# Keypad verification steps

* Gateway will subscribe to the MQTT topic {Gateway Id}/CMD (This is just an example, need to discuss on proper topic name)
* Gateway will get device ID from MQTT payload
* Gateway will convert device ID into Wirepas Node Address as per the CSV details
* Message (Keypad-key) will then either broadcast in WNS along with mac-address or will directly send to that particular Wirepas Node Address

MQTT Detail: -

JSON payload to set the Keypad-code: -

|  |  |
| --- | --- |
| MQTT Topic | **{Device ID}/CMD** |
| JSON Payload | {    "Command": "**set\_keypad\_code**",  "Seq": <integer>,  "Params": {  "key": <unsigned int>  }  } |
| Description | The JSON payload contains three fields: Device id, key  **{Device ID}** is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-Lock)  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.  The **{params}** field contains the following parameters:  **{key}** Keypad-key assigned to user to unlock rooms/doors. |

JSON payload to send response of set keypad-code: -

|  |  |
| --- | --- |
| MQTT Topic | **{Device ID}/CMD\_RESP** |
| JSON Payload | {  **"**Command": **"set\_keypad\_code**",  "Seq": <integer that was received in the request message>,  **"**Response": {  "status": "success",  }  } |
| Description | In the MQTT topic, **{Device ID}** is set with the user-readable string as listed in CloudExt’s UI (for example: F1-R101-Lock), indicating that the JSON payload is coming from that lock 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 status field in the response can be either “success” or “failed” |

JSON payload to send Alarm Response: -

|  |  |
| --- | --- |
| MQTT Topic | **{Device ID}/ALARM** |
| JSON Payload | { unlock-by-keypad-alert : true | false } |
| Description | unlock-by-keypad-alert can be either true (indicates door successfully unlocked) or false (failed to unlock the door). |

PDU details:-

From the following table first four column fields will be the highlighted fields and next four column fields will be as per the PDU Structure - IoT Gateway to a specific Lock controller node.

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

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

To set Keypad-Code: -

PDU Structure - IoT Gateway to a specific Lock controller node

The following PDU is sent to Lock controller nodes 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 |  | Key = keypad\_code, Length = 4 |
| 6:4 |  | 4 bytes. Unsigned Integer. Represents the keypad code stored in little-endian form. The lock controller should ensure the keypad\_code value is in the range 1000 to 9999 (4 digits). |
| 10:1 | 0x00 | End of Packet Marker |

Response for Set Keypad code

PDU Structure - Lock controller node to IoT Gateway

The following PDU is sent from the Lock controller node 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 Set Admin Key request message. |
| 5:1 | 0x91 | Key = admin\_key, 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. |















Notification-Alarm on door unlock: -

PDU Structure - Lock controller node to IoT Gateway

The lock controller node sends the following alarm message to IoT Gateway on Notification-Alarm Endpoint (EP) for the destination, which is 0x13:

|  |  |  |
| --- | --- | --- |
| **Byte:Length** | **Field Value** | **Description** |
| 1:4 | <Req Id> | Request ID. For alarm messages, the request must be set to -1. |
| 5:1 | 0x51 | Key = unlock-by-keypad-key-alert, Length = 1 |
| 6:1 |  | 0x1 means the door unlocked successfully and 0x0 means failure. |
|  |  |  |
|  |  |  |
| 7:1 | 0x00 | End of Packet Marker |

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-Lock,0xDFD81E,Lock,"Lock in 1st Floor, Room 101"  F8-R812-Lock,0xB56FCD,Lock,"Lock in 8th Floor, Room 812"  F1-R102-Lock,0x38E823,Lock,"Lock in 1st Floor, Room 102" |

The mapping CSV file contains the following columns/fields:

* name - The name of the lock 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-Lock, representing the lock in Room 812 on the 8th floor.
* address - The address assigned to the lock controller in Wirepas RF mesh network.
* type - Type of the node. For lock controllers, it should be **Lock**.
* remarks - Space for writing free form text, which can be used, if required, to include additional information. This field is optional.