Homard: OMA Resources Dashboard

SPECIFICATION DOCUMENT v1.1

Deploy the Internet of Things with Bluetooth Smart-based Applications in Hybrid, Mobile, Personal, Wearable, and Cloud Computing Environments

What's inside?

Connectivity and reliability is critical requirement for maintaining an adaptable network and accomplishing the scale, consolidation, and business continuity demanded by today's advanced applications based on the Internet of Things paradigm. HOP Ubiquitous can help you to achieve Internet of Things applications development with an agile, flexible, and efficient way to deploy and optimize application services. In addition, to integrate a wide-range of Smart devices that makes your solutions more personal and ubiquitous.

Using Homard, the HOP-Ubiquitous OMA Resource Dashboard, i.e., our IoT Platform based on OMA LwM2M Server, you can support Bluetooth Smart advanced functionalities through a simple and robust communication stack and APIs. Thereby, deploying software with Bluetooth Smart support in a transparent way, and build up your products and services over the defined application services.

Key benefits

Deploy with increased agility

Quickly and easily extend your products, gateways, and solutions with Bluetooth Smart to delivery services when and where you need it.

A solution for everybody

HOP Ubiquitous has a mission and compromise to make the Internet of Things accessible to everybody, through simple commissioning and bootstrapping mechanisms. Our mobile-driven and cloud computing solutions such as the HOP Engineer App and our coming HOPs Firmware Marketplace, offer a disruptive technology to tune and update your applications over the air through intuitive and users-friendly mechanisms.

Optimize application services more efficiently

Rapidly provision and consolidate application services on your own products and solutions with our flexible modules to extend existing systems (HOP Basic core), and extending your range of products with our HOPs for automation, security, social interactions, mHealth and Smart Cities.

Provide the ultimate in flexibility

Get the most flexible deployment options in the Internet of Things market, with support across all the key platforms and systems, covering from mobile and personal devices such as mobile phones and tables (iOS and Android OS), both private and public cloud (such as FI-WARE and OpenIoT solutions).

Future and present: ***Until our devices and technology is based on IPv6, an Internet of Things enabler protocol, we also allow support IPv4, since real word still working on IPv4.***

Components

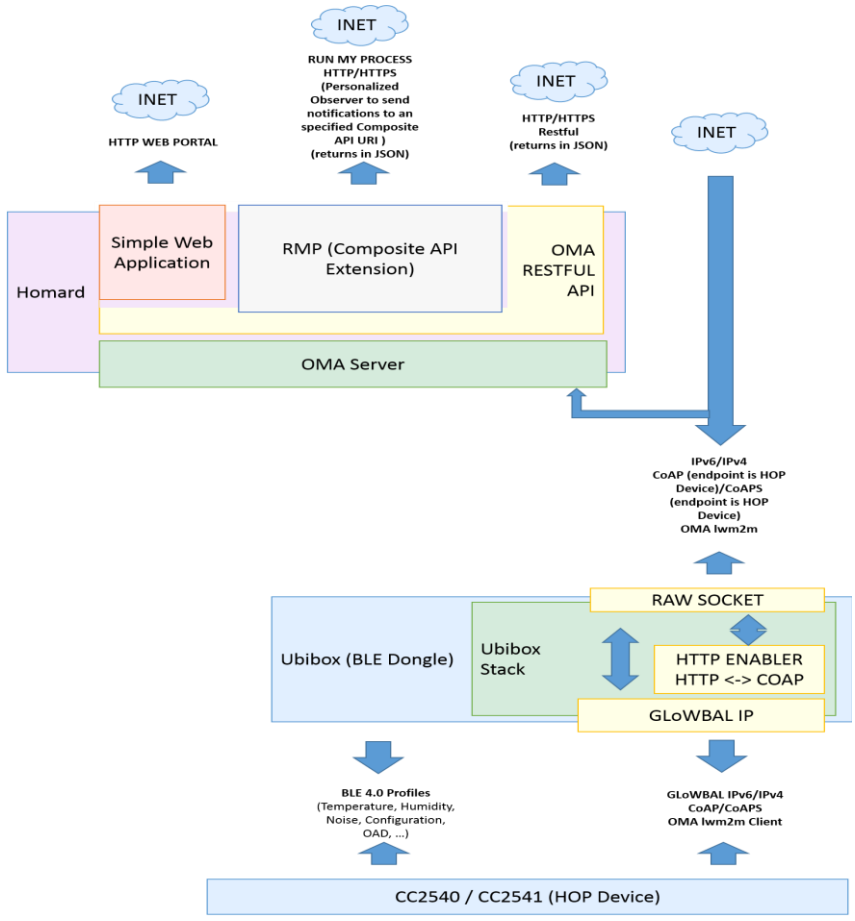
Bluetooth Low Energy, or Bluetooth Smart, is based on the version 4.0-4.2 of Bluetooth (IEEE 802.15.1). This protocol provides the physical layer for the HOP Ubiquitous solution. The HOP Ubiquitous communication stack and modules are built on top of the chipsets and physical layer libraries from Texas Instrument. Ubibox implements an advanced IPv6 header compression mechanism called GLoWBAL IP (IPv4 and IPv6 support) which brings Bluetooth Low Energy devices the opportunity to participate in more IoT use cases. In this way our devices are enabled with OMA LwM2M, IoT management protocol, to communicate with external platforms in a scalable, secure and interoperable approach.

In order to make feasible the integration with your systems, and make transparent the maintenance and evolution of these protocols. HOP Ubiquitous provides the libraries for integration into platforms/devices, our own Bluetooth Smart enablers to build your own sensor, a collection of OMA LwM2M ready sensors, and finally the Homard Server.

Homard acts as a rendezvous point to access to all your devices, where you can manage devices using a very easy to use and powerful API (described in Section Annex).

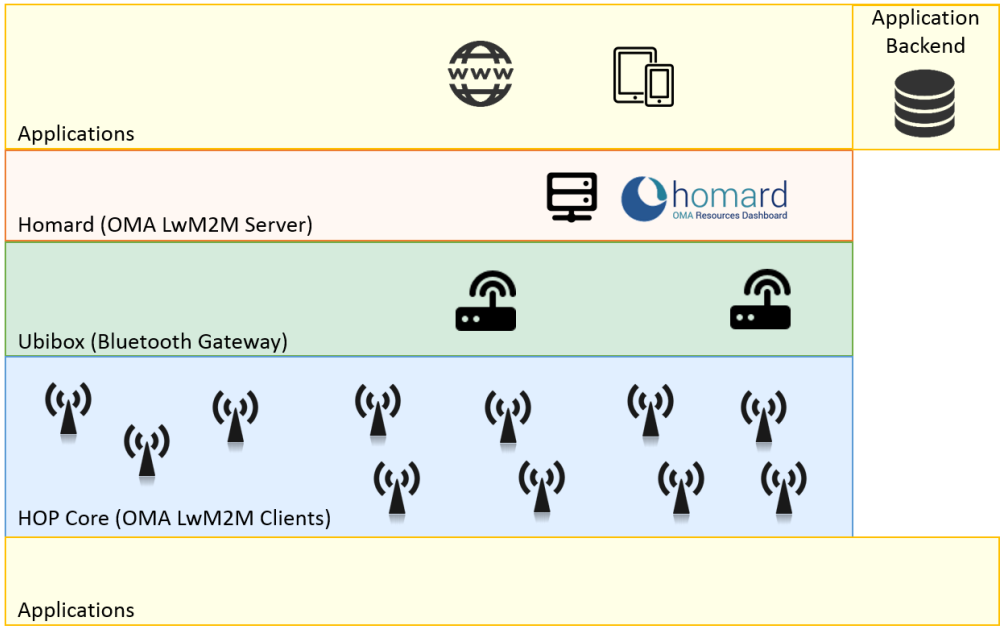
Architecture

The next diagram presents the integration of the different components and interfaces between them. Note API restful will provide a JSON interface, which is supported by the majority of the platforms and languages. Specifically ideal to work with Java-based platforms, Android, and JavaScript... technologies.



A Bluetooth Low Energy architecture ready to deployed

The common architecture offers a Homard server where devices are connected. The devices will use the Ubibox gateway to obtain access to the network via IPv6 or IPv4. Applications such as Web-based applications, Android applications, etc. can be created with based on the API offered by Homard, or with and end-to-end integration based on CoAP/OMA LwM2M primitives.



API Restful

Homard offers in top of the OMA LwM2M server a Restful services such as mechanism to communicate with the OMA LwM2M server. This HTTP/HTTPS API Restful allow users to manage devices connected to the server. The API resides on the port 8080, this port can be used also to access and use a simple web page application which also makes use of the API.

Launch the OMA LwM2M server / API services / simple-web-page-application is really easy:
java –jar homard.jar

The main API endpoint is the “/api/clients”, which allow show connected devices, and manage each of them. Inherit from OMA, the devices exposes a set of objects, which can contain one or more instances, and each instance, contains the final resources. Some objects allow multiple instances (such as a Digital I/O Object) and others only one (such as Device Object). Numbers are used to identify the tuple (Object ID / Instance ID / Resource ID).

Information returned by the API is encapsulated into a JSON data structure, since is the best way to communicate with high level applications (commonly developed in Java, JavaScript and Android.

Information returned by the API is encapsulated into a JSON data structure, since is the best way to communicate with high level applications (commonly developed in Java, JavaScript and Android).

Returned objects are formed in JSON, they are detailed on the “ANNEX: JSON Objects” section.

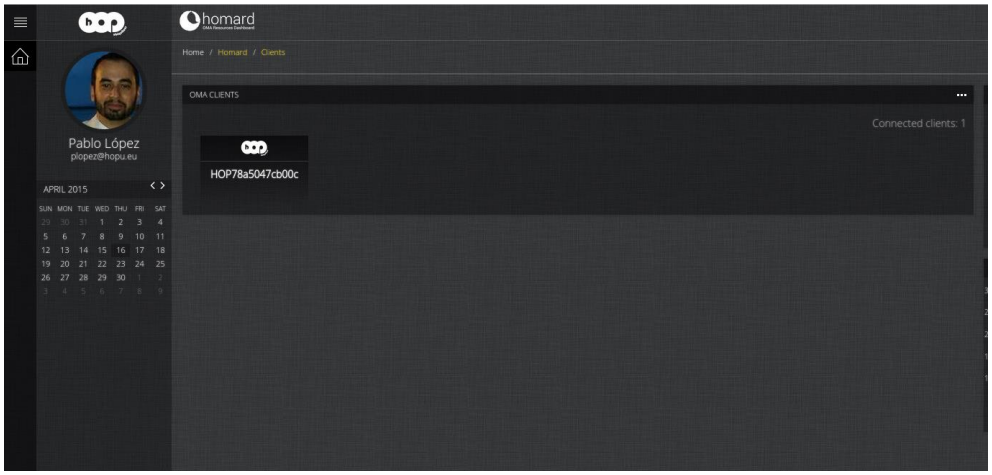
Next table shows combinations available by the Restful API:

Http Path	Method	Description	Result
/api/clients	GET	Returns all OMA clients connected to HOMARD	[{client1}, {client2}, … {clientN}]
/api/clients/CLIENT_ID	GET	Returns the the client description for a CLIENT_ID	{client}
/api/clients/CLIENT_ID/x/y	GET	Returns resource /x/y from client CLIENT_ID	[{resource1}, {resource2}, … {resourceN}]
/api/clients/CLIENT_ID/x/y/z	GET	Returns the value of the resource of id=z from client CLIENT_ID	{id, value}
/api/clients/CLIENT_ID/x/y/z	POST	Executes the resource /x/y/z on client CLIENT_ID	{ status : “CHANGED” } or { status : “BAD REQUEST” }
/api/clients/CLIENT_ID/x/y/z	PUT	Changes value of the resource on client CLIENT_ID. Content type need to be application/json and in the request body {“id”: 2, “value”: new_value}	{ status : “CHANGED” } or { status : “BAD REQUEST” }
/api/clients/CLIENT_ID/x/y	DELETE	Deletes the resource /x/y on client CLIENT_ID	{status: “DELETED”}
/api/clients/CLIENT_ID/x/y/z/ observe	POST	Keeps track on the resource /x/y/z	{id, value}
/api/clients/CLIENT_ID/x/y/z/ observe	DELETE	Stops the monitoring on resource /x/y/z	
/api/clients/CLIENT_ID/x/y/z/ observe/m/M/URL	POST	Keeps track on the resource /x/y/z and, when the resource value is between ‘m’ and ‘M’, posts its value on ‘URL’	{id, value}
/api/clients/CLIENT_ID/x/y/z/ observe/m/M/URL	DELETE	Stops the composite monitoring on resource /x/y/z	

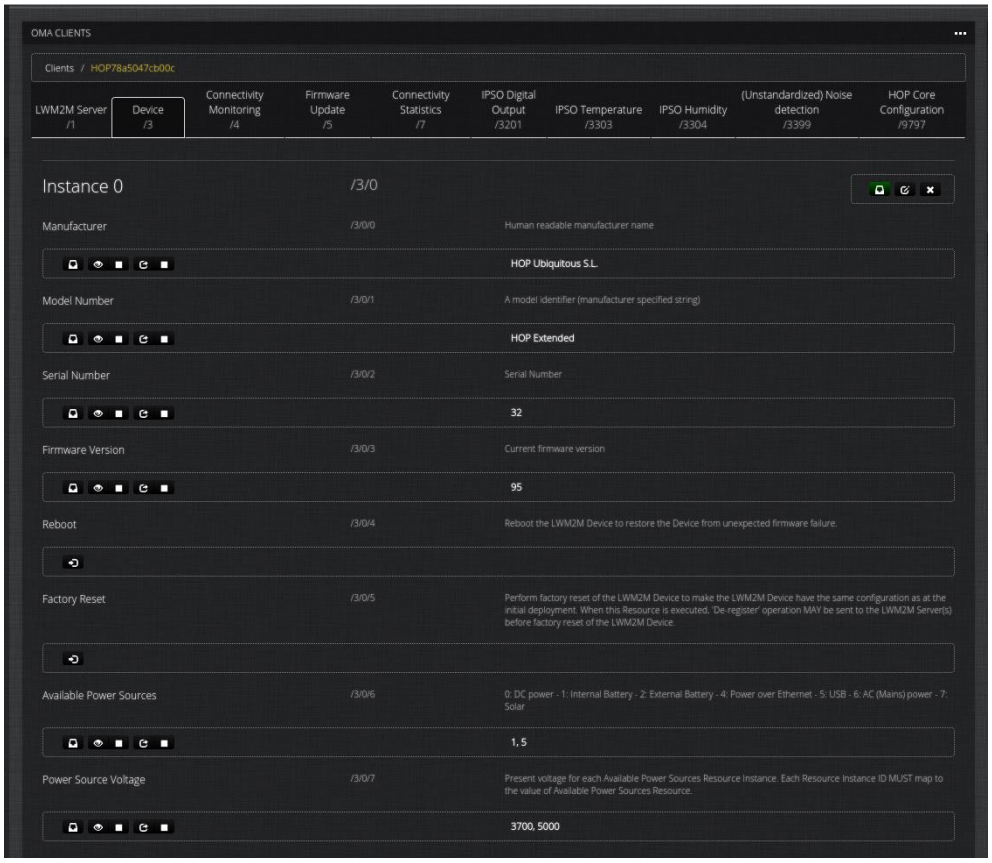
Simple Web Application

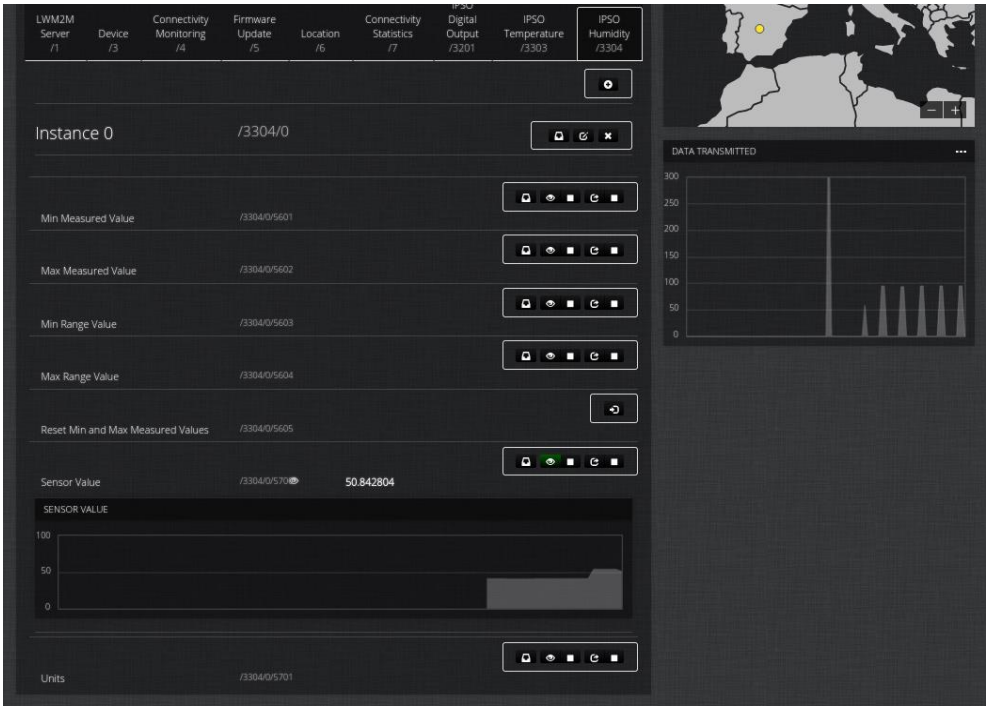
A web interface also in this ports allow manage devices in a visual way, this is a good visual way to understand how the API works, and also a good future way to manage this devices, we still working on enhance and provide new features on Homard. The following pictures present a selection of screenshots from the Homard Web Interface.

Homard is a continuously evolving platform with the inclusion of new semantics, interfaces to other platforms, and OMA Web Objects support. Note that we are active members and collaborators to make Homard aligned with oneM2M, ETSI, IPSO Alliance, OMA, and other bodies such as Industrial Internet Consortium (IIC).AllSeen and OpenInterconnect Consortium (OIC).



The client list shows Bluetooth Low Energy devices connected to the OMA LwM2M server. Click on them to access to the different objects offered by the OMA LwM2M client. Each tab is an object. You will be able to check the powerful API offered by Homard, since all this interface is based on the API, and you will be able to read entire object instances, specific resources, execute resources, enable/disable observers, subscribe for notifications based on defined conditions, …





ANEX: API Restful Examples

This section presents examples for the different resources available in the API Restful in order to give a reference of the JSON responses.

Query
GET /api/clients (array of clients)
Result


```
[
  {
    "endpoint": "HOP-Sensor-Debug",
    "registrationId": "MTKt5ejSeu",
    "registrationDate": "2015-02-16T01:06:27+01:00",
    "address": "/127.0.0.1:54604",
    "lwM2MmVersion": "1.0",
    "lifetime": 120,
    "bindingMode": "U",
    "rootPath": "/",
    "objectLinks": [
      {
        "url": "/1/0",
        "attributes": {},
        "objectId": 1,
        "objectInstanceId": 0
      },
      {
        "url": "/3",
        "attributes": {},
        "objectId": 3
      },
      {
        "url": "/4",
        "attributes": {},
        "objectId": 4
      },
      {
        "url": "/5",
        "attributes": {},
        "objectId": 5
      },
      {
        "url": "/3201/0",
        "attributes": {},
        "objectId": 3201,
        "objectInstanceId": 0
      },
      {
        "url": "/3201/1",
        "attributes": {},
        "objectId": 3201,
        "objectInstanceId": 1
      },
      {
        "url": "/3303/0",
        "attributes": {},
        "objectId": 3303,
        "objectInstanceId": 0
      },
      {
        "url": "/3304/0",
        "attributes": {},
        "objectId": 3304,
        "objectInstanceId": 0
      },
...
```

Query

GET / api /clients/ HOP-Sensor-Debug (client)

Result


```
{
  "endpoint": "HOP-Sensor-Debug",
  "registrationId": "MTKt5ejSeu",
  "registrationDate": "2015-02-16T01:06:27+01:00",
  "address": "/127.0.0.1:54604",
  "lwM2MmVersion": "1.0",
  "lifetime": 120,
  "bindingMode": "U",
  "rootPath": "/",
  "objectLinks": [
    {
      "url": "/1/0",
      "attributes": {},
      "objectId": 1,
      "objectInstanceId": 0
    },
    {
      "url": "/3",
      "attributes": {},
      "objectId": 3
    },
    {
      "url": "/4",
      "attributes": {},
      "objectId": 4
    },
    {
      "url": "/5",
      "attributes": {},
      "objectId": 5
    },
    {
      "url": "/3201/0",
      "attributes": {},
      "objectId": 3201,
      "objectInstanceId": 0
    },
    {
      "url": "/3201/1",
      "attributes": {},
      "objectId": 3201,
      "objectInstanceId": 1
    },
    {
      "url": "/3303/0",
      "attributes": {},
      "objectId": 3303,
      "objectInstanceId": 0
    },
    {
      "url": "/3304/0",
      "attributes": {},
      "objectId": 3304,
      "objectInstanceId": 0
    },
    {
      "url": "/3399/0",
      "attributes": {},
      "objectId": 3399,
      "objectInstanceId": 0
    }
  ],
  "secure": false
}
```

Query
GET /api/clients/CLIENT_ID/3303/0 (Temperature Object)
Result

{
"status": "CONTENT",
"content": {
"id": 0,
"resources": [
{
"id": 5604,
"value": 125
},
{
"id": 5700,
"value": 21.6512
},
{
"id": 5601,
"value": 21.393799
},
{
"id": 5602,
"value": 22.337608
},
{
"id": 5603,
"value": -40
}
]
}
}

Query
POST /api/clients/CLIENT_ID/3303/0/5700/observe/-5/15/https://runmyproces.com/composite/api/3023023232
Result for POST REQUEST

{
"status": "CONTENT",
"content": {
"id": 5700,
"value": 21.9086
}
}

Result in the specified URL is received in text/plain
<float value> Example: 21.122313

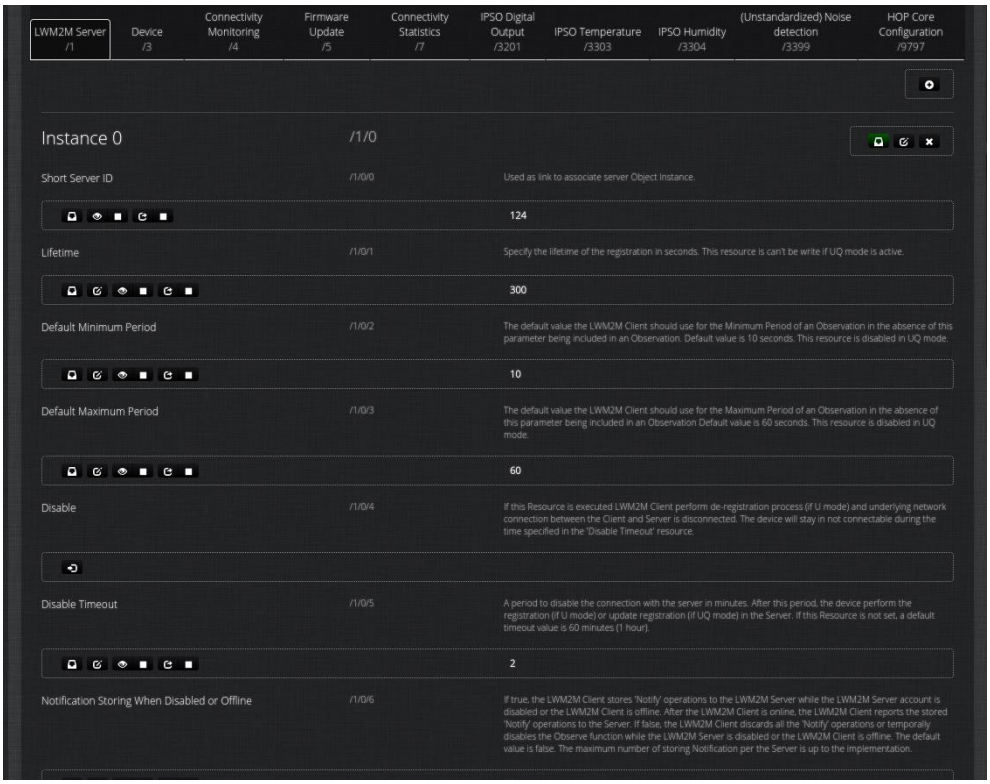
More Information

Some information related with this topic can be find the in "Getting Started Guide" of “HOP OMA LwM2M Lab Kit”.

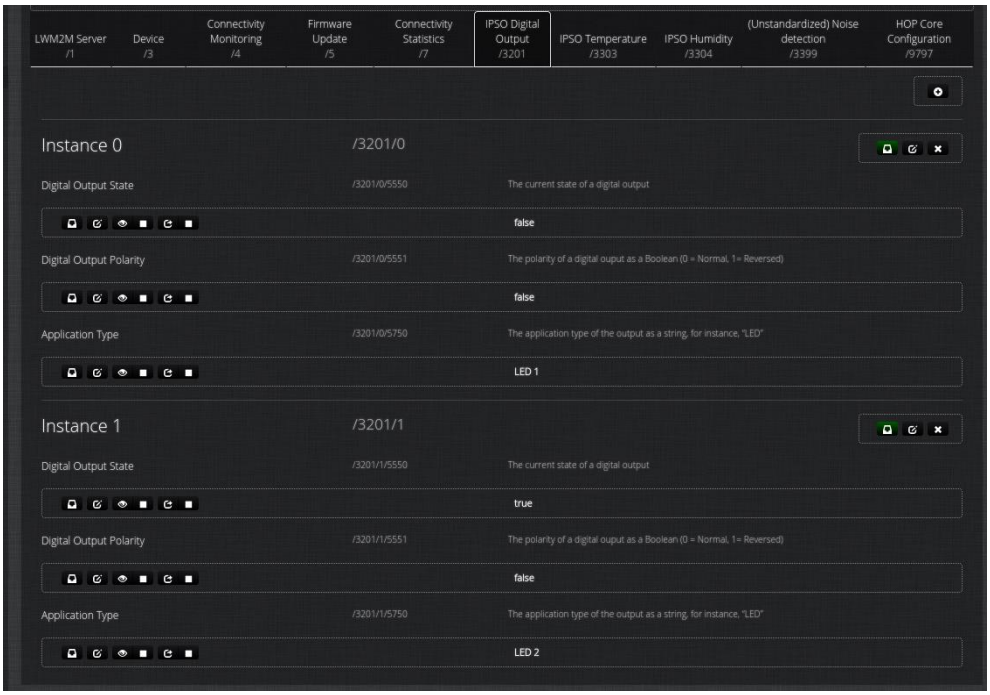
https://github.com/HOP-Ubiquitous/GettingStarted

ANNEX: Simple Web Application screenshots

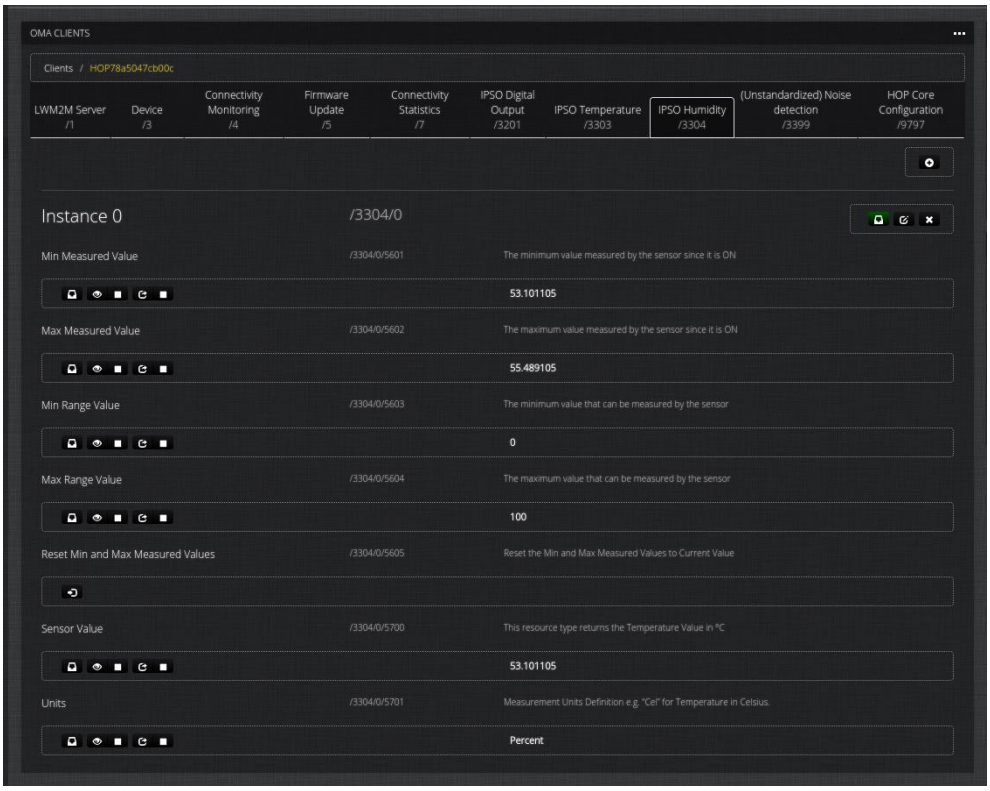
Devices offers several OMA objects depending of the firmware/device, these are some objects viewed from the Simple Web Application, but of course, there are more!



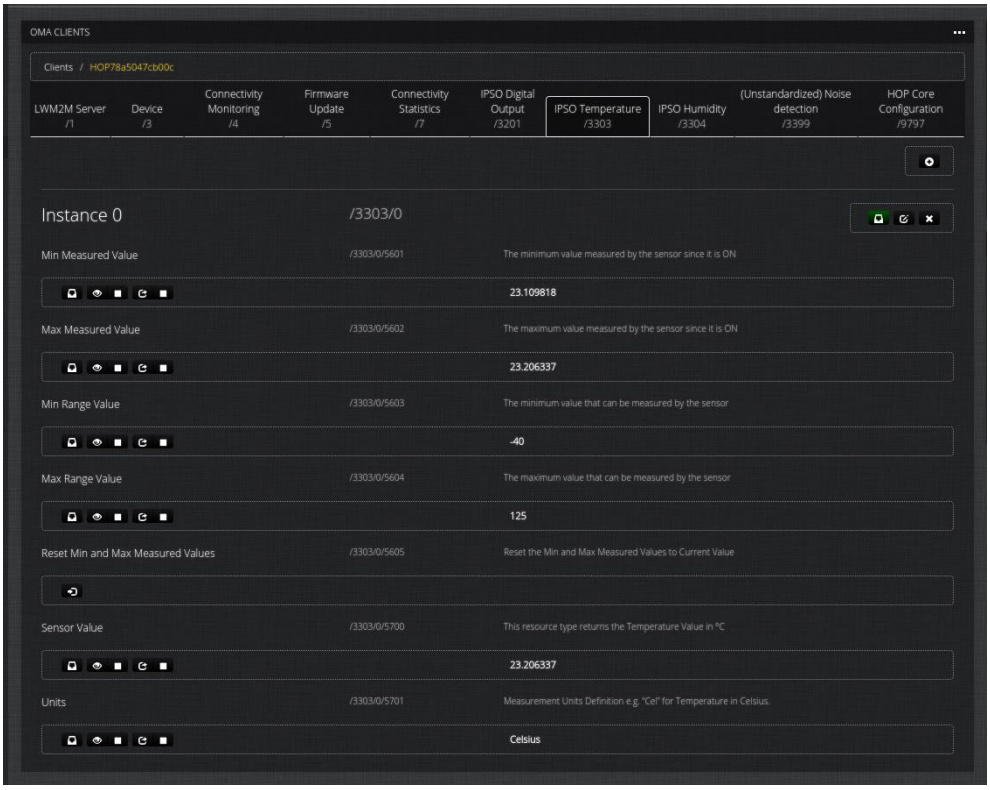
Screenshot 1. Object Server: Configure client-server interaction



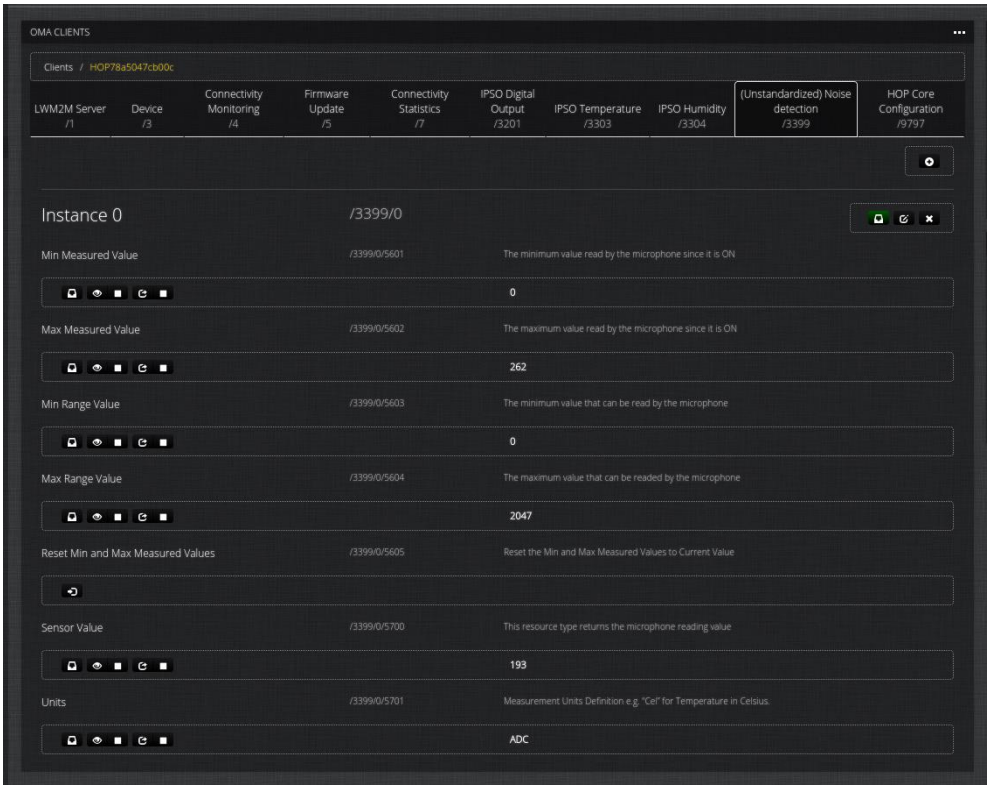
Screenshot 2. Object Digital I/O: Use digital outputs to manage external devices



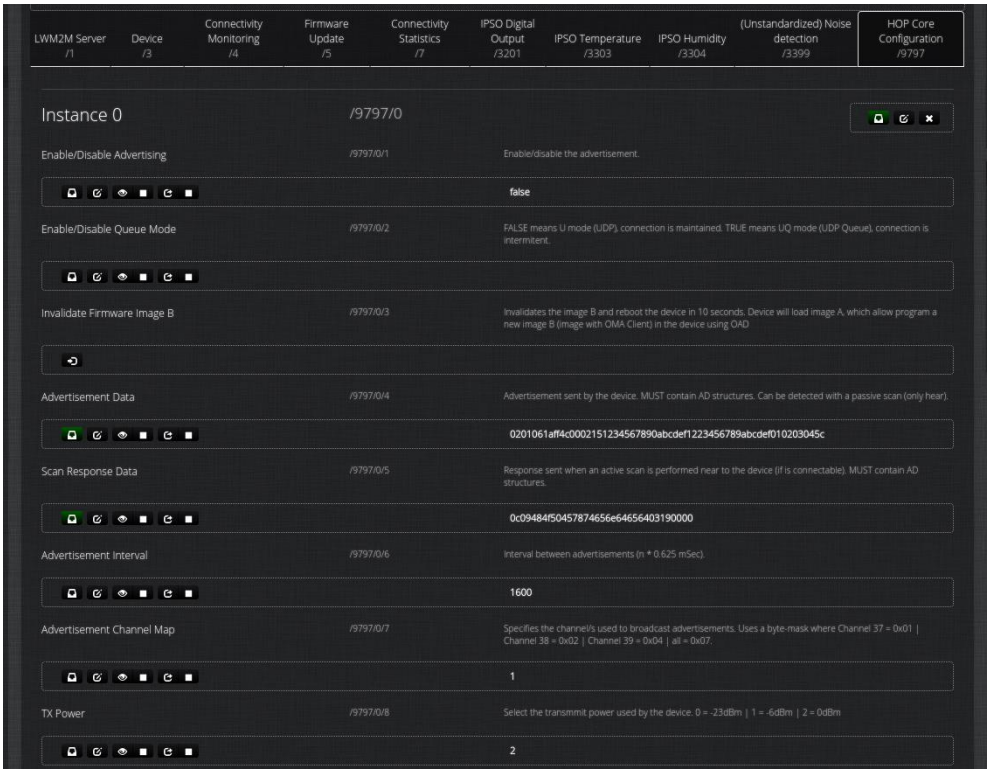
Screenshot 3. Object Humidity: Complete information relative with the ambient humidity.



Screenshot 4. Object Temperature: Complete information relative with the ambient temperature.



Screenshot 5. Object Noise detection: Complete information relative with the ambient noise.



Screenshot 6. Object HOP Configuration: Configurations regarding advertisements, channels, TX power,...

Version history

David Fernández Ros (davidfr@hopu.eu)	v0.8 - Creation
David Fernández Ros (davidfr@hopu.eu)	v0.9 - Improvements
Antonio Jara Valera (jara@hopu.eu)	v1.0 - Text review
David Fernández Ros (davidfr@hopu.eu)	v1.1 – Minor fix and new section

HOP Ubiquitous, S.L. Luis Bunuel, 6, CEUTI, MURCIA, ES-30562, ESPANA – SPAIN www.hopu.eu

Support +34 968978958
support@hopu.eu
Sales
order@hopu.eu
Spain

Switzerland +41(0)21 5880064
USA +1 650 6441356

Brazil +55 1149508796

ES-B73816589

®
A new world of opportunities

©2015 HOP Ubiquitous S.L.. All rights reserved.