Macrotech Lock Controllers

Requirements and MQTT References

11 Feb 2022

# Introduction

This document captures the requirements for the interworking of IoT gateway and Macrotech’s CloudExt IoT Platform and the proposal from LiteNet Technology for MQTT references required for the interworking.

The document is updated to incorporate the changes required by CloudExt IoT Platform for handling MQTT messages.

# Gateway Requirements

Gateway serves as the middleware between Lock controller nodes in RF mesh (Wirepas mesh) network and Cloud hosted IoT Server (Macrotech’s CloudExt IoT Platform). The gateway performs the data/payload translation for exchanging the information between Lock controller nodes and the IoT server.

Here is the list of broad requirements for the gateway as a middleware:

1. Enable Cloud IoT Server to query Lock controller nodes to get the status of the lock. The query can be either for an individual lock controller or for all lock controllers in the mesh network associated with the gateway.
2. Enable Cloud IoT Server to unlock door (opening the latch) in case of normal operation (when the deadbolt is not engaged)
3. Enable Cloud IoT Server to unlock door during emergency (when the deadbolt is engaged)
4. Send periodic heartbeat messages (frequency is configurable) of all lock controllers to Cloud IoT Server (telemetry/heartbeat message)
5. Send an alert/alarm to Cloud IoT Server when the lock controller battery level drops below the minimum operating value (the exact value is determined by the firmware running on the lock controller nodes).
6. Send an alert/alarm to Cloud IoT Server when the deadbolt is engaged or disengaged.
7. Send an alert/alarm to Cloud IoT Server when the latch is closed or opened.

# Gateway MQTT Reference for CloudExt IoT Server

This section contains the proposed MQTT reference for Gateway and CloudExt IoT Server for exchanging the data to fulfill the [gateway requirements](#_g60gpvff2fuz).

**Note 1:**

**MQTT topics and payload formats listed in this section are based on the Register Command document received from Macrotech on 4th Feb 2022.**

**Note 2:**

In the JSON payload exchanged between CloudExt IoT Server and Gateway, the field “Device Id” in MQTT topics corresponds to the unique identifier given for the lock controller. In the RF mesh (Wirepas mesh) network, MAC address is used as the node identifier, which poses a challenge in identifying the physical location of the corresponding lock. Assigning a unique human readable name like F1-R101-Lock (lock in Room 101 of Floor 1) or something similar is a better naming convention with regards to CloudExt’s user interface. So, the gateway shall maintain a mapping of RF mesh MAC address of human-readable unique name and use this mapping for translating the device id.

The mapping file (a CSV file) shall be provided to the gateway at the time of deployment/commissioning of the gateway.

## Registering lock controller nodes information with CloudExt IoT Server

The gateway supports the following MQTT topics for exchanging the data between lock controllers and CloudExt IoT server:

| **MQTT Topic** | **Type** | **Description** |
| --- | --- | --- |
| CMD | Subscription | Subscribes to this topic to receive commands from CloudExt IoT server for performing operations like unlocking, get status, … on the lock controllers. |
| NOTIFICATION | Publishing | The gateway publishes the telemetry data and one-time attribute data received from lock controllers. |
| ALARM | Publishing | The gateway publishes the alarms such as low-battery, deadbolt-alerts received from lock controllers. |
| CMD\_RESP | Publishing | The gateway publishes the command responses received from lock controllers. |

Accordingly, the gateway sends the following JSON registration message to CloudExt IoT server during the gateway startup:

| MQTT Topic | **system/{Gateway ID}/register/group** |
| --- | --- |
| JSON Payload | {  "group\_details": [  {  "device\_id" : "{Device ID} like F1-R101-Lock>",  "device\_name" : "<Name or can be device\_id>",  "command\_topics" : [  {  "topic" : "{Device ID}/CMD",  "qos" : "1"  }  ],  "event\_topics" : [  {  "topic" : "{Device ID}/ALARM",  "qos" : "1"  },  {  "topic" : "{Device ID}/CMD\_RESP",  "qos" : "1"  },  {  "topic" : "{Device ID}/NOTIFICATION",  "qos" : "1"  }  ]  },  {  "device\_id" : "<Another {Device ID} like F3-R304-Lock>",  "device\_name" : "<Name or can be device\_id>",  "command\_topics" : [  {  "topic" : "{Device ID}/CMD",  "qos" : "1"  }  ],  "event\_topics" : [  {  "topic" : "{Device ID}/ALARM",  "qos" : "1"  },  {  "topic" : "{Device ID}/CMD\_RESP",  "qos" : "1"  },  {  "topic" : "{Device ID}/NOTIFICATION",  "qos" : "1"  }  ]  },  ...  ]  } |
| Description | In the JSON payload, the **group\_details** array object contains the data (same data) for each lock controller in the given RF mesh network. If there are 100 lock controllers, then group\_details contains 100 elements in the array, each representing the individual lock controller. |

## Commands from Cloud IoT Server to Gateway



### Get Status - Individual Lock

The following MQTT message is sent from Cloud IoT Server for querying the status of a specific lock in the RF mesh network associated with the gateway:

| MQTT Topic | **{Device ID}/CMD** |
| --- | --- |
| JSON Payload | { “Command” : “**getstatus**” } |
| 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 status should be fetched from that lock controller. |

### Unlock Door - Normal Operation

The following MQTT message is sent from Cloud IoT Server for unlocking a door, when the deadbolt is not engaged from inside:

| MQTT Topic | **{Device ID}/CMD** |
| --- | --- |
| JSON Payload | { “Command” : “**unlock**” } |
| 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 corresponding door should be unlocked. |

### Unlock Door - Emergency Operation

The following MQTT message is sent from Cloud IoT Server for unlocking a door, when the deadbolt is engaged from inside :

| MQTT Topic | **{Device ID}/CMD** |
| --- | --- |
| JSON Payload | { “Command” : “**emunlock**” } |
| 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 corresponding door should be unlocked as part of emergency operation. |

## Command Responses from Gateway to Cloud IoT Server

### Response for Get Status

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

**Note:**

When CloudExt IoT Server sends a query to get the status of all locks, there is no single response containing the details of all locks. Instead, individual responses are sent for each lock.

| MQTT Topic | **{Device ID}/CMD\_RESP** |
| --- | --- |
| JSON Payload | {  “Command” : “**getstatus**”,  “Response” : {  “status” : “online”,  “battery” : 85,  “deadbolt” : false,  “latch” : false  }  } |
| 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 two fields: command and response.  The command field contains the name of the command for which the response is associated.  In the response object:   * The status field can be either “online” or “offline” (if the lock controller is down; currently, this is not supported) * The battery level can be between 0 to 100 (percentage) * For deadbolt, a boolean value of true/false is sent indicating whether the deadbolt is engaged (true) or disengaged (false). * For latch, a boolean value of true/false is sent indicating whether the latch is closed (true) or opened (false). |

### Response for Unlock Door - Normal Operation

The following MQTT message response is sent from Gateway to Cloud IoT Server for unlock door (normal operation) command:

| MQTT Topic | **{Device ID}/CMD\_RESP** |
| --- | --- |
| JSON Payload | {  “Command” : “**unlock**”,  “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 status field in the response can be either “success” or “failed” |

### Response for Unlock Door - Emergency Operation

The following MQTT message response is sent from Gateway to Cloud IoT Server for unlock door (emergency operation) command:

| MQTT Topic | **{Device ID}/CMD\_RESP** |
| --- | --- |
| JSON Payload | {  “Command” : **“emunlock**”,  “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 status field in the response can be either “success” or “failed” |

## Telemetry, Notification from Gateway to Cloud IoT Server

### Heartbeat Message

The following MQTT message is sent from Gateway to Cloud IoT Server periodically as part of lock status (heartbeat):

| MQTT Topic | **{Device ID}/NOTIFICATION** |
| --- | --- |
| JSON Payload | {  “status” : “online”,  “battery” : 85,  “deadbolt” : false,  “latch” : false  } |
| 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 status field can be either “online” or “offline” (if the lock controller is down; currently, this is not supported)  The battery level can be between 0 to 100 (percentage)  For deadbolt, a boolean value of true/false is sent indicating whether the deadbolt is engaged (true) or disengaged (false).  For latch, a boolean value of true/false is sent indicating whether the latch is closed (true) or opened (false). |



## Alerts/Alarms from Gateway to Cloud IoT Server

### Low-Battery Alert

The following MQTT message is sent from Gateway to Cloud IoT Server when a lock controller’s battery level drops below the minimum operating value (the exact value is determined by the firmware running on the lock controller nodes)

| MQTT Topic | {Device ID}/**ALARM** |
| --- | --- |
| JSON Payload | { “low-battery” : true } |
| Description |  |

### Deadbolt Alert

The following MQTT message is sent from Gateway to Cloud IoT Server the deadbolt is either engaged or disengaged.

| MQTT Topic | {Device ID}/**ALARM** |
| --- | --- |
| JSON Payload | { “deadbolt-alert” : true } |
| Description | deadbolt-alert can be either true (indicates deadbolt is engaged) or false (indicates deadbolt is disengaged). |

### Latch Alert

The following MQTT message is sent from Gateway to Cloud IoT Server the latch is either closed or opened.

| MQTT Topic | {Device ID}/**ALARM** |
| --- | --- |
| JSON Payload | { “latch-alert” : true } |
| Description | latch-alert can be either true (indicates latch is closed) or false (indicates latch is opened). |

## 