DALI PRO 2 IoT

MQTT

Light Control Quick Start Guide

Version 1.13.0

Table of Contents

1. Introduction

The present document is meant to help the adoption and usage of the MQTT interface that is part of the *DALI PRO 2 Cloud API*. This document provides some practical examples on how to use the MQTT interface for some common use cases.
The full API specification is available at (1).
More information of what a MQTT is can be found at (2)

2. Prerequisites

The following hardware and software need to be available in order to execute the step by steps examples in this guide.

2.1 Hardware

1. At least 1 DALI PRO 2 IoT
2. At least 1 DALI ballast and 1 DALI input device (e.g. motion sensor, push button, etc.)
3. A computer connected to the DALI PRO 2 IoT via LAN cable or Wi-Fi dongle

2.2 Software

4. *DALI Professional 3* software version 3.1.6 or higher (ask your sales representative to get the software installation package). This is the software needed to commission your lighting system.
5. *MQTT X* (https://mqttx.app/). This is the software needed to subscribe to the DALI PRO 2 MQTT topics.
6. *Mosquitto* (https://mosquitto.org/download/). This is the MQTT broker needed to relay messages sent by the DALI PRO 2 to MQTT X.
7. *Postman* (https://www.postman.com/downloads/). This is the software needed to issue REST calls to the DALI PRO 2.
   N.B.: *MQTT X*, *Mosquitto* and *Postman* are not the only software tools for MQTT and REST communication. There many others available, feel free to choose those you are more comfortable with.

3. Build up of a test system

3.1 Commission the system with *DALI Professional 3* software version 3.1.6 or higher.

8. This means, at minimum, creating a group of ballasts with one or more input devices connected (e.g. presence sensors, buttons, etc.) as depicted in
Figure 1.

Light control can only be performed at group level, not on a single ballast.

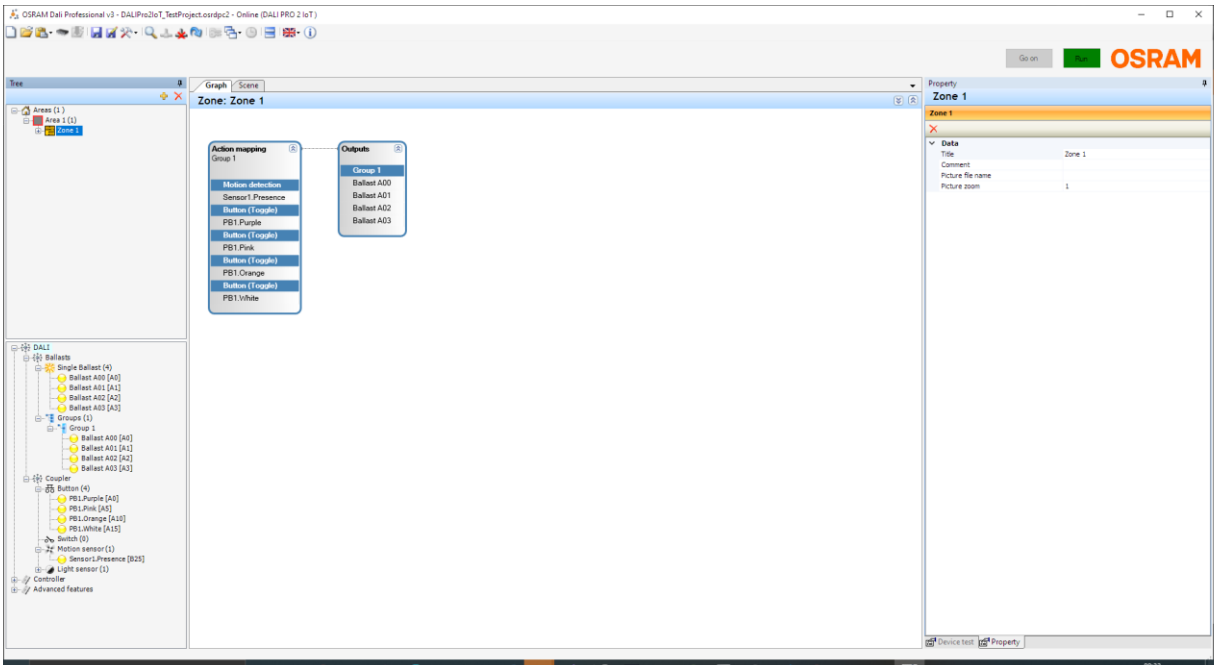


Figure 1

3.2 Setup the MQTT broker on your PC

The DALI PRO 2 is equipped with an MQTT client which is meant to send data to an MQTT Broker. To run the examples in this document, let’s install and run the *mosquito* broker on a pc.

1. Install *Mosquitto* (download from https://mosquitto.org/download/)

2. Open a terminal window (cmd command) and navigate to the mosquito installation directory (for example C:\Program Files\mosquitto)

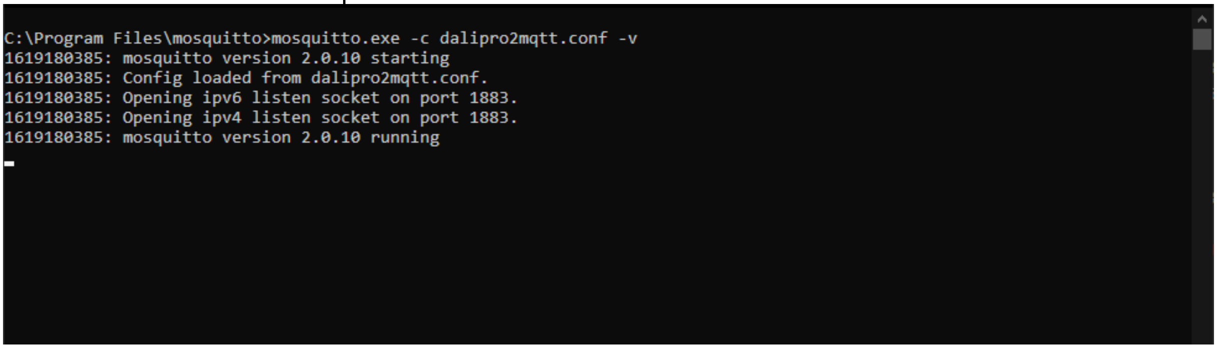
3. In the mosquito installation directory create a text file called dalipro2mqtt.conf. Open the text file and save the following lines (typical it must be open as Administrator to write into the "Program Files" sub folder):

```
listener 1883

protocol mqtt

allow_anonymous true
```

4. From the terminal window run the following command:

```
mosquitto -c dalipro2mqtt.conf -v
```

5. Leave the terminal window opened



```
C:\Program Files\mosquitto>mosquitto.exe -c dalipro2mqtt.conf -v
1619180385: mosquitto version 2.0.10 starting
1619180385: Config loaded from dalipro2mqtt.conf.
1619180385: Opening ipv6 listen socket on port 1883.
1619180385: Opening ipv4 listen socket on port 1883.
1619180385: mosquitto version 2.0.10 running
```

Figure 2

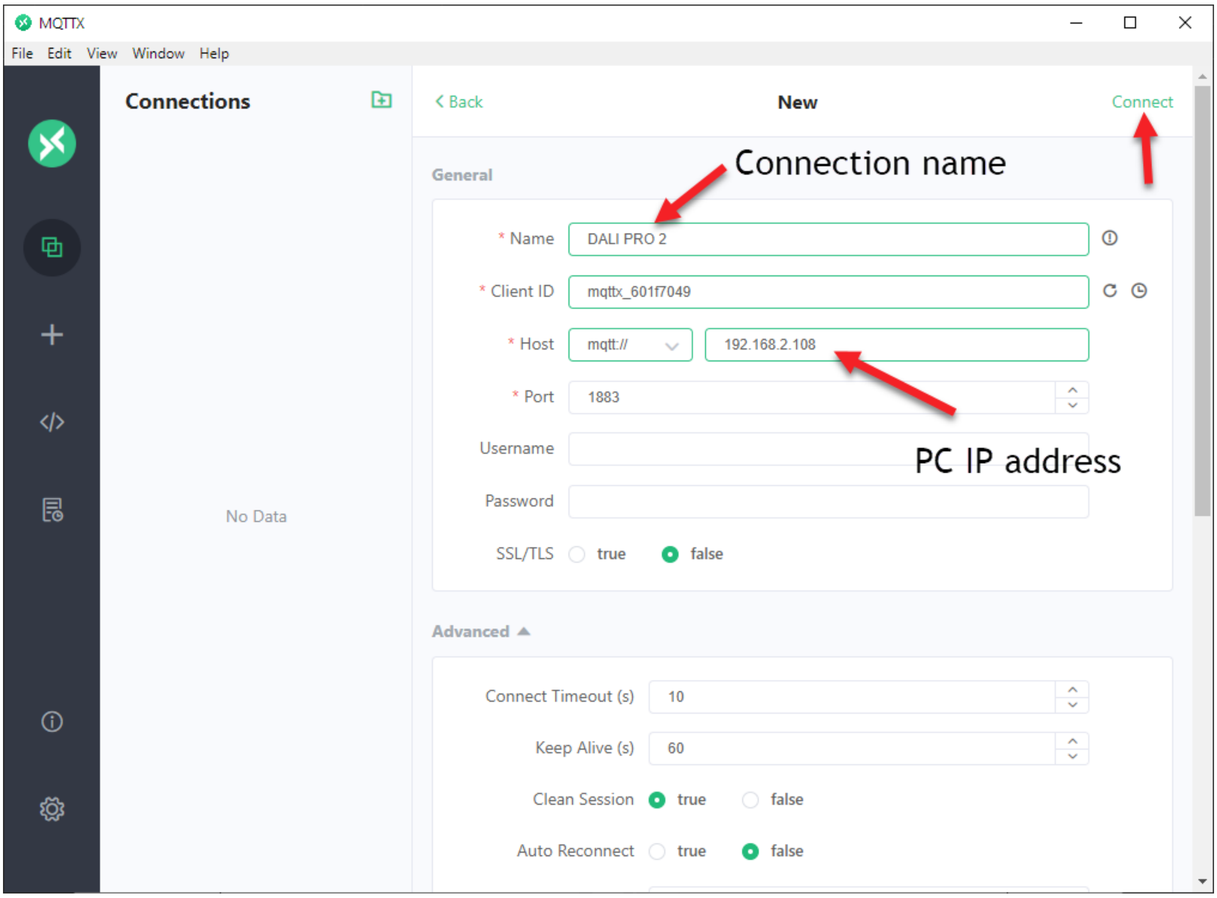
Alternatively, to always have MQTT running in the background, add the 3 lines at the end of the existing configuration file *mosquitto.conf* in the same folder. The terminal window is then no longer required.

3.3 Setup the MQTT client on your PC

In order to receive the messages sent by the *DALI PRO 2* we need to setup an MQTT client and subscribe to the topics of interest.

1. Install and run *MQTT X* (download: https://mqttx.app/)

2. Click on *Create new connection*. A settings panel will appear.
   a. In the *Host* field set the IP address of your PC.
      This is the IP address of the MQTT broker. As in this guide we are running the broker in the PC, we need to put the IP address of the PC.
   b. In the *Name* field set a name of your choice for this connection
   c. Click on *Connect*



- * Name: DALI PRO 2
- * Client ID: mqttx_601f7049
- * Host: mqtt:// 192.168.2.108
- * Port: 1883
- Username:
- Password:
- SSL/TLS: ○ true ● false
- Connect Timeout (s): 10
- Keep Alive (s): 60
- Clean Session: ● true ○ false
- Auto Reconnect: ○ true ● false

Figure 3

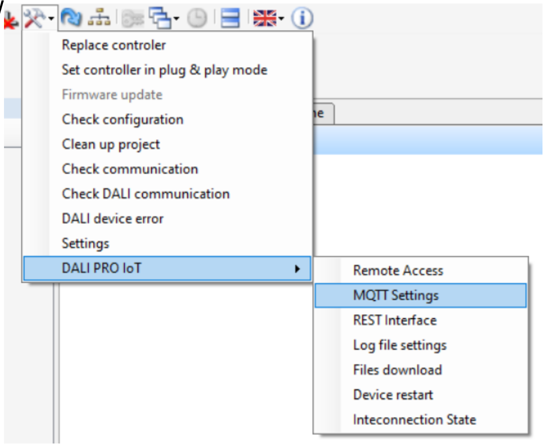
4. Setup the MQTT client inside the *DALI PRO 2 IoT*

The setup of the MQTT client can be done comfortable by the PC-Tool or over the REST API by using of any suitable tool (in den example by the Postman REST tool).

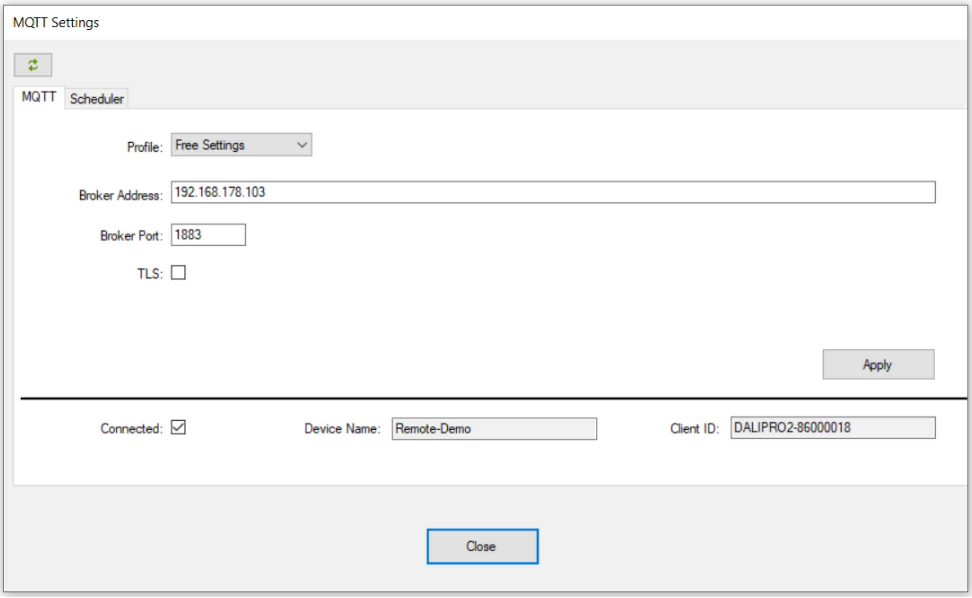
> There are multiple options the DALI PRO 2 IoT can use to connect to an MQTT broker, depending on the broker configuration. Please refer to Appendix B for a list of the available options.

4.1 Setup the MQTT client by using of the PC-Tool

The dialog w r the tool icon:



For the general MQTT settings will be used the profile *Free Settings*. The other profiles offer presetting of some properties depending well known applications.


MQTT Settings

Profile: Free Settings

Broker Address: 192.168.178.103

Broker Port: 1883

TLS: □

Connected: ☑

Device Name: Remote-Demo

Client ID: DALIPRO2-86000018

After changing the properties and clicking on the button *Apply* button, the *Connected* field shows whether the connection was successful.

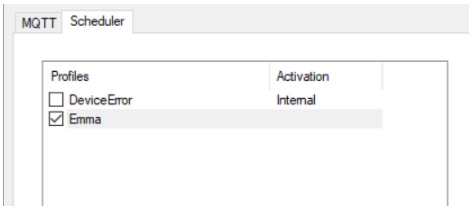
The field *Device Name* informs about the current valid <device name> in the published topics starting general with:

```
DALI-PRO-IoT/<device name>/
```

The field *Client ID* shows the necessary Client ID value used in connection to the MQTT broker.

The query and publishing of DALI data will be triggered from the scheduler part and the current set active scheduler profiles.

In this dialog it is also possible to enable/disable the scheduler profiles:



Profiles	Activation
☐ DeviceError	Internal
☑ Emma	

The default profiles are *Device Error* (check for device errors) and *Emma* (energy monitoring). Internal activated profiles can not be disabled over the interface.

Individual new profiles can be built over the REST API.

4.2 Setup the MQTT client by using of the REST API

The *DALI PRO 2* is equipped with an MQTT client which is meant to send data to an MQTT broker.

This client needs to be configured with the broker parameters and for this task we will use one of the REST API calls.
For a quick reference on how to install Postman and access the REST API, please refer to (3)

1. Open *Postman*

2. Follow the authentication handling descript in reference (3)

3. In the address bar input the following URL:

   ```
   https://<DALI_PRO2_IP>/api/bmsapi/mqtt-interface
   ```

   Where:
   <DALI_PRO2_IP> shall be substituted with the actual DALI PRO 2 IP address

4. In the methods dropdown select “PUT”

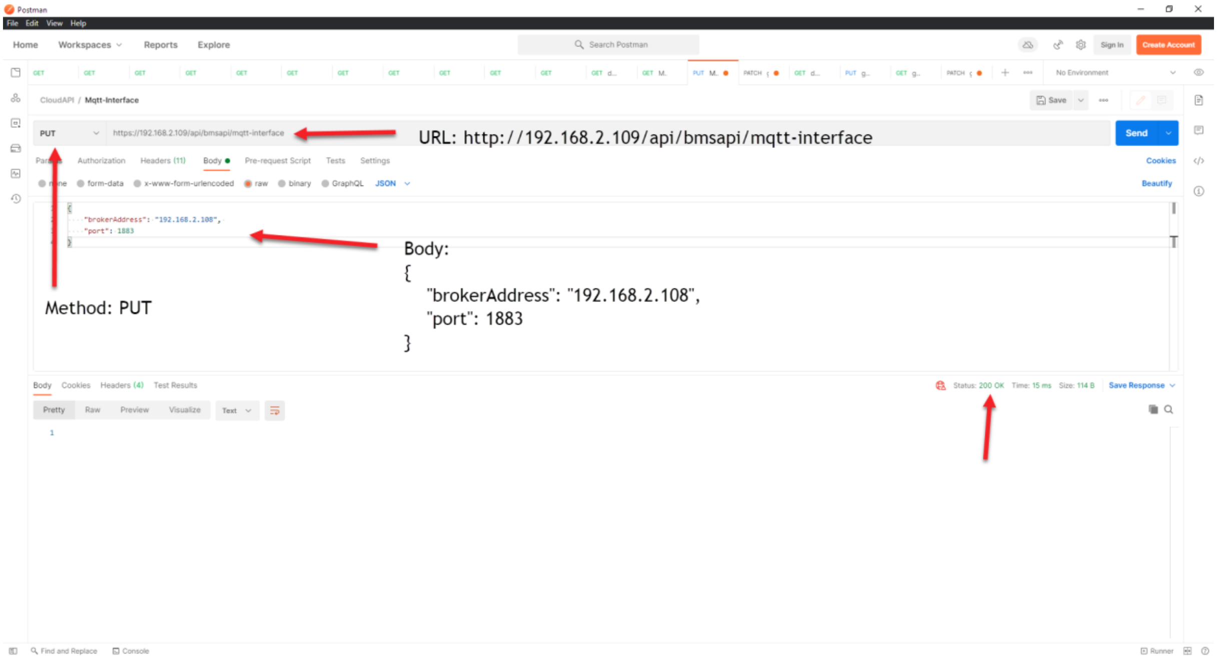
5. Click on the Body tab and insert the following payload:

   ```
   {
     "brokerAddress" : "<BROKER_IP_ADDRESS>",
     "port": 1883
   }
   ```

   Where:
   <BROKER_IP_ADDRESS> shall be substituted with the IP address of the PC where the MQTT broker is running.

6. Press the “Send” button

7. The response status should be *200 OK*



PUT https://192.168.2.109/api/bmsapi/mqtt-interface

```
{
    "brokerAddress": "192.168.2.108",
    "port": 1883
}
```

URL: http://192.168.2.109/api/bmsapi/mqtt-interface

Method: PUT

Body:

```
{
    "brokerAddress": "192.168.2.108",
    "port": 1883
}
```

Status: 200 OK Time: 15 ms Size: 114 B

```
1
```

Figure 4

5. Examples

5.1 Get notifications on the light group status

In order to get notifications every time a group status changes (e.g. dimming level), it's enough to subscribe to the *groups* topic.

8. Open MQTT X

9. Click on *New Subscription*

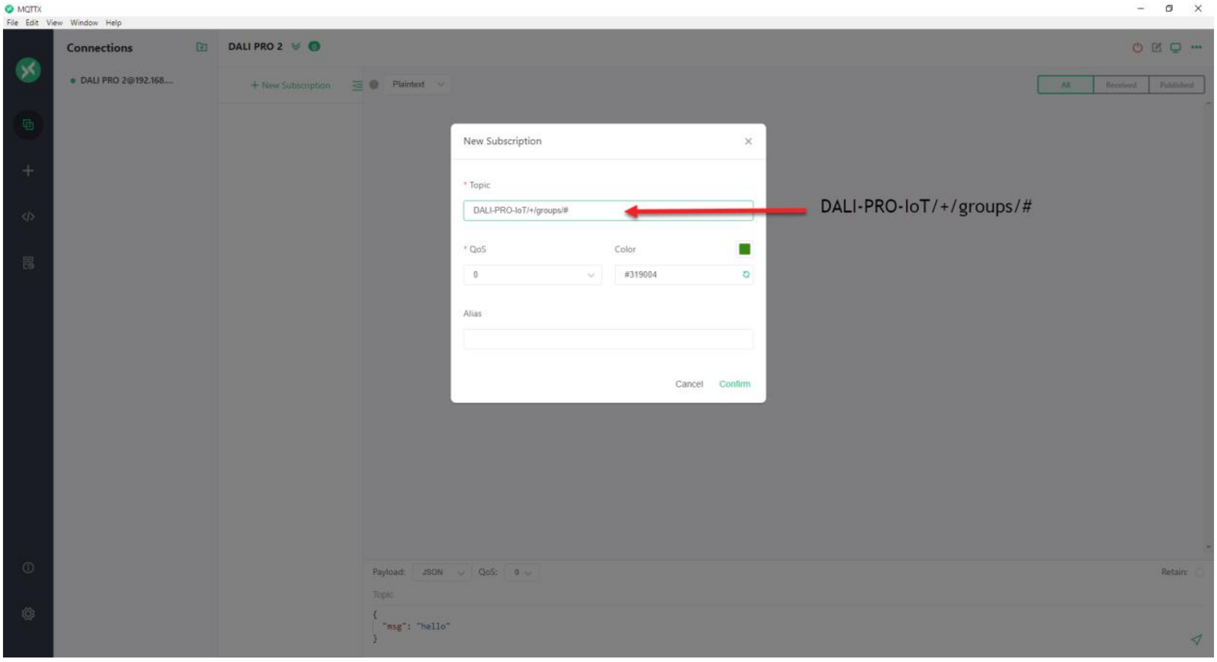
10. In the topic field enter

```
DALI-PRO-IoT/+/groups/#
```

    This will allow us to receive notification from any group configured in the DALI PRO 2.

11. Click Confirm

12. Change the dimming level of a group (e.g. by pressing a button). The notifications will appear in the MQTT X panel.



New Subscription

* Topic: DALI-PRO-IoT/+/groups/#

* QoS: 0

Color: #319004

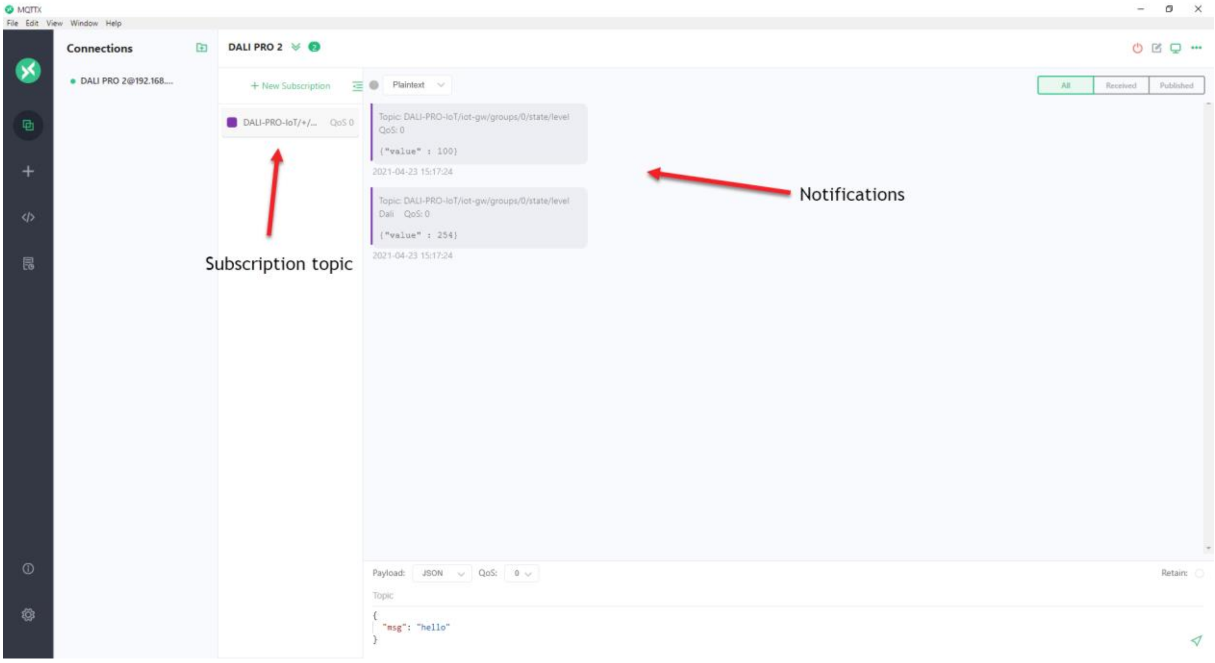
Alias:

Payload: JSON QoS: 0

```
{
  "msg": "hello"
}
```

DALI-PRO-IoT/+/groups/#

Figure 5



Topic: DALI-PRO-IoT/iot-gw/groups/0/state/level
QoS: 0

```
{"value" : 100}
```

2021-04-23 15:17:24

Topic: DALI-PRO-IoT/iot-gw/groups/0/state/level
Dali QoS: 0

```
{"value" : 254}
```

2021-04-23 15:17:24

```
{
  "msg": "hello"
}
```

Figure 6

Hint:
The publishing of general information about all groups and the last state can be trigger by publishing of any message to the topic:

```
DALI-PRO-IoT/<device name>/devices/update
```

5.2 Control the level of the connected lights

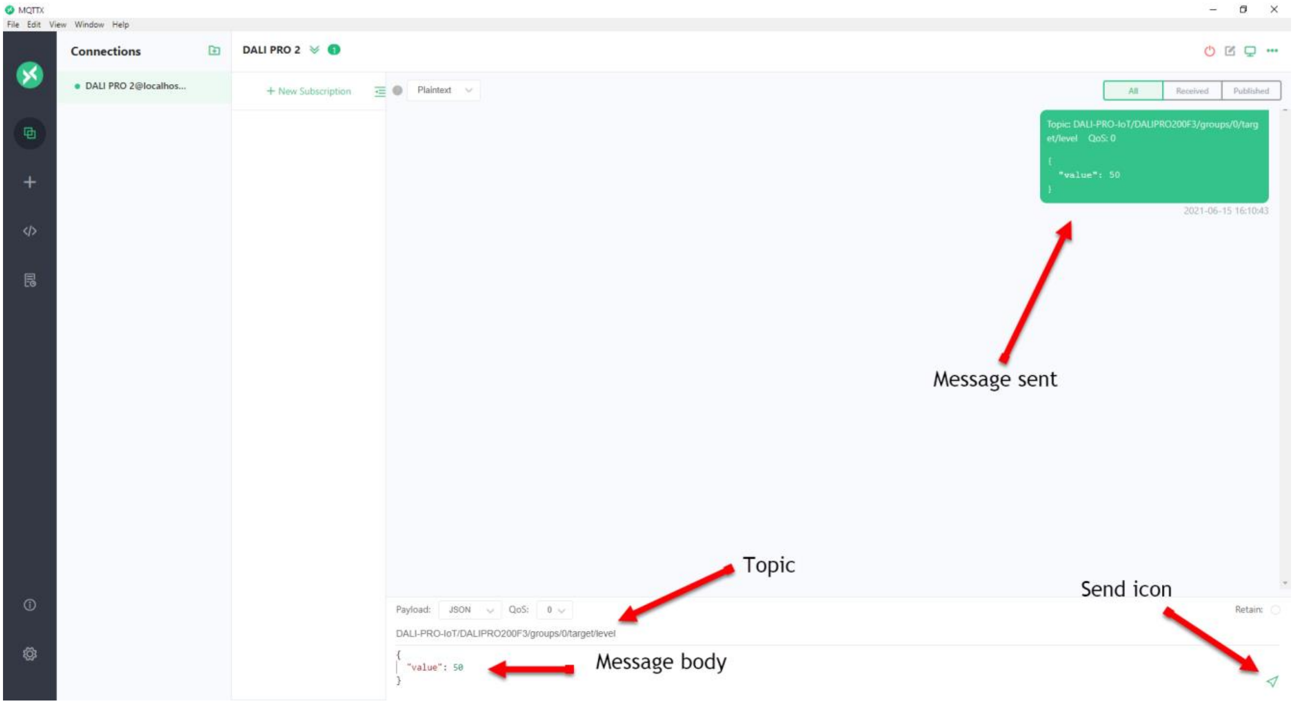
In order to control the level of the connected lights, the following steps shall be followed.

1. Open MQTT X
2. Click on *New Subscription*
3. In the “publish” area, in the topic field enter

   ```
   DALI-PRO-IoT/<device name>/groups/<group ID>/target/level
   ```

   Where:
   - <device name>: is the DALI PRO controller name set in the WebUI/PC-Tool
   - <group ID>: is the ID of the group that needs to be controlled as number 0, 1, ..

4. In the body area specify the value of the property. As detailed in
5. Table 1 every property has its own type. The right type shall be used in message body.
6. Click on the send icon to publish the message



Topic: DALI-PRO-IoT/DALIPRO200F3/groups/0/target/level QoS: 0

```
{
  "value": 50
}
```

2021-06-15 16:10:43

Payload: JSON QoS: 0

DALI-PRO-IoT/DALIPRO200F3/groups/0/target/level

```
{
  "value": 50
}
```

Figure 7

{property}	Type	Description
title	text	Title of the group (read only)
level	value	The actual light level as percent value (0 - 100)
levelDali	value	The actual light level as DALI value (0 - 254)
twMired	value	The actual color temperature in the unit "Mired" Kelvin = 100000 / Mired
twKelvin	value	The actual color temperature in the unit "Kelvin"
rgb	value	The RGB value or the RGB part of RGBW as decimal number For example: red: 0xFF0000 = 16711680 green: 0x00FF00 = 65280 blue: 0x0000FF = 255
white	value	The white part of RGBW in the range 0-511 0 - 255: additional component to the RGB part 256 - 511: white always on maximum and the RGB part will be reduced
sceneNr	value	The active scene number (range 0…15) or (-1) for no active scene
scene	text	The active scene title or null, if there are no active scenes

Table 1

Example message body for the type "value"

```
{
  "value" : 80
}
```

Example message body for the type "text"

```
{
  "text" : "scene 7"
}
```

5.3 Get DALI Data

The requirement to get DALI data is an active scheduler profile. The profile can be activated over the MQTT settings dialog in the PC-Tool or by using of the REST API. With the REST API (3) it is also be possible to build individual scheduler profiles. The overview about all possible properties is in Appendix A.

From the Scheduler will be triggered in intervals the DALI communication to retrieve the current value from each DALI driver. If the value different to the last value, the new value will be published over MQTT.

To get notified about "DALI Data" properties change it’s enough to subscribe to the *data* topic.

1. Open MQTT X
2. Click on *New Subscription*
3. In the topic field enter

```
DALI-PRO-IoT/+/devices/+/+/data/#
```

4. Click Confirm
5. Wait until changed DALI property values will be retrieved from the device.

Hint:
The publishing of general information about all devices and the last values can be trigger by publishing of any message to the topic:

```
DALI-PRO-IoT/<device name>/devices/update
```

6. Topic

6.1 Overview

Typical all topics (except the discovery topic) starting with:

```
DALI-PRO-IoT/<device name>/
```

Supported topics:

Topic
DALI-PRO-IoT/broadcast/discovery
DALI-PRO-IoT/<device name>/application-controller/info/<property>
DALI-PRO-IoT/<device name>/application-controller/state/<property>
DALI-PRO-IoT/<device name>/application-controller/update
DALI-PRO-IoT/<device name>/groups/<group ID>/state/<property>
DALI-PRO-IoT/<device name>/groups/<group ID>/target/<property>
DALI-PRO-IoT/<device name>/groups/update
DALI-PRO-IoT/<device name>/devices/<category>/<guid>/properties/<property>
DALI-PRO-IoT/<device name>/devices/update
DALI-PRO-IoT/<device name>/devices/<category>/<guid>/data/<property>

The controller is listening to publish topics to his own <device name> or to "broadcast".

To receive all messages published by any controller subscribe to:

```
DALI-PRO-IoT/#
```

6.2 Application controller

Topic for general information about the application controller:

```
DALI-PRO-IoT/<device name>/application-controller/info/<property>
```

with the possible values for <property>:

<property>	Type	Description
serialNumber	text	Serial number of the controller

<property>	Type	Description
gtin	text	GTIN of the controller

These topics will be published after receiving any message to the discovery and update topic.

Main topic for state information about the application controller:

```
DALI-PRO-IoT/<device name>/application-controller/state/<property>
```

with the possible values for <property>:

<property>	Type	Description
running	value	1 if configuration running, else 0
plugAndPlayMode	value	1 if in unconfigured state, else 0 for configuration is existing
applicationGuid	text	The guid of the configuration or null if in unconfigured state

These topics will be published after receiving any message to the update topic or is any value internal changed.

The payload as json, for the value type:

```
{
  "value" : 1
}
```

and for the text type:

```
{
  "text" : "86000018"
}
```

The instantaneous publishing of this messages by the controller can be triggered by publishing of any message to the topic:

```
DALI-PRO-IoT/<device name>/application-controller/update
```

6.3 Light groups

Topic for data about light groups:

```
DALI-PRO-IoT/<device name>/groups/<group ID>/state/<property>
```

with the possible values for <property>:

<property>	Type	Description
title	text	Title of the group (read only)
info	special	General information about the light group (read only).
level	value	The actual light level as percent value
levelDali	value	The actual light level as DALI value
twMired	value	The actual color temperature in the unit "Mired"
twKelvin	value	The actual color temperature in the unit "Kelvin"
rgb	value	The RGB value or the RGB part of RGBW
white	value	The white part of RGBW
sceneNr	value	The active scene number (range 0…15) or (-1) for no active scene
scene	text	The active scene title or null if there no active scene

The payload as json, for the value type:

```
{
  "value" : 90
}
```

and for the text type:

```
{
  "text" : "Scene A"
}
```

For the property *info* is the payload equal to the response to the REST endpoint:

```
GET https://<DALI_PRO2_IP>/api/bmsapi/groups/{groupId}
```

with the example payload:

```
{
  "groupId": 1,
  "colorType": "tw",
  "title": "Tunable White Demo",
  "scenes": [
    {
      "title": "Scene A",
      "sceneNr": 0
    },
    {
      "title": "Scene B",
      "sceneNr": 1
    },
    {
      "title": "Scene C",
      "sceneNr": 2
    }
  ]
}
```

The light state can be changed by publishing of the new value to the topic:

```
DALI-PRO-IoT/<device name>/groups/<group ID>/target/<property>
```

The parameter <*group ID*> is a up counting number 0, 1, 2, … and will be automatically assigned to the group while configuration upload.

The instantaneous publishing of information about all existing groups and the current light state can be triggered by publishing of any message to the topic:

```
DALI-PRO-IoT/<device name>/groups/update
```

6.4 Device data (DALI data)

The description of a device will be published on the topic:

```
DALI-PRO-IoT/<device name>/devices/<category>/<guid>/properties/info
```

with the parameter <category>

<category>	**Description**
ballast	ECG Will be changed in "gear"!
sensor	Any DALI input device (button, light sensor, motion sensor) Will be changed in "coupler"!

and the parameter <guid> with the unique ID for each device.

The payload is equal to the response to the REST endpoint:

```
GET https://<DALI_PRO2_IP>/api/bmsapi/dali-devices/<guid>/property/info
```

with the example payload:

```
{
  "guid": "77a4bf85-56b0-460b-96c0-68e248cec0cd",
  "type": "gear",
  "groupId": 0,
  "title": "Ballast B04",
  "port": 1,
  "shortAddress": 4,
  "emergencyLight": false,
  "gtin": "4008321371560",
  "gtinLabel": "OTi DALI 75220...24024 1...4 CH",
  "serial": "4294706693",
  "zone": "Zone 1",
  "zones": ["Zone 1"],
  "properties": ["lightLevel", "gearStatus", "lampFailure", "errorBits"]
}
```

The publishing is triggered after receiving of an *update* message.

Device data will be published with for the topic:

```
DALI-PRO-IoT/<device name>/devices/<category>/<guid>/data/<property>
```

The possible values for the parameter <property> are part of the general device description.

The payload contains the current value, the unit of the value and the data/time:

```
{
  "date": "2022.09.14",
  "time": "11:07:10",
  "value": 232.6,
  "unit": "Vrms"
}
```

(In the future version the date/time will be changed to the UTC time format.)

The data value will be published if the value changed or after receiving of the *update* message.

The internal update of the data will do by the scheduler. Only properties that are part of an active scheduler profile will be updated in the interval set in the profile.

The values published after receiving of the *update* message are the last internal stored value for each property independent how old is this value.

The instantaneous publishing of the device descriptions and all last internal stored values can be triggered by publishing of any message to the topic:

```
DALI-PRO-IoT/<device name>/devices/update
```

6.4.1 Switch, *switchState* and *inputEvent*

To get the current state of a switch, with receive of the *update* message the current input value of all connected switches (button coupler in switch) mode will be queried. After receiving on the DALI side, the current value will be published as <property> *switchState* and is 0 for "off" and 1 for "on".

In normal operation the change of the state will be published as <property> *inputEvent* as represent of the forward frame send out by the coupler to the application controller with the values 9 for "switch off" 10 and for "switch on".

7. References

1. **OSRAM.** DALI PRO 2 - Cloud API Specification.

2. **MQTT: The Standard for IoT Messaging. [Online] https://mqtt.org/.**

3. **OSRAM. DALI PRO 2 IoT - REST API Quick Start Guide.**

8. Appendix A

8.1 DALI control gear properties

The availability in the device depends on the supported device types.

propertyId	Description	Mem Bank
level	DALI control gear light level	-
status	DALI control gear status byte	-
lampFailure	DALI control gear lamp error bit	-

8.1.1 Energy consumption, DALI-part 252, device type 51

propertyId	Description	Mem Bank
driverInputPower	Active input power	202
driverEnergyConsumption	Active input energy	202
driverInputApparentPower	Apparent input power	203
driverApparentEnergyConsumption	Apparent input energy	203
driverLoadsidePower	Output power	204
driverLoadsideEnergy	Output energy	204

8.1.2 Control gear diagnostics and maintenance, DALI-part 253, device type 52

Method	Description	Mem Bank
driverOperationTime	Driver operation time [s]	205
mainPowerUpCount	Driver power cycle counter	205
driverInputVoltage	Driver input voltage [Vrms]	205
powerFactor	Driver power factor	205
inputFrequency	Driver input frequency [Hz]	205
errorOverall	Overall error flag	205
errorOverallCount	Overall-Error counter	205
errorUndervoltage	Driver input undervoltage detection flag	205

Method	Description	Mem Bank
errorUndervoltageCount	Driver input undervoltage detection counter	205
errorOvervoltage	Driver input overvoltage error flag	205
errorOvervoltageCount	Driver input overvoltage error counter	205
errorOutputPowerLimit	Driver output power limitation flag	205
errorOutputPowerLimitCount	Driver output power limitation count	205
errorThermalDerating	Driver thermal derating flag	205
errorThermalDeratingCount	Driver thermal derating counter	205
errorThermalShutDown	Driver over temperature shutdown flag	205
errorThermalShutDownCount	Driver over temperature shutdown counter	205
driverTemperature	Driver temperature [°C]	205
outputCurrentPercent	Real output arc power [%]	205

8.1.3 Light source diagnostics and maintenance, DALI-part 253, device type 52

Method	Description	Mem Bank
lampOnCountRelativ	Lamp switch on counter, resettable	206
lampOnCount	Lamp switch on counter	206
lampOperationTimeRelativ	Lamp operation timer, resettable [s]	206
lampOperationTime	Lamp operation timer [s]	206
outputVoltage	Driver output voltage [V]	206
outputCurrent	Driver output current [A]	206
errorLamp	Lamp error flag	206
errorLampCount	Lamp error counter	206
errorLampShortCircuit	Lamp short circuit error flag	206
errorLampShortCircuitCount	Lamp short circuit error counter	206
errorLampOpenCircuit	Lamp open circuit detection counter	206
errorLampOpenCircuitCount	Lamp open circuit detection counter	206
errorLampThermalDerating	Lamp thermal derating flag	206
errorLampThermalDeratingCount	Lamp thermal derating counter	206
errorLampThermalShutDown	Lamp over temperature shutdown flag	206

Method	Description	Mem Bank
errorLampThermalShutDownCount	Lamp over temperature shutdown counter	206
lampTemperature	Lamp temperature [°C]	206

8.1.4 Device information data, DALI-part 251, device type 50 and DALI-part 253, device type 52

The device information data are static data and will be generated while commissioning. The base data are read out from the memory banks and set by the luminary manufacture. Selected data items can additional changed in the ECG property page in the DALI PRO TOOL.

Method	Description	Mem Bank
oemGtin	GTIN of the luminary Property "OEM GTIN" in PC-Tool	1
oemSerial	Serial number of the luminary	1
luminaireYearOfManufacture	Year of the luminary manufacture	1
luminaireWeekOfManufacture	Week of the luminary manufacture	1
nominalInputPower	Nominal input power Property "Nominal input power" in PC-Tool	1
powerAtMinDimLevel	Power a minimum dim level [W]	1
nominalMinAcMainsVoltage	Nominal minimum AC mains voltage [V]	1
nominalMaxAcMainsVoltage	Nominal maximum AC mains voltage [V]	1
nominalLightOutput	Nominal light output in lumen [Lm]	1
colourRenderingIndex	Colour rendering index of the light Property "Colour rendering index " in PC-Tool	1
colourTemperature	Colour temperature of the light [K] Property "Colour temperature" in PC-Tool	1
lightDistributionType	Light distribution type (type I…type V)	1
luminaireColor	Colour as string	1
luminaireIdentification	Luminaire identification string Property "Luminaire identification" in PC-Tool	1
ratedMedianUsefullLifeOfLuminaire	Rates median useful lifetime in hour Property "Expected livetime" in PC-Tool	207
internalControlGearReferenceTemperature	Reference temperature on the device for 25 °C ambient temperature	207
ratedMedianUsefulLightSourceStarts	Rates median lamp switch on actions	207

8.2 DALI input device (control device) properties

The availability in the device depends on the supported device types.

propertyId	Description	Mem Bank
inputStatus	The input device status byte	-
switchState	The current state of button coupler in switch mode (0-off, 1-on)	
inputEvent	The last input event for button and motion sensor	-
lightLevel	The light level from light sensor	-

8.3 Default scheduler profile

propertyId	Interval time [s]
driverInputVoltage	600
driverEnergyConsumption	300
driverInputPower	60
driverOperationTime	300
driverTemperature	60
errorOverall	3600
lampOperationTime	300
lightLevel	60

9. Appendix B

Depending on the MQTT broker configuration, the client shall be configured in such a way that a connection with the broker can be established.
As already described in paragraph 4, the MQTT client shall be configured using the following REST API call:

```
http://<DALI_PRO2_IP>/api/bmsapi/mqtt-interface
```

The body of the call may vary depending on the following use cases.

9.1 Unencrypted broker not requiring user authentication (*anonymous* access)

This is the case of a broker not requiring any user authentication and transmitting data on a clear channel (not encrypted)
The body of the REST call shall contain the following information:

```
{
“brokerAddress“ : “<BROKER_ADDRESS>“,
“port“: <BROKER_PORT_NUMBER>
}
```

Where:

- <BROKER_ADDRESS> shall be substituted with the IP address of the broker
- <BROKER_PORT_NUMBER> shall be substituted with the TCP port the broker is listening to (default is 1883).

9.2 Unencrypted broker requiring user authentication

This is the case of a broker requiring user authentication and transmitting data on a clear channel (not encrypted)
The body of the REST call shall contain the following information:

```
{
“brokerAddress“ : “<BROKER_ADDRESS>“,
“port“: <BROKER_PORT_NUMBER>,
“username“: <USERNAME>,
“password”: <PASSWORD>
}
```

Where:

- <BROKER_ADDRESS> shall be substituted with the IP address of the broker
- <BROKER_PORT_NUMBER> shall be substituted with the TCP port the broker is listening to (default is 1883).
- <USERNAME>, <PASSWORD> shall be substituted with a username and password that are accepted by the broker

9.3 Encrypted broker (with or without user authentication)

This is the case of a broker which requires a secure MQTT (aka “MQTTs”) connection to be established. A secure MQTT connection is an MQTT connection where all packets are encrypted following the TLS standard.

An encrypted MQTT connection can be established on top of either an anonymous access connection (see paragraph 9.1) or a user authenticated connection (see paragraph 9.2)

The body of the REST call shall contain the following information:

```
{
“brokerAddress“ : “<BROKER_ADDRESS>“,
“port“: <BROKER_PORT_NUMBER>,
“username“: <USERNAME>, -> only for user authenticated sessions
“password”: <PASSWORD>, -> only for user authenticated sessions
“encrypted”: true,
“validateServerCert” : true,
“cacert”: <CA_CERTIFICATE>,
“clientcert” : <CLIENT_CERTIFICATE>,
“clientkey” : <CLIENT_KEY>
}
```

Where:

- <BROKER_ADDRESS> shall be substituted with the IP address of the broker
- <BROKER_PORT_NUMBER> shall be substituted with the TCP port the broker is listening to (default is 1883).
- <USERNAME>, <PASSWORD> shall be substituted with a username and password that are accepted by the broker (not needed for anonymous connections)
- <CA_CERT>: Certification Authority certificate. This is the certificate released by the certification authority (CA) which signed the server (broker) certificate. This is needed to verify the broker identity.
- <CLIENT_CERTIFICATE>: this is the certificate the client uses to establish the secure connection.
- <CLIENT_KEY> : this is the private key associated to the <CLIENT_CERTIFICATE>.

Note: the <CA_CERT>, <CLIENT_CERTIFICATE> and <CLIENT_KEY> strings have to be replaced with the content of the corresponding certificate files, but with special encoding: all newlines have to be replaced with "\n" (two characters)!
This can easily be achieved, for example, with the Linux utility *awk* :

```
awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' client.key; echo
```

As an example, the payload to configure a secure MQTT broker could look like the following:

```
{
    "brokerAddress": "192.168.13.142",
    "port": 1883,
    "encrypted" : true,
    "validateServerCert": true,
    "username": "dalipro",
    "password": "dalipro",
    "cacert":"-----BEGIN CERTIFICATE-----
\nMIIDCTCCAfECFDYlKMTz5Gqyzmi4aklaKOviQkN8MA0GCSqGSIb3DQEBCwUAMEEx\nCzAJBgNVBAY
TAklUMREwDwYDVQQIDAhUcmV2aXNkbzEQMA4GA1UEBwwHVHJldmlz\nbzENMAsGA1UECgwEUGllcjA
eFw0yMTEwMDgxMzQ2MzdaFw0yMjEwMDgxMzQ2Mzda\nMEExCzAJBgNVBAYTAklUMREwDwYDVQQID
AhUcmV2aXNkbzEQMA4GA1UEBwwHVHJl\ndmlzbzENMAsGA1UECgwEUGllcjCCASIwDQYJKoZIhvcNAQ
EBBQADggEPADCCAQoC\nggEBAOKxhfriq0GF8bgerCu+lVvPLB3bR1vTfnwaqV8qjtjb99/SaFNVKkuAubYR
\n/EHKy4tzYs1KpySyMqq544s8jVIr5YFSlLp1+XBrgJxDYrLjTgjZMJcTJQFtWWfs\nb5ahDtBDR/eEPnYJ5QH
7nPzFz+lxslVnonYUjfZolXPN0kU2aie6mjU+mmwXYvJl\nSkoFyYsNLCi7p9YrU+YCt62fu94EnxsWp+5kNBm
ydkpxzM8YvkBprQndoDxbDl4r\n4mVF8594ssEbnrFj7nj914hpFPQ3bAWE6NjEpUx0m+wTuUxlHyvzZ+xecm
pfrDxx\npA70LL/4ukJ0CIHxzR/EmnnJk0MCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAX4Mv\n21BoG7Yn
mGmrMS80y2YqjiMUFEw+XAB9WQ8BRJamru+r0N9cb4BVyL2/3rYJcfc/\nzHTkOAk8wjmm3qgx/LHTvQCdd
31lCznhgFzp1iM37lgmKz1hEwcnifOQCsWk9M2n\n6dCj8k2z/T2BR23fPHbDcvsTXsdGK097J1wjp5xXyHWH
zWwCiLur+Msz8O9h+BtN\nwCGRsjBjZtaNSyPSXdVkQtxicOjf3BYc847jEdiaEg8oesQRUF+4uRLCzQ96bjw
G\nYwsOiL+snd2YvBsrHWybx2WmZsrwDoCQeMYVwxLzVvp3O47WFK+k2zSa9EwxW2F9\nUtup7hkslD+hr
By0Gw==\n-----END CERTIFICATE-----\n",
        "clientcert":"-----BEGIN CERTIFICATE-----
\nMIIDCTCCAfECFDYlKMTz5Gqyzmi4akIaKOviQkN8MA0GCSqGSIb3DQEBCwUAMEEx\nCzAJBgNVBAY
TAklUMREwDwYDVQQIDAhUcmV2aXNkbzEQMA4GA1UEBwwHVHJldmlz\nbzENMAsGA1UECgwEUGllcjA
eFw0yMTEwMDgxMzQ2MzdaFw0yMjEwMDgxMzQ2Mzda\nMEExCzAJBgNVBAYTAklUMREwDwYDVQQID
AhUcmV2aXNkbzEQMA4GA1UEBwwHVHJl\ndmlzbzENMAsGA1UECgwEUGllcjCCASIwDQYJKoZIhvcNAQ
EBBQADggEPADCCAQoC\nggEBAOKxhfriq0GF8bgerCu+lVvPLB3bR1vTfnwaqV8qjtjb99/SaFNVKkuAubYR
\n/EHKy4tzYs1KpySyMqq544s8jVIr5YFSlLp1+XBrgJxDYrLjTgjZMJcTJQFtWWfs\nb5ahDtBDR/eEPnYJ5QH
7nPzFz+lxslVnonYUjfZolXPN0kU2aie6mjU+mmwXYvJl\nSkoFyYsNLCi7p9YrU+YCt62fu94EnxsWp+5kNBm
ydkpxzM8YvkBprQndoDxbDI4r\n4mVF8594ssEbnrFj7nj914hpFPQ3bAWE6NjEpUx0m+wTuUxIHyvzZ+xecm
pfrDxx\npA70LL/4ukJ0CIHxzR/EmnnJk0MCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAX4Mv\n21BoG7Yn
mGmrMS80y2YqjiMUFEw+XAB9WQ8BRJamru+r0N9cb4BVyL2/3rYJcfc/\nzHTkOAk8wjmm3qgx/LHTvQCdd
31lCznhgFzp1iM37lgmKz1hEwcnifOQCsWk9M2n\n6dCj8k2z/T2BR23fPHbDcvsTXsdGK097J1wjp5xXyHWH
zWwCiLur+Msz8O9h+BtN\nwCGRsjBjZtaNSyPSXdVkQtxicOjf3BYc847jEdiaEg8oesQRUF+4uRLCzQ96bjw
G\nYwsOiL+snd2YvBsrHWybx2WmZsrwDoCQeMYVwxLzVvp3O47WFK+k2zSa9EwxW2F9\nUtup7hksID+hr
By0Gw==\n-----END CERTIFICATE-----\n",
        "clientkey":"-----BEGIN PRIVATE KEY-----
\nMIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDisYX64qtBhfG4\nHqwrvpVbzywd20db
0358GqlfKo7Y2/ff0mhTVSpLgLm2EfxBysuLc2LNSqcksjKq\nueOLPI1SK+WBUpS6dflwa4CcQ2Ky404I2TCX
EyUBbVln7G+WoQ7QQ0f3hD52CeUB\n+5z8xc/pcbJVZ6J2FI32aJVzzdJFNmonupo1PppsF2LySEpKBcmLD
Swou6fWK1Pm\nAretn7veBJ8bFqfuZDQZsnZKcczPGL5Aaa0J3aA8WwyOK+JlRfOfeLLBG56xY+54\n/deIaR
T0N2wFhOjYxKVMdJvsE7lMSB8r82fsXnJqX6w8caQO9Cy/+LpCdAiB8c0f\nxJp5yZNDAgMBAAECggEBAMI
Alt6MmW8ryB9/8GxHX98AP4sVm84UA00GLCjLbRBd\nv5YGpU2btuSCt2v/nFhZyzP+fXGfD8ex0PaWuwrvu
Sm3c2PnXmzshZoWBiy05oSd\nTqz34kOIGMVN5ZOwnhbUoq7MxYwn2qR990RpJoKUb0gjCzFXzYssMAn
OmwhN/kNH\nn4rtZ0g+7REEekn19F0i+qWjRI1Z6WLSExCQQg9sr6XRX06qUpqbePEHFXcPuclY\nWzgf3g
dW9TAEQ47k9qFpoK3o1W7o/4eG0neeq5PrIjjYO0hUWeuc/aemXYMvm1gz\n6VzrdJ9Hi9RcAYdlJHOoeiCXt
u8sguk+9/oeNHovILECgYEA+teSGrxUbvmGR9Gc\nuCkW+3d1imgcgWm3Msr/e+7EQswfIkfMNIll85YNHkDj
gpyc/Gp+ZUnJShjOeo7j\nGs7RVSEmldTFJ3fwIPO8o8TQ5spksQ0ki218LMeRV9b2GYghlVxyJp0EIwphBcN
9\nInjHKQJc+iPP5xIrTnGs7N7XVH0CgYEA51rVqXLFIxz3wPgeaP029dbEasRAuib8\nUq5rR/Fv0h6SXUj47K
hhxHzRQFjqn8jHq4XFpDFKcNFpCvVJLKITS9DPp9/Y+DZm\n1wMzOHHHT9b6FY2ZG89ucOJr9wmCAvFZ
zs/IvsFOqm0rG7UvId36oi9dEvNh0X8a\nbc4UQIqP0r8CgYEArNWD0fAZrS04nJuNBEcrkrDIyEjb+z8//ztntZja
mbvTZ+dY\niIpzONcWytWi7QfEExe9Z0Tj8K4O9tslF/+QtLtWvr4P6hX12n6WZ7Dgz2R85JFN\nsNQLTz9iocd
3YjguC5/a90gQ5aPtHe+IdQwLDfKW5bNvOKOIO1CP+dXBQHECgYAd\ndJsvb8VZqx6AGxcEFvkgY6+uSD5
VdRE7FanjBwJNpkbMSbYnq2iHM70QjEGTBmUl\nmWqIdR4dLwxIsOIYJKqARJ2rqfxd3c9KIMYssY/GwoAkr
lqIB+jvxmQol7VN2pys\nkBfm8YWqfSu342GU1kZKz+zxtLsHiT88K7i4Xuvg0QKBgQDWG1mIFiI0brMI4/MV\n
1h6D1n9o9lpM5TYW20rKQHfPSIul08cxX/RQWGMYXwUdt5w8fQAE+iGvnaa7FSGQ\nQA4yPcxv1c1mdCVI
6oUpSQXx4mplX/Z+7RmnZYu0Ft/4nfP+Ys9O7OkIbuRzPKIQ\naPKTSOaX17x9grlWvPsgN832CA==\n-----
END PRIVATE KEY-----\n"
}
```