Alarm Codes

9.2.3.3. Fan Control Group

To be completed

9.2.3.4. Dehumidification Control Group

To be completed

9.2.3.5. Thermostat User Interface Configuration Group

To be completed

9.2.4. Lighting Functional Domain Attribute Sets

9.2.4.1. Colour control Group

To be completed

9.2.4.2. Ballast Configuration Group

To be completed

9.2.5. Measurement and sensing Functional Domain Attribute Sets

9.2.5.1. Illuminance Measurement Group

To be completed

9.2.5.2. Illuminance Level Sensing Group

To be completed

9.2.5.3. Temperature Measurement Group

To be completed

9.2.5.4. Pressure Measurement Group

The group provides an interface to pressure measurement functionality, including configuration and provision of notifications of pressure measurements.

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 7.2.5.4.1.

Attribute Set Identifier	**Description**
0x000	Pressure Measurement Information
0x001	Extended Pressure Measurement

	Information
0x002 – 0xfff	Reserved

Table 7.2.5.4.1: Pressure Measurement Attribute Sets

9.2.5.4.1. Pressure Measurement Information Attribute Set

The Pressure Measurement Information attribute set contains the attributes summarized in Table 7.2.5.4.1.1.

Attribute ID	**Name**	**Type**	**Range**	**Default**	**Mandatory/ Optional**
0x0000	MeasuredValue	Signed 16-bit integer	MinMeasuredValue to MaxMeasuredValue	0	M
0x0001	MinMeasuredValue	Signed 16-bit integer	0x8001-0x7ffe	-	M
0x0002	MaxMeasuredValue	Signed 16-bit integer	0x8002-0x7fff	-	M
0x0003	Tolerance	Unsigned 16-bit integer	0x0000 – 0x0800	-	O

Table 7.2.5.4.1.1: Pressure Measurement Information Attribute Set

This set provides for measurements with a fixed maximum resolution of 0.1 kPa.

MeasuredValue Attribute

MeasuredValue represents the pressure in kPa as follows:

MeasuredValue = 10 x Pressure

Where -3276.7 kPa <= Pressure <= 3276.7 kPa, corresponding to a MeasuredValue in the range 0x8001 to 0x7fff.

A MeasuredValue of 0x8000 indicates that the pressure measurement is invalid.

MinMeasuredValue Attribute

The MinMeasuredValue attribute indicates the minimum value of MeasuredValue that can be measured. A value of 0x8000 means this attribute is not defined

MaxMeasuredValue Attribute

The MaxMeasuredValue attribute indicates the maximum value of MeasuredValue that can be measured. A value of 0x8000 means this attribute is not defined.

MaxMeasuredValue shall be greater than MinMeasuredValue. MinMeasuredValue and MaxMeasuredValue define the range of the sensor.

Tolerance Attribute

The Tolerance attribute indicates the magnitude of the possible error that is associated with MeasuredValue. The true value is located in the range (MeasuredValue – Tolerance) to (MeasuredValue + Tolerance).

9.2.5.4.2. Extended Pressure Measurement Information Attribute Set

To be completed

9.2.5.5. Flow Measurement Group

To be completed

9.2.5.6. Relative Humidity Measurement Group

The group provides an interface to relative humidity measurement functionality, including configuration and provision of notifications of relative humidity measurements.

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 7.2.5.6.1.

Attribute Set Identifier	Description
0x000	Relative Humidity Measurement Information
0x001 – 0xfff	Reserved

Table 7.2.5.6.1 Relative Humidity Measurement Attribute Sets

9.2.5.6.1. Relative Humidity Measurement Information Attribute Set

The Relative Humidity Measurement Information attribute set contains the attributes summarized in Table 7.2.5.6.1.

Identifier	Name	Type	Range	Default	Mandatory/ Optional
0x0000	MeasuredValue	Unsigned 16- bit integer	MinMeasuredValue to MaxMeasuredValue	-	M
0x0001	MinMeasuredValue	Unsigned 16-bit integer	0x0000 – 0x270f	-	M
0x0002	MaxMeasuredValue	Unsigned 16-bit integer	0x0001 – 0x2710	-	M
0x0003	Tolerance	Unsigned 16-bit integer	0x0000 – 0x0800	-	O

Table 7.2.5.6.1: Attributes of the Relative Humidity Measurement Information Attribute Set

MeasuredValue Attribute

MeasuredValue represents the relative humidity in % as follows:
MeasuredValue = 100 x Relative humidity

Where 0% <= Relative humidity <= 100%, corresponding to a MeasuredValue in the range 0 to 0x2710.

The maximum resolution this format allows is 0.01%.

A MeasuredValue of 0xffff indicates that the measurement is invalid.

MinMeasuredValue Attribute

The MinMeasuredValue attribute indicates the minimum value of MeasuredValue that can be measured. A value of 0xffff means this attribute is not defined

MaxMeasuredValue Attribute

The MaxMeasuredValue attribute indicates the maximum value of MeasuredValue that can be measured. A value of 0xffff means this attribute is not defined

MaxMeasuredValue shall be greater than MinMeasuredValue. MinMeasuredValue and MaxMeasuredValue define the range of the sensor

Tolerance Attribute

The Tolerance attribute indicates the magnitude of the possible error that is associated with MeasuredValue. The true value is located in the range (MeasuredValue – Tolerance) to (MeasuredValue + Tolerance).

9.2.5.7. Occupancy Sensing Group

To be completed

9.2.6. Security and safety Functional Domain Attribute Sets

9.2.6.1. IAS Zone Group

To be completed

9.2.6.2. IAS ACE Group

To be completed

9.2.6.3. IAS WD Group

To be completed

9.2.7. Protocol interfaces Functional Domain Attribute Sets

9.2.7.1. Generic Tunnel Group

To be completed

9.2.7.2. BACnet Protocol Tunnel Group

To be completed

9.2.7.3. Analog Input (BACnet Regular) Group

To be completed

9.2.7.4. Analog Input (BACnet Extended) Group

To be completed

9.2.7.5. Analog Output (BACnet Regular) Group

To be completed

9.2.7.6. Analog Output (BACnet Extended) Group

To be completed

9.2.7.7. Analog Value (BACnet Regular) Group

To be completed

9.2.7.8. Analog Value (BACnet Extended) Group

To be completed

9.2.7.9. Binary Input (BACnet Regular) Group

To be completed

9.2.7.10. Binary Input (BACnet Extended) Group

To be completed

9.2.7.11. Binary Output (BACnet Regular) Group

To be completed

9.2.7.12. Binary Output (BACnet Extended) Group

To be completed

9.2.7.13. Binary Value (BACnet Regular) Group

To be completed

9.2.7.14. Binary Value (BACnet Extended) Group

To be completed

9.2.7.15. Multistate Input (BACnet Regular) Group

To be completed

9.2.7.16. Multistate Input (BACnet Extended) Group

To be completed

9.2.7.17. Multistate Output (BACnet Regular) Group

To be completed

9.2.7.18. Multistate Output (BACnet Extended) Group

To be completed

9.2.7.19. Multistate Value (BACnet Regular) Group

To be completed

9.2.7.20. Multistate Value (BACnet Extended) Group

To be completed

10. OEMan Status codes

On completion of a exchange (successful or not), the appropriate status code found in Table 10.1 should be returned.

Enumerated Status	Value	Description
SUCCESS	0x00	Operation was successful.
FAILURE	0x01	Operation was not successful.
-	0x02 – 0x7d	Reserved.
NOT_AUTHORIZED	0x7e	The sender of the command does not have authorization to carry out this command.
RESERVED_FIELD_NOT_ZERO	0x7f	A reserved field/subfield/bit contains a non-zero value.
MALFORMED_COMMAND	0x80	The command appears to contain the wrong fields, as detected either by the presence of one or more invalid field entries or by there being missing fields. Command not carried out. Implementer has discretion as to whether to return this error or INVALID_FIELD.
UNSUP_CLUSTER_COMMAND	0x81	The specified cluster command is not supported on the device. Command not carried out.
UNSUP_GENERAL_COMMAND	0x82	The specified general ZCL command is not supported on the device.
UNSUP_MANUF_CLUSTER_COMMAND	0x83	A manufacturer specific unicast, cluster specific command was received with an unknown manufacturer code, or the manufacturer code was recognized but the command is not supported.
UNSUP_MANUF_GENERAL_COMMAND	0x84	A manufacturer specific unicast, ZCL specific command was received with an unknown manufacturer code, or the manufacturer code was recognized but the command is not supported.
INVALID_FIELD	0x85	At least one field of the command contains an incorrect value, according to the specification the device is implemented to.
UNSUPPORTED_ATTRIBUTE	0x86	The specified attribute does not exist on the device.
INVALID_VALUE	0x87	Out of range error, or set to a reserved value. Attribute keeps its old value. Note that an attribute value may be out of range if an attribute is related to another, e.g. with minimum

		and maximum attributes. See the individual attribute descriptions for specific details.
READ_ONLY	0x88	Attempt to write a read only attribute.
INSUFFICIENT_SPACE	0x89	An operation (e.g. an attempt to create an entry in a table) failed due to an insufficient amount of free space available.
DUPLICATE_EXISTS	0x8a	An attempt to create an entry in a table failed due to a duplicate entry already being present in the table.
NOT_FOUND	0x8b	The requested information (e.g. table entry) could not be
UNREPORTABLE_ATTRIBUTE	0x8c	Periodic reports cannot be issued for this attribute.
INVALID_DATA_TYPE	0x8d	The data type given for an attribute is incorrect. Command not carried out.
INVALID_SELECTOR	0x8e	The selector for an attribute is incorrect.
WRITE_ONLY	0x8f	A request has been made to read an attribute that the requestor is not authorized to read. No action taken.
INCONSISTENT_STARTUP_STATE	0x90	Setting the requested values would put the device in an inconsistent state on startup. No action taken.
DEFINED_OUT_OF_BAND	0x91	An attempt has been made to write an attribute that is present but is defined using an out-of-band method and not over the air.
INCONSISTENT	0x92	The supplied values (e.g. contents of table cells) are inconsistent.
ACTION_DENIED	0x93	The credentials presented by the device sending the command are not sufficient to perform this action.
TIMEOUT	0x94	The exchange was aborted due to excessive response time.
ABORT	0x95	Failed case when a client or a server decides to abort the upgrade process.
INVALID_IMAGE	0x96	Invalid OTA upgrade image (ex. failed signature

		validation or signer information check or CRC check).
WAIT_FOR_DATA	0x97	Server does not have data block available yet.
NO_IMAGE_AVAILABLE	0x98	No OTA upgrade image available for a particular client.
REQUIRE_MORE_IMAGE	0x99	The client still requires more OTA upgrade image files in order to successfully upgrade.
-	0x9a– 0xbf	Reserved
HARDWARE_FAILURE	0xc0	An operation was unsuccessful due to a hardware failure.
SOFTWARE_FAILURE	0xc1	An operation was unsuccessful due to a software failure.
CALIBRATION_ERROR	0xc2	An error occurred during calibration.
-	0xc3 – 0xff	Reserved.