HVAC

Problem Statement:-

Mode= 0Off, 1=Cool, 2=Heat, 3=Auto

Fan= 0=Off, 1=On

SetTemp=60

Single Command to update all settings together:-

{

"device=AC101",

"mode":1,

"fan":0,

"settemp":60

}

Whenever we want to update any settings there will be only one command. no individual command.

HVAC STATUS and whenever any changes made device will send updated status in /Notification/{"mode"=1,"fan":1,"settemp":60,"room temp":64}

Notes:

1. whenever Temp, or mode change thermostat will post current status

2. Every 30 minutes the thermostat will post the current status as a heartbeat

3. Whenever we send {Get Status} command it will post the current status

"HVAC" : {

"Attribute" : [

{

"name" : "Device ID",

"key" : "0x10",

"is\_writeable" : false,

"size" : 6,

"type" : "macaddr"

},

{

"name" : "Firmware Version",

"key" : "0x20",

"is\_writeable" : false,

"size" : 2,

"type" : "verstr"

},

{

"name" : "Location",

"key" : "0x30",

"is\_writeable" : true,

"size" : 15,

"type" : "string"

},

{

"name" : "Telemetry Sampling Interval",

"key" : "0x40",

"is\_writeable" : true,

"size" : 2,

"type" : "u\_short"

}

],

"State" : [

{

"name" : "mode",

"key" : "0x10",

"is\_writeable" : true,

"size" : 2,

"type" : "short”

},

{

"name" : "temperature",

"key" : "0x20",

"is\_writeable" : true,

"size" : 2,

"type" : "short"

},

{

"name" : "room\_temperature",

"key" : "0x30",

"is\_writeable" : false,

"size" : 2,

"type" : "short"

},

{

"name" : "fan",

"key" : "0x40",

"is\_writeable" : true,

"size" : 1,

"type" : "bool"

}

],

"Alarm" : [],

"Telemetry" : [

{

"name" : "mode",

"key" : "0x10",

"is\_writeable" : false,

"size" : 2,

"type" : "short”

},

{

"name" : "temperature",

"key" : "0x20",

"is\_writeable" : false,

"size" : 2,

"type" : "short"

},

{

"name" : "room\_temperature",

"key" : "0x30",

"is\_writeable" : false,

"size" : 2,

"type" : "short"

},

{

"name" : "fan",

"key" : "0x40",

"is\_writeable" : true,

"size" : 1,

"type" : "bool"

}

]

}

|  |  |
| --- | --- |
| **MQTT Topic** | {Device Id}/CMD |
| **JSON Payload** | {  “Command” : “**set**”,  “Seq” : <integer>,  “Params” : {  "mode" : <short> 0-off, 1-cool, 2-Heat, 3-Auto  "temperature” : <unsigned int>,  “fan”: <bool> 0-Off 1-On  } |
| **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 params.  The command field contains the name of the command for which the response is associated.  The Seq field contains an integer that should be returned back as is in the response payload.  The params object contains multiple fields:   * *mode* - can be heating, cooling, auto and off * *temperature* - unsigned integer. * *Fan* – on and off |

|  |  |  |
| --- | --- | --- |
| **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 | Configure mode of operation. 0x01 means Heating, 0x02 - Cooling, 0x03 - Auto, 0x04 - OFF. |
| 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 | 0x41 | Key = fan, Length = 1 |
| 10:1 | 0x0 or 0x1 | Configure Fan operation. 0x1 means ON and 0x0 means OFF |
| 11:1 | 0x00 | End of Packet Marker |

|  |  |  |
| --- | --- | --- |
| **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 | 0x21 | Key = temperature, Length = 1 |
| 8:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 9:1 | 0x41 | Key = fan, Length = 1 |
| 10:1 | 0x0 or 0x1 | Error code. 0x0 means a successful operation. 0x1 indicates a failed operation. |
| 11:1 | 0x00 | End of Packet Marker. |

|  |  |
| --- | --- |
| **MQTT Topic** | {Device Id}/CMD\_RESP |
| **JSON Payload** | {  “Command” : “**set**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  “mode\_status” : “success”,  “temperature\_status” : “success”,  “fan\_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”. |

|  |  |
| --- | --- |
| **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. |

|  |  |  |
| --- | --- | --- |
| **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 | 0x30 | Key = room\_temperature, Length = 0 |
| 8:1 | 0x40 | Key = fan, Length = 0 |
| 9:1 | 0x00 | End of Packet Marker |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | | Configure mode of operation. 0x01 means Heating, 0x02 - Cooling, 0x03 - OFF, 0x04 - Auto. |
| 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 | 0x31 | | Key = room\_temperature, Length = 1 |
| 10:1 |  | | Byte. Represents the current temperature setting on the HVAC unit between 0 and 255 |
| 11:1 | 0x41 | | Key = fan, Length = 1 |
| 12:1 | 0x0 or 0x1 | | Configure Fan operation. 0x1 means ON and 0x0 means OFF |
| 13:1 | 0x00 | | End of Packet Marker |
|  | | | |
| **MQTT Topic** | | {Device Id}/CMD\_RESP | |
| **JSON Payload** | | {  “Command” : “**get-hvac-status**”,  “Seq” : <integer that was received in the request message>,  “Response” : {  "mode" : <short> 0-off, 1-cool, 2-Heat, 3-Auto  "temperature” : <unsigned int>,  "room\_temperature” : <unsigned int>,  “fan”: “fan\_on” | “fan\_off”  }  } | |
| **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, off and auto * *temperature* - unsigned integer. * *Room temperature* - unsigned integer. * *Fan* – on and off | |