Macrotech HVAC Controllers

Use cases and Message Payloads

Last updated on: 22 Aug 2022

Table of Contents

[**Introduction**](#_39p6cl3ktedd) **3**

[**Messages - Packet Structures and JSON Payload**](#_fxgs24v28ydn) **3**

[**Message Data Types**](#_dt4z0jwzctcb) **5**

[**Device Id - Mapping**](#_pf0dq7xwksic) **7**

[**Use cases**](#_3phfkf2mvbwx) **7**

[Get Status - Individual HVAC Unit](#_sujl27bi7b86) 7

[JSON Payload - Cloud IoT Server to IoT Gateway](#_l379482sb34g) 7

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_wb3evhharqgh) 8

[Response for Get Status - Individual HVAC](#_u0n591h1njzo) 8

[PDU Structure - HVAC controller node to IoT Gateway](#_ck4xwk76eej) 8

[JSON Payload - IoT Gateway to Cloud IoT Server](#_4a8d5dudq0g) 9

[Get Status - From all HVAC Controllers](#_a4ebgegb1zsu) 9

[JSON Payload - Cloud IoT Server to IoT Gateway](#_1n7zurdhpmga) 9

[PDU Structure - IoT Gateway to all HVAC controller nodes](#_csarwuntnwug) 10

[Response for Get Status - From all HVAC Controllers](#_4i85hssd2g0k) 10

[PDU Structure - HVAC controller node to IoT Gateway](#_yrw047nsb2ah) 10

[JSON Payload - IoT Gateway to Cloud IoT Server](#_rsh9r3tefhej) 10

[Enable Heating Mode](#_k5j4bnby1avx) 10

[JSON Payload - Cloud IoT Server to IoT Gateway](#_7ghm4j11fp8b) 10

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_zi4ev4dfgu8n) 11

[Response for Enable Heating Mode](#_6vuicsvk4hjn) 11

[PDU Structure - HVAC controller node to IoT Gateway](#_w4q3pdlfx4kg) 11

[JSON Payload - IoT Gateway to Cloud IoT Server](#_r7e7tsloheh5) 11

[Enable Cooling Mode](#_20l8dipvshvc) 12

[JSON Payload - Cloud IoT Server to IoT Gateway](#_kue4g5t9cetd) 12

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_pmrorfy743go) 12

[Response for Enable Cooling Mode](#_kd40gekc4qhg) 13

[PDU Structure - HVAC controller node to IoT Gateway](#_ojdnjx7yujqz) 13

[JSON Payload - IoT Gateway to Cloud IoT Server](#_h8oxih89tpl5) 13

[Enable Fan Mode](#_ge9l84kix45m) 14

[JSON Payload - Cloud IoT Server to IoT Gateway](#_b7796skez7d) 14

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_g46606r8p6uz) 14

[Response for Enable Fan Mode](#_ozjsocfwyht) 15

[PDU Structure - HVAC controller node to IoT Gateway](#_q79va5565l1n) 15

[JSON Payload - IoT Gateway to Cloud IoT Server](#_iqfdp4imzbz4) 15

[Enable Heating and Fan Modes](#_6tw3ks4v51v) 16

[JSON Payload - Cloud IoT Server to IoT Gateway](#_15um5uoxv1cc) 16

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_43236u4gx3sb) 16

[Response for Enable Heating and Fan Modes](#_3latc5n7dtoq) 16

[PDU Structure - HVAC controller node to IoT Gateway](#_1i0t1pz8v1tb) 16

[JSON Payload - IoT Gateway to Cloud IoT Server](#_e13gb0nxopvb) 17

[Enable Cooling and Fan Modes](#_nef4nmdgbhv6) 17

[JSON Payload - Cloud IoT Server to IoT Gateway](#_epwvrr2dip7z) 17

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_x2mdghn07w03) 18

[Response for Enable Cooling and Fan Modes](#_kd5we2wc46zs) 18

[PDU Structure - HVAC controller node to IoT Gateway](#_j33r4y90d3ia) 18

[JSON Payload - IoT Gateway to Cloud IoT Server](#_2vcj2o2absdq) 18

[Set Temperature](#_r6jbt9w7hc2z) 19

[JSON Payload - Cloud IoT Server to IoT Gateway](#_727887mqyono) 19

[PDU Structure - IoT Gateway to a specific HVAC controller node](#_kh0d7ugrr3eg) 20

[Response for Set Temperature](#_fzwqyrigy0m5) 20

[PDU Structure - HVAC controller node to IoT Gateway](#_315n5upbhtcq) 20

[JSON Payload - IoT Gateway to Cloud IoT Server](#_mwru4bkm9dhp) 20

[**Appendix A - Wirepas Node Address and Device Id mapping CSV file**](#_tnxjvkvd5h8b) **21**

# Introduction

This document lists various use cases pertaining to managing the RF network of Macrotech’s HVAC controllers along with the corresponding message exchanging payload structures between:

* Cloud IoT Platform (Macrotech’s CloudExt) <-> IoT Gateway
* IoT Gateway <-> Wirepas Sink node <-> HVAC controller nodes

# Messages - Packet Structures and JSON Payload

Between Cloud IoT Platform and IoT Gateway, the messages are exchanged in JSON payload using MQTT protocol.

Between IoT Gateway and HVAC controller nodes, the messages are exchanged as encoded packets of bytes via Wirepas Sink node. The structure of this encoded packets is given below:

| **Byte:Length** | **Field Name** | **Description** |
| --- | --- | --- |
| 0:1 | Beginning Marker | Beginning of Packet Marker. Should be 0xAA. |
| 1:1 | Message Type | First 2 Most Significant Bits (MSBs) specify destination address type:  01 - Unicast  10 - Multicast  11 - Broadcast  Next 3 bits specify the operation type:  000 - Get  001 - Set  010 - Notification  011 - Acknowledgement  Last 3 bits specify the type of data:  000 - Attribute  001 - State  010 - Telemetry  011 - Alarm |
| 2:4 or 1 or 0 | Source/Destination Address | * For messages sent from the individual node, the source address should be 4 bytes containing Wirepas node address of that individual node. * For messages sent from the sink node to an individual node, the destination address should be 4 bytes Wirepas node address of the target node. * For multicast messages sent from the sink node, this field is 1 byte containing the group number. * For broadcast messages sent from the sink node, this field should be absent (0 bytes). |
| 6 or 3 or 2:4 | Request ID | 4 bytes signed int used as the request identifier. The nodes should process the request id field in the following way:   * When a Get or Set type of packet is received, this request id should be sent back when sending the response for Get operation (Notification) or Set operation (Acknowledgement). This is very important as the gateway uses the request id to map the corresponding response it sends to Cloud IoT Server. * While sending Telemetry or Alarm messages, the request id should be -1. |
| 10 or 7 or 6:1 (optional) | Key and Length | First 4 Most Significant Bits (MSBs) specify the key of the data element.  With 4 bits, there can be 15 keys for each data type (Attribute, State, Telemetry and Alarm). Note that only 0x1 to 0xF are used (0x0 is not used).  Last 4 bits specify the length of the data element. With 4 bits, the data length can vary upto a maximum of 15 bytes. If no value is specified, then 0x0 is used. For example, in Get messages, only the key field is passed without any value. |
| 11 or 8 or 7:x | Value | If the data element’s length in the previous byte is non-zero, then the corresponding value is captured in N number of bytes, where N = Length as specified in the previous byte. |
| x:1 | End Marker | End of Packet Marker. Should be 0x00 (for this reason, 0x0 is not used for the key).  The packet may contain more than 1 data element and the end marker is used to mark the end of Key-Length-Value tuples. |

# Message Data Types

The following tables provide the list of supported HVAC controller data types (Attribute, State, Alarm and Telemetry) in HVAC controller nodes (Wirepas RF mesh) and Cloud IoT Server (CloudExt IoT Platform).

| **Attribute** | | | |
| --- | --- | --- | --- |
| Device ID | Wirepas specific data type | Key | **0x10** |
| Data type | **Byte** |
| Size | **6** |
| Type | **Read-only** |
| IoT Server specific data type | ***macaddr*** - String in the form of xx-xx-xx-xx-xx-xx | |
| This field provides mapping of Device to it's hardware ID. Useful to map server side name to device side ID. Last 3 bytes of this ID are used by Wirepas as a Node address. | | |
| Firmware Version | Wirepas specific data type | Key | **0x20** |
| Data type | **Byte** |
| Size | **2** |
| Type | **Read-only** |
| IoT Server specific data type | ***verstr*** - String in the form {major}.{minor}.{maint}.  For example: 1.4.21, 1.0.00 | |
| Wirepas side:  1st byte contains 4 bits major version no. and 4 bits minor version number, and 2nd byte contains maintenance version number. Example 0x10 0x01 means version 1.0.1 | | |
| Location | Wirepas specific data type | Key | **0x30** |
| Data type | **String** |
| Size | **15** |
| Type | **Read Write** |
| IoT Server specific data type | String.  For example: “F1-101-HVAC” | |
| This field contains free form text upto 15 characters long. It is used for uniquely naming the device to identify it’s geographical location. | | |
| Telemetry Sampling Interval | Wirepas specific data type | Key | **0x40** |
| Data type | **Short Integer** |
| Size | **2** |
| Type | **Read Write** |
| IoT Server specific data type | Integer | |
| Represents the telemetry sampling interval (periodicity) in seconds. For disabling telemetry, the value should be set 0. | | |

| **State** | | | |
| --- | --- | --- | --- |
| mode | Wirepas specific data type | Key | **0x10** |
| Data type | **Byte** |
| Size | **1** |
| Type | **Read Write** |
| IoT Server specific data type | String | |
| Used for turning on various modes of the HVAC unit.  To turn on/enable various modes, the following byte value should be used:   * Heating - 0x01 * Cooling - 0x02 * Fan - 0x03 * Heating and Fan - 0x04 * Cooling and Fan - 0x05 | | |
| temperature | Wirepas specific data type | Key | **0x20** |
| Data type | **Byte** |
| Size | **1** |
| Type | **Read Write** |
| IoT Server specific data type | Unsigned Short Integer between 0 and 255 | |
| Used for configure the temperature setting on the HVAC unit between 0 and 255 Fahrenheit. | | |

# Device Id - Mapping

Each HVAC controller node is mapped with a unique identifier. In Wirepas RF mesh network, the HVAC controller node uses *Wirepas node address*, which is a 6 byte integer value.

In CloudExt IoT Platform, each controller node is identified with a human-readable string instead of *Wirepas node address*.

The IoT Gateway performs the device id translation (Wirepas node address to CloudExt human-readable string and vice-versa) while exchanging the messages between Wirepas RF mesh network and Cloud IoT Platform.

So, it is important to provide the mapping file (CSV) containing Wirepas node address mapped to CloudExt human-readable string before deploying/starting IoT Gateway.

See [Appendix A](#_4zqfbezibtzq) for an example mapping CSV file.

# Use cases

**Note:**

* Device Id mentioned in MQTT topics represents the human-readable string used in CloudExt IoT Platform to represent a HVAC controller node.
* Node Address mentioned in PDU Structure represents the wirepas node address of the corresponding HVAC controller node in Wirepas RF mesh network.

## Get Status - Individual HVAC Unit

Cloud IoT Server initiates this operation by sending a command to get the status of a specific HVAC unit in the Wirepas RF mesh network to which the gateway is connected.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**get-hvac-status**”, “Seq” : <integer> } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the status should be fetched from that HVAC controller.  The Seq field contains an integer that should be returned back as is in the response payload. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller nodes for Get-State Endpoint (EP), which is **0x01**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x10 | Key = mode, Length = 0 |
| 6:1 | 0x20 | Key = temperature, Length = 0 |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Get Status - Individual HVAC

### PDU Structure - HVAC controller node to IoT Gateway

The following PDU is sent from the HVAC controller node on Notification-State Endpoint (EP) for the destination, which is **0x11**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the *Get Status* request message. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x01 or 0x02 or 0x03 or 0x04 or 0x05 | Configure mode of operation. 0x01 means Heating, 0x02 - Cooling, 0x03 - Fan, 0x04 - Heating and Fan, 0x05 - Cooling and Fan. |
| 7:1 | 0x21 | Key = temperature, Length = 1 |
| 8:1 |  | Byte. Represents the current temperature setting on the HVAC unit between 0 and 255 |
| 9:1 | 0x00 | End of Packet Marker |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for Get Status request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**get-hvac-status**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  "mode" : “heating” | “cooling” | “fan” | “heating and fan” | “cooling and fan”,  "temperature” : <unsigned int>  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains multiple fields:   * *mode* - can be heating, cooling, fan, heating and fan, cooling and fan * *temperature* - unsigned integer. |

### Get Status - From all HVAC Controllers

The following MQTT message is sent from Cloud IoT Server to the gateway for querying the status from all HVAC controllers in the RF mesh network associated with the gateway:

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Gateway Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**get-hvac-status**”, “Seq” : <integer> } |
| **Description** | In the MQTT topic, **{Gateway ID}** is set with the user-readable string as listed in CloudExt’s UI (for example: Macrotech\_GW\_1) so that the command is broadcasted to all HVAC controllers in the mesh network associated with the gateway.  The Seq field contains an integer that should be returned back as is in the response payload. The Seq id is used by the IoT server for tracking the requests.  **NOTE:**   * When this command is sent to the gateway, the gateway shall take care of broadcasting this command to all HVAC controllers in the given mesh network. * CloudExt IoT broker/server would receive as many responses as the number of active HVAC controllers in the given mesh network. All such responses carry the same Seq number sent in the command request. |

### PDU Structure - IoT Gateway to all HVAC controller nodes

IoT Gateway sends the PDU as given in [PDU Structure - IoT Gateway to a specific HVAC controller node](#_wb3evhharqgh) to all HVAC controller nodes in the mesh network. It should be noted that the same sequence number is used in the PDU.

## Response for Get Status - From all HVAC Controllers

### PDU Structure - HVAC controller node to IoT Gateway

Each HVAC controller node sends the PDU as given in [Response for Get Status - Individual HVAC](#_ck4xwk76eej).

### JSON Payload - IoT Gateway to Cloud IoT Server

The MQTT response message as given in [Response for Get Status - Individual HVAC](#_4a8d5dudq0g) is sent from IoT Gateway to Cloud IoT Server for every response received from the HVAC controllers in the mesh network:

## Enable Heating Mode

Cloud IoT Server initiates this operation to enable heating mode on a specific HVAC unit.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**enable-heating**”, “Seq” : <integer> } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the heating mode should be enabled on the corresponding HVAC unit.  The Seq field contains an integer that should be returned back as is in the response payload. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller node for Set-State Endpoint (EP), which is **0x09**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x01 | 0x01 means enable the heating mode |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Enable Heating Mode

### PDU Structure - HVAC controller node to IoT Gateway

The HVAC controller node sends the following response for enable heating request on Acknowledgement-State Endpoint (EP) for the destination, which is **0x19**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the enable heating request message. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 7:1 | 0x00 | End of Packet Marker. |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for enable heating request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**enable-heating**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “status” : “success”  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains a status field, which can be either “success” or “failed”. |

## Enable Cooling Mode

Cloud IoT Server initiates this operation to enable cooling mode on a specific HVAC unit.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**enable-cooling**”, “Seq” : <integer> } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the cooling mode should be enabled on the corresponding HVAC unit.  The Seq field contains an integer that should be returned back as is in the response payload. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller node for Set-State Endpoint (EP), which is **0x09**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x02 | 0x02 means enable the cooling mode |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Enable Cooling Mode

### PDU Structure - HVAC controller node to IoT Gateway

The HVAC controller node sends the following response for enable colling request on Acknowledgement-State Endpoint (EP) for the destination, which is **0x19**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the enable cooling request message. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 7:1 | 0x00 | End of Packet Marker. |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for enable colling request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**enable-cooling**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “status” : “success”  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains a status field, which can be either “success” or “failed”. |

## Enable Fan Mode

Cloud IoT Server initiates this operation to enable fan mode on a specific HVAC unit.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**enable-fan**”, “Seq” : <integer> } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the fan mode should be enabled on the corresponding HVAC unit.  The Seq field contains an integer that should be returned back as is in the response payload. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller node for Set-State Endpoint (EP), which is **0x09**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x03 | 0x03 means enable the fan mode |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Enable Fan Mode

### PDU Structure - HVAC controller node to IoT Gateway

The HVAC controller node sends the following response for enable fan request on Acknowledgement-State Endpoint (EP) for the destination, which is **0x19**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the enable fan request message. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 7:1 | 0x00 | End of Packet Marker. |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for enable fan request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**enable-fan**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “status” : “success”  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains a status field, which can be either “success” or “failed”. |

## Enable Heating and Fan Modes

Cloud IoT Server initiates this operation to enable heating and fan modes on a specific HVAC unit.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**enable-heating-fan**”, “Seq” : <integer> } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the heating and the fan modes should be enabled on the corresponding HVAC unit.  The Seq field contains an integer that should be returned back as is in the response payload. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller node for Set-State Endpoint (EP), which is **0x09**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x04 | 0x04 means enable the heating and the fan mode |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Enable Heating and Fan Modes

### PDU Structure - HVAC controller node to IoT Gateway

The HVAC controller node sends the following response for enable heating and fan request on Acknowledgement-State Endpoint (EP) for the destination, which is **0x19**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the enable heating and fan request message. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 7:1 | 0x00 | End of Packet Marker. |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for enable heating and fan request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**enable-heating-fan**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “status” : “success”  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains a status field, which can be either “success” or “failed”. |

## Enable Cooling and Fan Modes

Cloud IoT Server initiates this operation to enable cooling and fan modes on a specific HVAC unit.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | { “Command” : “**enable-cooling-fan**”, “Seq” : <integer> } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the cooling and the fan modes should be enabled on the corresponding HVAC unit.  The Seq field contains an integer that should be returned back as is in the response payload. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller node for Set-State Endpoint (EP), which is **0x09**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x05 | 0x05 means enable the cooling and the fan mode |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Enable Cooling and Fan Modes

### PDU Structure - HVAC controller node to IoT Gateway

The HVAC controller node sends the following response for enable cooling and fan request on Acknowledgement-State Endpoint (EP) for the destination, which is **0x19**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the enable cooling and fan request message. |
| 5:1 | 0x11 | Key = mode, Length = 1 |
| 6:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 7:1 | 0x00 | End of Packet Marker. |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for enable cooling and fan request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**enable-cooling-fan**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “status” : “success”  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains a status field, which can be either “success” or “failed”. |

## Set Temperature

Cloud IoT Server initiates this operation to configure the temperature setting on a specific HVAC unit.

### JSON Payload - Cloud IoT Server to IoT Gateway

| **MQTT Topic** | {Device Id}/CMD |
| --- | --- |
| **JSON Payload** | {  "Command" : "**set\_temperature**",  "Seq" : <integer>,  "Params" : {  "temperature" : <unsigned int>  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the temperature setting should be configured on the corresponding HVAC unit.  The Seq field contains an integer that should be returned back as is in the response payload.  The params field contains the following parameters:  (1) temperature, which is an unsigned integer representing the value of the temperature to be configured in Fahrenheit in the range 0 and 255. |

### PDU Structure - IoT Gateway to a specific HVAC controller node

The following PDU is sent to HVAC controller node for Set-State Endpoint (EP), which is **0x09**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This is actually the value received in the request payload for the **Seq** field. |
| 5:1 | 0x21 | Key = temperature, Length = 1 |
| 6:1 |  | 1 byte. Represents temperature in Fahrenheit in the range 0 and 255. |
| 7:1 | 0x00 | End of Packet Marker |

## Response for Set Temperature

### PDU Structure - HVAC controller node to IoT Gateway

The HVAC controller node sends the following response for set temperature request on Acknowledgement-State Endpoint (EP) for the destination, which is **0x19**:

| **Byte:Length** | **Field Value** | **Description** |
| --- | --- | --- |
| 1:4 | <Req Id> | Request ID. Signed int (4 bytes). This request id must be the request id that was sent in the enable cooling and fan request message. |
| 5:1 | 0x21 | Key = temperature, Length = 1 |
| 6:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 7:1 | 0x00 | End of Packet Marker. |

### JSON Payload - IoT Gateway to Cloud IoT Server

The following MQTT response message is sent from IoT Gateway to Cloud IoT Server for set temperature request:

| **MQTT Topic** | {Device Id}/CMD\_RESP |
| --- | --- |
| **JSON Payload** | {  “Command” : “**set-temperature**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “status” : “success”  }  } |
| **Description** | In MQTT topic, {Device Id} is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-HVAC), indicating that the response corresponds to that HVAC controller.  The JSON payload contains three fields: command, seq, and response.  The command field contains the name of the command for which the response is associated.  The seq (sequence) field contains the integer that was received in the corresponding command request’s seq field.  The response object contains a status field, which can be either “success” or “failed”. |

# Appendix A - Wirepas Node Address and Device Id mapping CSV file

Here is an example file showing the mapping of Wirepas Node Address and Device Id:

| name,address,type,remarks  F1-R101-HVAC,0xDFD81E,HVAC,"HVAC in 1st Floor, Room 101"  F8-R812-HVAC,0xB56FCD,HVAC,"HVAC in 8th Floor, Room 812"  F1-R102-HVAC,0x38E823,HVAC,"HVAC in 1st Floor, Room 102" |
| --- |

The mapping CSV file contains the following columns/fields:

* name - The name of the HVAC controller as identified in CloudExt IoT Platform UI. It corresponds to **Device Id**. It is expected to be in human-readable format, for example, F8-R812-HVAC, representing the HVAC unit in Room 812 on the 8th floor.
* address - The address assigned to the HVAC controller in Wirepas RF mesh network.
* type - Type of the node. For HVAC controllers, it should be **HVAC**.
* remarks - Space for writing free form text, which can be used, if required, to include additional information. This field is optional.