libelium

Behind the change. Beyond the challenge.

Smart Spot Application Note

AN13: Data integration via MQTT v1.7

Índex

1. Introduction

MQTT is one of the most wide IoT protocols used both due to its scalable capabilities and the number of platforms for Smart Cities accepting this protocol. For that reason, MQTT has been introduced in the Smart Spot to be a protocol to report values to 3rd party platforms.

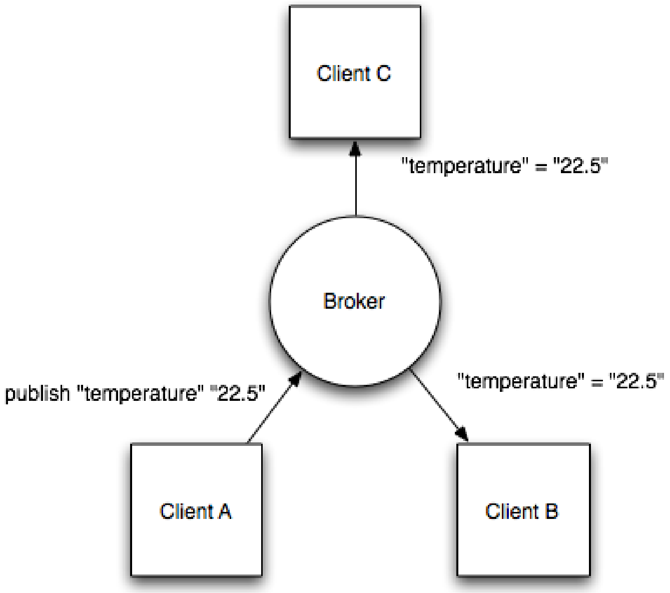
Nowadays, it can be used only to report values, while managing it will be carried out using the original Smart Spot protocol, which is LwM2M. The management can be used both from the platform provided by Libelium so called Homard.

1.1. MQTT communication architecture

MQTT (Message Queue Telemetry Transport) is a lightweight and efficient communication protocol used for sending messages between devices connected to the Internet of Things (IoT). MQTT is based on the publish-subscribe model, in which devices that generate data (publishers or publishers) send information to a central server (broker), which in turn distributes that data to other interested devices (subscribers) that have registered to receive that information.

In this model, publishers publish information on a channel of a specific topic, and subscribers receive information from the topics they are interested in. This allows for efficient and scalable communication, as devices only receive information relevant to them, rather than receiving all network traffic generated by other devices. Additionally, MQTT is a lightweight and efficient protocol, making it ideal for use on resource-limited devices such as sensors and IoT devices.

Next picture illustrates the common message architecture used on MQTT.



Picture 1. Example of interactions between MQTT Clients through an MQTT Broker

1.2. Supported MQTT Version

The SmartSpot uses the most widely used and stable version of MQTT, v3.1.1. This version was released in 2014 and is backward compatible with earlier versions of MQTT, which means that MQTT clients and servers from earlier versions can communicate with MQTT v3.1.1 devices seamlessly.

This version of MQTT also includes significant security and reliability improvements compared to earlier versions. For example, authentication and authorization mechanisms were added to enable safer and more controlled communication between devices. Session management was also improved, and a "last will and testament" mechanism was added to ensure that devices disconnect properly in case of a network failure.

Overall, MQTT v3.1.1 is considered a mature and stable version of the MQTT protocol and is widely used in a variety of IoT applications, such as environmental monitoring, vehicle telemetry, asset tracking, and more.

1.3. Supported Security

MQTT v3.1.1 supports the two main security mechanisms to ensure secure communication between devices. These include:

1. User and password-based authentication: MQTT devices can be authenticated using user and password credentials stored on the MQTT server.
2. Digital certificates: MQTT devices can be authenticated using digital certificates, which provides greater security and mutual authentication between devices.

Both of them can be configured using the remote device management platform Homard provided by Libelium.

1.4. MQTT Ports

MQTT uses by default two different ports for communication:

1. TCP port 1883: This is the default port used for unencrypted connection. MQTT clients connect to the MQTT server using this port. The communication between the client and the server is performed without encryption, meaning that the exchanged data is not protected.

2. TCP port 8883: This is the port used for encrypted connection. MQTT clients can connect to the MQTT server using this port and establish a secure connection over TLS (Transport Layer Security). TLS provides an additional layer of security to MQTT communication, protecting the exchanged data between the client and the server.

It is important to note that default ports may vary depending on the MQTT provider or the MQTT server configuration. Therefore, the MQTT server documentation should be consulted to confirm the ports used in MQTT communication and ensure that clients are properly configured to connect to the server.

1.5. Last Will and Testament

The Last Will and Testament (LWT) system in MQTT allows a device to send a final message to a specific MQTT topic if it unexpectedly disconnects or its connection fails.

When an MQTT device establishes a connection with an MQTT server, it can specify an LWT message along with an MQTT topic. If the device unexpectedly disconnects or its connection fails, the MQTT server will publish the LWT message to the specified MQTT topic. This allows other devices on the network to know that the original device disconnected and take appropriate action.

For example, if an MQTT temperature sensor establishes a connection with an MQTT server and specifies an LWT message to publish "No Signal" to the topic "Sensor/Temperature/Status" in case of disconnection, other devices on the network can monitor that topic and take action if they receive the "No Signal" message.

To use the LWT feature, it is possible to configure it using the device remote platform Homard provided by Libelium.

2. Configuration

2.1. Data Reporting organisation: Vertical Data Concept

Data reporting through the MQTT protocol in the Smart Spot device has always followed the philosophy of being "FIWARE Ready," so that the data is easily integrable into the widespread FIWARE platform, while also being compatible with any other platform not based on FIWARE.

This philosophy is maintained in the latest version of data integration from the Smart Spot through MQTT, but it is important to note that the data publication has been particularly improved with disruptive changes compared to previous versions thanks to the experience and feedback received from customers in recent years.

These changes mainly respond to a need to introduce ease of:

- Separating data sets
- Facilitating the configuration of FIWARE components
- Increasing the scalability of the components and/or backends that receive the data.

The vertical data concept involves grouping and organising the data reported by the device into different verticals depending on the transmitted data set. The grouping of this data aligns with the definition of the official Data Models of the FIWARE platform described on the website https://smartdatamodels.org/.

Given the versatility of the Smart Spot to be configurable at the hardware level during the order with different types of sensors, not all data verticals have to be active on the device. The device transmits data using the necessary verticals depending on the hardware configuration of the device. Additionally, it is possible to enable or disable specific verticals separately through the Homard device remote platform as indicated in the next section.

The definition of a vertical at MQTT integration level is mainly composed of two elements:

- Topic: The topic for publishing each data set of the vertical will be composed based on a default prefix or a topic prefix specified by the client through the Homard device remote platform. The topic prefix must have the following format

  '/{base_apikey}/{device_id}'

  where 'base_apikey' is usually an identifying key for a set of devices and 'device_id' is a specific name for the device. If a topic prefix is not configured, the device will use a default one.

  This topic prefix will be internally modified by the SmartSpot depending on the vertical. Specifically, the SmartSpot will add both the '/attrs' postfix corresponding to the one required by the FIWARE IoT Agents to receive data and the 'lower_vertical_suffix' and 'upper_vertical_suffix' fields within the topic prefix to separate the data into different topics, one for each vertical, resulting in the final topic being composed in the following way:

  '/{base_apikey}{lower_vertical_suffix}/{device_id}_{upper_vertical_suffix}/attrs'

  The following table illustrates an example of the prefix formation according to the publication vertical.

Vertical	Configured topic prefix (Homard)	Vertical Suffix	Final MQTT Topic composed to publish vertical data
Air Quality	/JRI9jTbEKJQS29y3bc8BC6 /SmartSpot_AQ_1	aqo	/JRI9jTbEKJQS29y3bc8BC6aqo/SmartSpot_AQ_1_AQO/attrs
Noise Level		nlo	/JRI9jTbEKJQS29y3bc8BC6nlo/SmartSpot_AQ_1_NLO/attrs
Crowd Monitoring		cfe	/JRI9jTbEKJQS29y3bc8BC6cfe/SmartSpot_AQ_1_CFE/attrs
		cfo	/JRI9jTbEKJQS29y3bc8BC6cfo/SmartSpot_AQ_1_CFO/attrs
Weather		wto	/JRI9jTbEKJQS29y3bc8BC6wto/SmartSpot_AQ_1_WTO/attrs
Device		dev	/JRI9jTbEKJQS29y3bc8BC6dev/SmartSpot_AQ_1_DEV/attrs
Device Health		dho	/JRI9jTbEKJQS29y3bc8BC6dho/SmartSpot_AQ_1_DHO/attrs

Table 1. Example of topic generation for the different possible verticals available on the device

- Message: The data published through JSON-encoded text messages on each of the specified topics contain at least the data from the different sensors included in your Smart Spot device for each vertical, and may also include extra information regarding how the device is configured. This extra information is provided as it may be useful to the user and may enable device configuration via MQTT protocol in the future, although this feature is not currently implemented. The following tables illustrate the information that can be included in each vertical depending on your device's configuration.

Vertical	Air Quality	**Topic Example**	/JRI9jTbEKJQS29y3bc8BC6aqo/SmartSpot_AQ_1_AQO/attrs
Field name	**Format**	**Units**	**Definition**
Mandatory fields always included in all the messages			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	minutes	Sampling period configured in the device in minutes.
status	Text	n/a	Reports ‘connected’ to update the device status on the backend
The following items are dependent of the device hardware configuration			
no2-a4	Number	µg/m3	NO2 sensor measurement
ox-a431	Number	µg/m3	O3 sensor measurement
co-a4	Number	µg/m3	CO sensor measurement
so2-a4	Number	µg/m3	SO2 sensor measurement
h2s-a4	Number	µg/m3	H2S sensor measurement
no-a4	Number	µg/m3	NO sensor measurement
nh3-a4	Number	µg/m3	NH3 sensor measurement
cl2-a1	Number	µg/m3	CL2 sensor measurement
pm1	Number	µg/m3	PM1.0 sensor measurement
pm2	Number	µg/m3	PM2.5 sensor measurement
pm10	Number	µg/m3	PM10 sensor measurement
voc	Number	µg/m3	VOC sensor measurement
co2	Number	µg/m3	CO2 sensor measurement

Message example

```
{
  "TimeInstant":"2023-03-20T13:50:00Z",
  "period":5,
  "status":"connected",
  "no2-a4":0.640869140625,
  "ox-a431":28.586013793945312,
  "co-a4":91.760452270507812,
  "so2-a4":0.927800178527832,
  "pm10":49.334125518798828,
  "pm2":15.33404541015625,
  "pm1":2.1444158554077148
}
```

Table 2. List of fields contained in the transmitted messages for Air Quality vertical

Vertical	Weather	**Topic Example**	/JRI9jTbEKJQS29y3bc8BC6wto/SmartSpot_AQ_1_WTO/attrs
Field name	**Type**	**Units**	**Definition**
Mandatory fields always included in all the messages			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	Minutes	Sampling period configured in the device in minutes.
status	Text	n/a	Reports ‘connected’ to update the device status on the backend
The following items are dependent of the device hardware configuration			
temp8	Number	ºC	Ambient Temperature
hum8	Number	%RH	Ambient Relative Humidity
pres1	Number	Millibar	Barometric pressure
hum3	Number	ºC	Ambient Temperature
temp3	Number	%RH	Ambient Relative Humidity
thsw_index	Number	ºC	THSW Index
wind_dir	Number	º	Wind Direction
wind_sp	Number	km/h	Wind Speed
uv_index	Number	Float	UV Index
solar_rad0	Number	Watts/m2	Solar Radiation
et_daily	Number	millimetres	Accumulated evapotranspiration along the day
dew_point	Number	ºC	Dew Point
rain_rate	Number	Litres/hour	Instant rain rate
rain_15a	Number	Litres/hour	15 min averaged rain rate
rain_daily	Number	Litres/hour	Accumulated rain along the day
rain_24ha	Number	Litres/hour	24 hour averaged rain rate
wind_chill	Number	ºC	Wind Chill
illuminance		lux	Illuminance
Message example			

```
{
  "TimeInstant":"2023-03-20T13:45:00Z",
  "period":5,
  "status":"connected",
  "pres1":980.83404541015625,
  "rain_15a":0,
  "rain_rate":0,
  "hum3":67,
  "temp3":18.05555534362793,
  "thsw_index":19.44444465637207,
  "wind_dir":34,
  "wind_sp":24.140100479125977,
  "uv_index":8.6999998092651367,
  "solar_rad0":900,
  "et_daily":2.0066001415252686,
  "dew_point":11.666666984558105,
  "rain_daily":0,
  "rain_24ha":0,
  "wind_chill":15.55555534362793,
  "illuminance":168750
}
```

Table 3. List of fields contained in the transmitted messages for Weather vertical

Vertical	Noise Level	**Topic Example**	/JRI9jTbEKJQS29y3bc8BC6nlo/SmartSpot_AQ_1_NLO/attrs
Field name	**Type**	**Units**	**Definition**
Mandatory fields always included in all the messages			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	Minutes	Sampling period configured in the device in minutes.
status	Text	n/a	Reports ‘connected’ to update the device status on the backend
son_laeq	Number	dB	Equivalent LA value measured during the period
son_lamax	Number	dB	Maximum LA value measured during the period
son_lamin	Number	dB	Minimum LA value measured during the period
son_la1	Number	dB	Value for percentile 1 measured during the period
son_la10	Number	dB	Value for percentile 10 measured during the period
son_la50	Number	dB	Value for percentile 50 measured during the period
son_la90	Number	dB	Value for percentile 90 measured during the period
son_la99	Number	dB	Value for percentile 99 measured during the period

Message example

```
{
  "TimeInstant":"2023-03-20T13:45:00Z",
  "period":3,
  "status":"connected",
  "son_laeq":66.512710571289062,
  "son_lamax":75.870002746582031,
  "son_lamin":62.689998626708984,
  "son_la1":75.5999984741211,
  "son_la10":69.055557250976562,
  "son_la50":65.357139587402344,
  "son_la90":63.5062484741211,
  "son_la99":63.020454406738281
}
```

Table 4. List of fields contained in the transmitted messages for Noise Level vertical

Vertical	Crowd Monitoring	**Topic Example**	/JRI9jTbEKJQS29y3bc8BC6cfo/SmartSpot_AQ_1_CFO/attrs
Field name	**Type**	**Units**	**Definition**
Mandatory fields always included in all the messages (Do not take in account other extra fields you can receive until next releases)			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	Minutes	Sampling period configured in the device in minutes.
status	Text	n/a	Reports 'connected' to update the device status on the backend
peopleCount	Number	Number of devices	Total number of devices detected within the sampling period
AvgDuration	Number	Seconds	Average time devices are detected within the sampling period
measurementInterval	Number	Minutes	Sampling time set in minutes. Same value as 'period' field.
reportingPeriod	Number	Minutes	Data sending time configured in minutes (If "-1", the same reporting time as sampling time applies). Otherwise, it must be a multiple of the sampling time.
dataCollection	Text	n/a	Data reporting configuration status Possible values: "on", "off".
peopleCountCollection	Number	n/a	Indicates the filtering being applied by the device on the peopleCount report. Possible values: 0 (No count), 1 (Counting restricted to real MACs), 2 (Count restricted to random MACs), 3 (Count all kind of MACs)
eventCollection	Number	n/a	Indicates the sending status of the event report on the "CrowdFlowEvent" entity. Possible values: 0 (No event is reported), 1 (Only real MAC events are reported), 2 (Only random MAC events are reported), 3 (All events are reported).

```
{
  "TimeInstant":"2023-03-20T13:44:06Z",
  "period":1,
  "status":"connected",
  "peopleCount":20,
  "avgDuration":11,
  "measurementInterval":1,
  "reportingPeriod":-1,
  "dataCollection":"on",
  "peopleCountCollection":3,
  "eventCollection":3
}
```

Table 5. List of fields contained in the transmitted messages for Crowd Monitoring vertical

Vertical	Crowd Monitoring	Topic Example	/JRI9jTbEKJQS29y3bc8BC6cfe/SmartSpot_AQ_1_CFE/attrs
Field name	**Type**	**Units**	**Definition**
Mandatory fields always included in all the messages			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	Minutes	Sampling period configured in the device in minutes.
detectionType	Number	n/a	Detection type Possible values: wifi (1), bluetooth (2)
visitorId	Text	n/a	Result of applying a hash algorithm to the device detected MAC. It's an obfuscation in order to protect the user's privacy.
random	Number	n/a	Indicates whether the detected MAC is random or not. Possible values: 0 (false) or 1 (true)
duration	Number	Seconds	Time that the detected device remains at that location within the sampling period performed by the device.
Message example			

```
{
  "TimeInstant":"2023-03-20T13:45:07Z"
  "period":1,
  "detectionType":1,
  "visitorId":"cdb6be6326af1dc10b55fd91d6c26548db230203",
  "random":0,
  "duration":21,
}
```

Table 6. List of fields contained in the transmitted messages for Crowd Monitoring vertical

Vertical	Device	Topic Example	/JRI9jTbEKJQS29y3bc8BC6dev/SmartSpot_AQ_1_DEV/attrs
Field name	**Type**	**Units**	**Definition**
Mandatory fields always included in all the messages			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	Minutes	Sampling period configured in the device in minutes.
status	Text	n/a	Reports ‘connected’ to update the device status on the backend
modelName	Text	n/a	Device Model
serialNumber	Text	n/a	Device Serial Number
firmwareVersion	Text	n/a	Device Firmware Version
updateFirmwareDownloadProgress	Number	%	Percent of download firmware progress

updateFirmwareState	Number	n/a	Attributes for Update firmware process. These attributes should be used in conjunction with the next meaning: State: 0 Result: 0 -> no update in progress. State: 1 Result: 0 -> downloading State: 2 Result: 0 -> downloaded State: 3 Result: 0 -> installing State: 0 Result: 1 -> installed, restart in progress State: 2 Result: 5 -> Fail during download
updateFirmwareResult	Number	n/a	
cellularEnabled	Text	boolean	Indicates whether Cellular connectivity is configured and enabled
wifiEnabled	Text	boolean	Indicates whether WiFi connectivity is configured and enabled
cellularApn	Text	Text	APN configured for cellular connectivity. Only available if 'cellularEnabled' field is 'true'
wifiSsid	Text	Text	WiFi SSID configured for WiFi connectivity. Only available if wifiSsid field is 'true'

Message example

```
{
  "TimeInstant":"2023-03-20T13:45:00Z",
  "period":5,
  "status":"connected",
  "modelName":"A72gG21N1P1E1GDIB10Y1H1",
  "serialNumber":"ac67b2ce52da",
  "firmwareVersion":"0.18.14",
  "updateFirmwareDownloadProgress":0,
  "updateFirmwareResult":0,
  "updateFirmwareState":0,
  "cellularEnabled":"false",
  "wifiEnabled":"false",
  "cellularApn":"example.apn.com",
  "wifiSsid":"defaultSSAP",
}
```

Table 7. List of fields contained in the transmitted messages for Device vertical

Vertical	Device Health	**Topic Example**	/JRI9jTbEKJQS29y3bc8BC6dho/SmartSpot_AQ_1_DHO/attrs
Field name	**Type**	**Units**	**Definition**
Mandatory fields always included in all the messages			
TimeInstant	Text	UTC Date	Start date of sampling performed by the device
period	Number	Minutes	Sampling period configured in the device in minutes.
status	Text	n/a	Reports 'connected' to update the device status on the backend
networkBearer	Number	n/a	Network Bearer: Possible values: GPRS (0), 4G (6), NB-IoT (7), WiFi (21)
rssi	Number	dBm	Connection RSSI value Possible values: worst (0) to best (1)

The following items are dependent of the device hardware configuration			
batteryLevel	Number	n/a	Battery level in percent of one.
batteryState	Number	n/a	Battery State Possible values: 0 (Bypass), 1 (Load), 2 (Discharge)
batteryVoltage	Number	mV	Battery voltage
batteryCurrent	Number	mA	Battery current. Indicates with more detail the exact level of charging or discharging state.

Message example

```
{
  "TimeInstant":"2023-03-20T13:45:00Z",
  "period":5,
  "status":"connected",
  "networkBearer":21,
  "rssi":0.77999997138977051,
  "batteryLevel": 0.87
  "batteryState":0,
  "batteryVoltage":11785,
  "batteryCurrent":-312
}
```

Table 8. List of fields contained in the transmitted messages for Device Health vertical

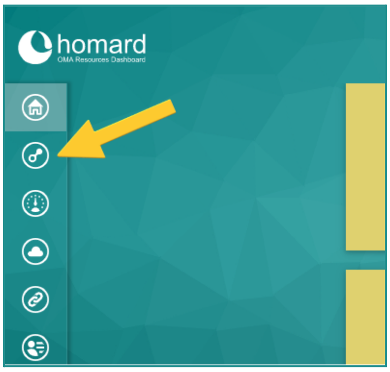
It is important to note that if a sensor is not connected or not working properly, the field is still included in the transmitted message, but its value is set to 'null'. The reasons for carrying out this procedure are mainly to:

- Help identify device malfunctions for maintenance tasks.
- Prevent the introduction of '0' or repetitive non-real values that can cause confusion when data persists in databases or is represented in graphs.

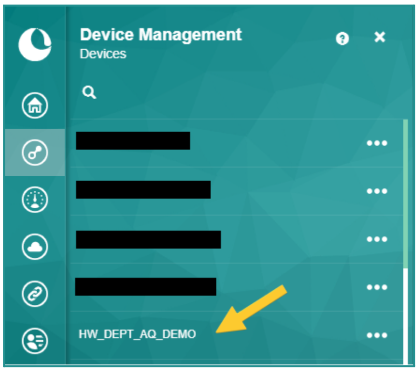
2.2. MQTT Client configuration

Following points illustrate the method to configure the MQTT Client embedded on the Smart Spot to configure a connection to a MQTT Broker by using the Homard Device Management platform provided with the device.

1. Open Homard Device Management platform and login with your account.
2. Use “Device Management” button to access to the list of devices as illustrated in picture 1
3. On the list of devices, click on the device to configure to access to the list of LwM2M Objects as illustrated in picture 2
4. Select the LwM2M Object called “MQTT Client Configuration” from the LwM2M Object list as illustrated in picture 3
5. Use the resources illustrated in Picture 1 and described on Table 9 to configure the MQTT Client. Section 2.2.1 provides configuration examples.



Picture 2. Click on “Device Management” (Step 2)

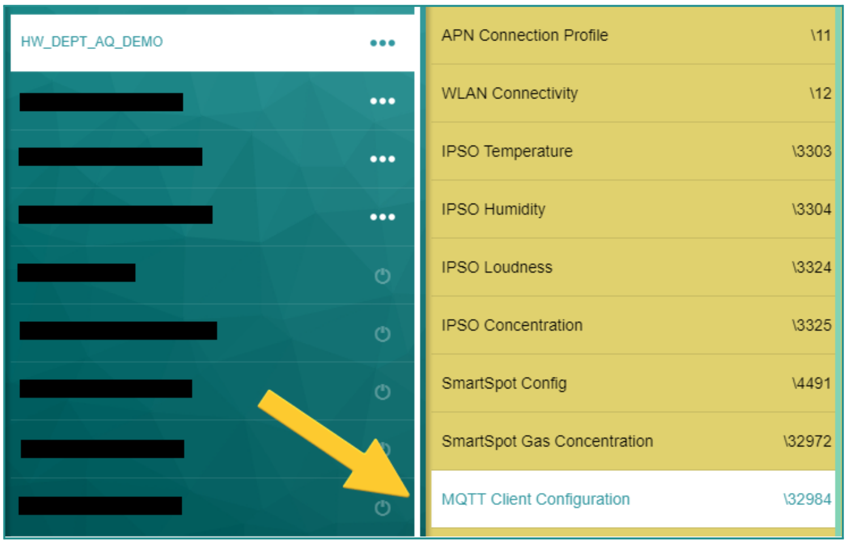


Device Management
Devices

[redacted]

HW_DEPT_AQ_DEMO

Picture 3. Select the device to configure (Step 3)

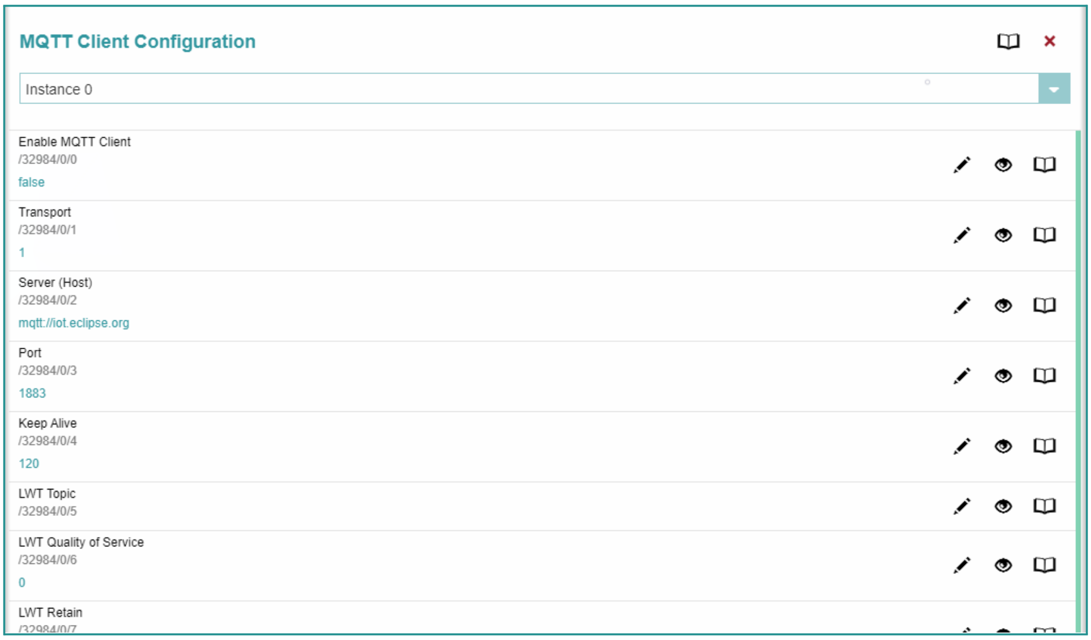


HW_DEPT_AQ_DEMO

[redacted]

APN Connection Profile	\11
WLAN Connectivity	\12
IPSO Temperature	\3303
IPSO Humidity	\3304
IPSO Loudness	\3324
IPSO Concentration	\3325
SmartSpot Config	\4491
SmartSpot Gas Concentration	\32972
MQTT Client Configuration	\32984

Picture 4. Access to the “MQTT Client Configuration” LwM2M Object (Step 4)



MQTT Client Configuration

Instance 0

Enable MQTT Client
/32984/0/0
false

Transport
/32984/0/1
1

Server (Host)
/32984/0/2
mqtt://iot.eclipse.org

Port
/32984/0/3
1883

Keep Alive
/32984/0/4
120

LWT Topic
/32984/0/5

LWT Quality of Service
/32984/0/6
0

LWT Retain
/32984/0/7

Picture 5. Use the resources to configure the MQTT Client (Step 5)

The following table specifies in depth the different resources available on the “MQTT Client Configuration”. Grey rows correspond to non-editable resources used just to report information.

Resource ID	Resource name	Mandatory	Accepted Values	Default value	Description
0	Enable MQTT Client	Yes	Boolean	false	Enables the use of MQTT protocol
1	Transport	Yes	Integer [1-4]	1	Defines the transport method to connect to the MQTT Broker: ● Transport over TCP: Transport publish/receive messages over TCP without security. Set value to 1 to use this transport method. ○ Examples for server field: ■ mqtt://iot.eclipse.org (MQTT over TCP) ■ mqtt://username:password@iot.eclipse.org (MQTT over TCP with username and password) ● 2. Transport over SSL: Transport publish/receive messages using SSL. This messages travel encrypted through the network using a CA (certification authority). Set value to 2 to use this transport method. ○ Examples for server field: ■ mqtts://iot.eclipse.org (MQTT over SSL) ● 3. Transport over WebSocket: Transport publish/receive messages over WebSocket. Set value to 3 to use this transport method. ○ Examples for server field: ■ ws://iot.eclipse.org/ws ● 4. Transport over WebSocket Secure: Transport publish/receive messages over WebSocket using SSL. Set value to 4 to use this transport method. ○ Examples for server field: ■ wss://iot.eclipse.org/ws
2	Server	Yes	String	mqtt://iot.eclipse.org	Indicates the broker server where the messages will be published/received. This parameter depends on the previous parameter. ○ If the previous parameter is configured as “transport over TCP”, the server URL must begin with the ‘mqtt://’ prefix. ○ If the previous parameter is configured as “transport over SSL”, the server URL must begin with the ‘mqtts://’ prefix. ○ If the previous parameter is configured as “transport over WebSocket”, the server URL must begin with the ‘ws://’ prefix. ○ If the previous parameter is configured as “transport over WebSocket Secure”, the server URL must begin with the ‘wss://’ prefix.

3	Port	Yes	Integer [1-4]	1883	Defines the port of the MQTT Broker.
4	Keep Alive	Yes	Integer [20-6000]	120	If there is no data flow over an open connection for a certain time of period, then the client will generate a PINGREQ and expect to receive a PINGRESP from the broker. This message exchange confirms that the connection is open and working. This behaviour is known as the keep alive. If during the period of time indicated in the parameter there is no PINGREQ/PINGRESP, the broker will close the connection with the MQTT Client.
5	LWT Topic	No	String	-	Defines the topic where to publish the LWT message when a device gets disconnected without specifically send the MQTT DISCONNECT message. When the broker detects this disconnection, it sends a message to all the clients subscribed to the indicated topic.
6	LWT Quality Of Service	No	Integer [0-2]	0	Defines the Quality of Service level used for the LWT message (0, 1 or 2).
7	LWT Retain	No	Boolean	false	LWT is often combined with retained messages to store the state of a client on a specific topic. This option alerts subscribed clients and new subscriptions that the client has unexpectedly disconnected.
8	LWT Message	No	String	-	Last Will and Testament message that will be published if an unexpected disconnection happens.
9	Reserved				
10	Username	No	String	-	This option defines the username used to publish messages on servers with the username and password authentication method enabled
11	Password	No	String	-	This option defines the password used to publish messages on servers with the username and password authentication method enabled
12	Certified PEM	Just if SSL or WSS transport method is selected	String	-	String without spaces that defines the server's certificate in PEM format.
13	Client Certified PEM	No	String	-	String without spaces that defines the client certificate in PEM format.
14	Client Key PEM	Just if resource 13 has been filled	String	-	String without spaces that defines the client key in PEM format.
15	QoS for published messages	No	Integer [0-2]	1	Defines the Quality of Service for the published messages. The Quality of Service (QoS) level is an agreement between the sender of a message and the receiver of a message that

					defines the guarantee of delivery for a specific message. There are 3 QoS levels in MQTT. ○ At most Once (0). This service level guarantees a best-effort delivery. There is no guarantee of delivery. The recipient does not acknowledge receipt of the message and the message is not stored and re-transmitted by the sender. QoS level 0 is often called "fire and forget" and provides the same guarantee as the underlying TCP protocol. ○ At least Once (1). This QoS guarantees that a message is delivered at least one time to the receiver. The sender stores the message until it gets a PUBACK packet from the receiver that acknowledges receipt of the message. It is possible for a message to be sent or delivered multiple times. ○ Exactly Once (2). This level guarantees that each message is received only once by the intended recipients. QoS 2 is the safest and slowest quality of service level. The guarantee is provided by at least two request/response flows (a four-part handshake) between the sender and the receiver.
16	Topic prefix for published messages	No	String	-	Define a prefix to be included on the topics used by the MQTT Client to publish information. Allows to define group of devices or identify devices.
17	MQTT Client status	No	Boolean	-	Reports true when the MQTT Client successfully achieve to connect with the MQTT Broker

Table 9. Description of resources available on the "MQTT Client Configuration"

2.2.1. MQTT Client configuration examples

Following lines provides the steps to configure the most common configurations used to integrate the values from the Smart Spot on other platforms with and without security:

MQTT over TCP (without security)

- Enable MQTT Client: true
- Transport: 1
- Server (host): "mqtt://iot.eclipse.org"
- Port: 1883
- Keep Alive:120 (default)
- Quality Of Service for published messages: 1
- Topic prefix for published messages: /JRI9jTbEKJQS29y3bc8BC6/SmartSpot_AQ_1

MQTT over SSL/TLS (with security)

- Enable MQTT Client: true
- Transport: set 2
- Server (host): "mqtts://iot.eclipse.org"

- Port: 8883.
- Keep Alive: 120 (default)
- Certified Pem:

-----BEGIN
CERTIFICATE-----MIIEBTCCAu2gAwIBAgIJAP4RVtdGKZoMMA0GCSqGSIb3DQEBCwUAMIGYMQswCQYDVQQGEwJFUzEP
MA0GA1UECAwGTVVSQ0lBMQ4wDAYDVQQHDAVDRVVUSTEjMCEGA1UECgwaSE9QVSBDRVJUSUZJQ0FURSBBVVRIT1J
JVFkxIzAhBgNVBAMMGkhPUFUgQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR4wHAYJKoZIhvcNAQkBFg92aWxlbGxhQGhvc
HUuZXUwHhcNMTkwMzA4MDc1MTIyWhcNMjkwMzA1MDc1MTIyWjCBmDELMAkGA1UEBhMCRVMxDzANBgNVBAg
MBk1VUkNJQTEOMAwGA1UEBwwFQ0VVVEkxIzAhBgNVBAoMGkhPUFUgQ0VSVElGSUNBVEUgQVVUSE9SSVRZMSMwI
QYDVQQDDBpIT1BVIENFUlRJRklDQVRFIEFVVEhPUklUWTEeMBwGCSqGSIb3DQEJARYPdmlsZWxsYUBob3B1LmV1MII
BIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzw99dx4DebSk7NrLbE75ZA3545M9pA5QOwlz9fa7stE1D5SQNP
Oc/tcZHc5wpQIz+U0PxvJ96G2g5KPyw+iNNTbTBnrycHd3G1Smes7vDSAp/PCT36GcXrOqtoRGQxXK7aSYUubykzOegm
P7uDdMyfcIfCMeXJR7pyPmHCljC4TJWS9rIdsGzBLJ2bOYvjWUmiBnxLf/8ZsPDqDT0GkUGRuAA/pZuS2LHDi96TIYCayY
DMxa/7RnVOMMh7dQKiFrdYhJy9RVDncz9imU+rAe23YZ1UWhi3+a2rQunvCrAEVBeRV4fo/cNo5GZdFjxNxvs4enpwm0
fTbCDMLn7iBZEwIDAQABo1AwTjAdBgNVHQ4EFgQUSSSGUn8IBtEhOlYvHYHa2YIP/iUwHwYDVR0jBBgwFoAUSSSGUn
8IBtEhOlYvHYHa2YIP/iUwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAeto98v5JlNBF70uI7Gznfmz7c+
H/c5ug1PhoTcbWh6W2nU+JNgcNxXLxPtQaGudnfv8nqJfJZpQueLRJXGJ4lbnZgEakA/KFvZZs/5ijKKWibn2YMtOqGbVNh
xiKi3TKWuoGdEfNu8dcU5Cwz8E05ABV8Wex9En8N7FP6cX6ojJk+gTlxqFsfHct1xOKZ3uc3oMOSemT018z6vwC3z84+U
DAG0/ksv+kmbbdPVLYRL2otPx2Rs9/znuTCttqLIlFoD+Vv21v2i0UDJxkJgxNIRHGefb8v2is4CpkIOOncHYqN8xjfoEhlGQo
Yq+Khe5KrAL738eg28SouwnoNAUlFA==-----END CERTIFICATE-----

- Quality Of Service for published messages: 1.
- Topic prefix for published messages: /JRI9jTbEKJQS29y3bc8BC6/SmartSpot_AQ_1

Note that after configuring a MQTT connection will be required to reboot the device. To do this you can use the LwM2M resource 3/0/4 ("Device" OMA LwM2M Object -> Reboot) from Homard.

Changelog

Version	Author	Description	Date
v1.0	Alejandro V.	Draft	07/02/2018
v1.1	David F.	Review and improvements over draft	09/02/2018
v1.2	David F.	Improved format and extra MQTT information added	26/09/2019
v1.3	David F.	Topic names modified to achieve compatibility with latest FIWARE IoTAgent-JSON version	15/10/2019
v1.4	David F.	Introduction of multi-value reporting compatible with latest FIWARE IoTAgent-JSON version	23/10/2019
v1.5	David F.	Improve explanation about FIWARE ‘attrs’ topic	20/12/2020
v1.6	Alejandro V.	Introduce ‘Using verticals for FIWARE’ section	03/01/2023
v1.7	David F.	Move to official Libelium format. Add new Vertical Data Concept organisation. Re-work	16/03/2023