HOP Bluetooth Smart Devices (HBSD)

SPECIFICATION DOCUMENT v1.0

Deploy the Internet of Things with Bluetooth Smart-based Applications in Mobile Hybrid, 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 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.

HOP Bluetooth Smart Devices (HBSD) are intelligent devices based on Bluetooth Low Energy which provide the advanced capability of IoT integration through the newest and most adopted IoT protocol: OMA Lightweight Machine to Machine (LwM2M).

The features of the Bluetooth Low Energy in conjunction with OMA LwM2M allow develop multi-purpose devices with long lifetime such as sensors, actuators, multi-advertisement beacons, and even combination of them.

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).

Device's core module: HOP Core

Texas Instruments CC2541 module is currently the main chipset for HOP Bluetooth Smart Devices (HBSD). This tiny single-device integrates the logic controller, communication stack, and application libraries from HOP Ubiquitous. This chipset is one of the most cost effective Bluetooth Low Energy chipsets in the market, reaching a competitive cost for critical mass production and with an excellent trade-off for their consumption and power performance.

From the TI CC254x family we use CC2541. This revision provides I2C interface in order to integrate additional sensors (e.g. SHT21 for temperature and humidity), at the same time that it offers lower power consumption in comparison with the CC2540.

The main module of our devices called HOP Core is composed by the CC2541 chipset in conjunction with an external antenna connector to support external antennas, in order to increase the coverage, and a 512 Kbytes flash memory dedicated for over-the-air programming, firmware upgrade and persistent data storage.

HOP core can be extended with several sensors via analogue, digital and I2C interfaces, the number of possible use cases and application is almost infinite.

Even for more specific and under demand integrations, it is offered the integration of legacy technologies by serial interfaces such as RS232. This is an open possibility if the device is feed using other ways rather than batteries.

These are the main features of the core module:

- Texas Instruments CC2541 chip based
  - Programmable Output Power up to 0 dBm.
  - 8-KB RAM with retention in all power modes
  - I2C interface
  - 12-Bit ADC with 8 channels and configurable resolution
  - 8 digital input/output channels
  - Two USARTS with support for several serial protocols
  - Active-Mode RX Down to: 14.7 mA
  - Active-Mode TX (0 dBm): 14.3 mA
  - Power Mode 1 (4-µs Wake-Up): 270 µA
  - Power Mode 2 (Sleep Timer On): 1 µA
  - Power Mode 3 (External Interrupts): 0.5 µA
- External 512 Kbytes flash memory for firmware update and storage.
- External antenna support

Robust BLE Stack

Since devices will work 24/7/365 an important feature is use a debugged and robust stack. For this, our devices uses the latest release of the TI Bluetooth Low Energy 4.0 (see image 1).

Our software is added on the top of this Bluetooth Low Energy stack as tasks or profiles. HOP Ubiquitous provides several firmware options which can be changed or updated easily through our IoT Devices management platform (Homard).

In addition, the devices can receive updates including the full stack to ensure provide the most robust and updated software in case of Texas Instruments publish new stack updates or enhancements via the firmware upgrade over the air.

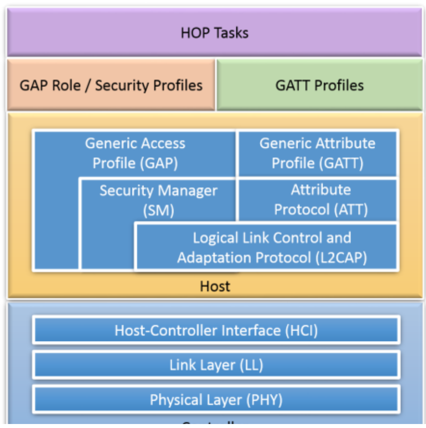


Image 1. Texas Instruments 4.0 BLE Stack

Change or update the firmware remotely will take between 15 and 40 minutes.

Software

Bluetooth Low Energy technology allows to create a huge amount of solutions for IoT, physical web, beacons, home automation etc. Each solution has different requirements or needs. For that reason, it is necessary to offer a wide range of firmware to satisfy these different needs. In HOP Ubiquitous we provide a baseline of 4 firmware with different characteristics to cover the needs presented in the creation of new products based on this technology. Table 1 presents a summary about the available firmware.

	Firmware 1: **Proximity**	Firmware 2: **Emergency & Panic**	Firmware 3: **Eddystone & iBeacon**	Firmware 4: **IoT sensor & actuator**
Proximity Profile	•			
Battery Profile	•			
HOP Over-the-Air Programming Profile	•	•	•	
Eddystone URL Profile			•	
iBeacon Profile			•	
LwM2M Profile				•
Available from the HOP IoT Platform (Homard web and REST API)			•	•
Capable to integrate any kind of analogue, digital or I2C sensor/actuator				•
Higher CC254x Power Mode Achieved	PM2	PM3	PM2	PM2

Table 1. Firmware summary

<u>Firmware 1: Proximity</u>
This version is focused on proximity-based solutions such as "children locator", "key finders", “indoor navigation”, etc. This firmware contains the Bluetooth Proximity Profile which is used by smartphone proximity applications to check and calculate the distance between the smartphone and the device.

<u>Firmware 2: Emergency / Panic button</u>
This version is dedicated to emergency and panic button solutions. There is no specific Bluetooth profile since device behaviour is send a specific broadcast message when a button is pressed. Since the most part of time is sleeping (until button is pressed), it is possible to use the more conservative power mode (PM3). Thereby, even small coin cell batteries will have a very large duration. An Ubibox or a smartphone application will detect this emergency beacons to carry out a notification to the user about the emergency.

<u>Firmware 3: Google Eddystone + Apple iBeacon</u>
This version is focused on advertising solutions (beacons) and physical web. The device is only connectable during the first minute. The device configuration can be done from any iBeacon or Eddystone smartphone application or receiving a configuration from our IoT platform (also via REST API) using the Ubibox gateways.

The remote management via the IoT Platform (Homard) is possible since these devices are mapped by the Ubibox gateway to Homard. Thereby, common commercial Bluetooth Low Energy devices are also managed from the HOP IoT platform (Homard). Since Eddystone does not provide a profile to configure an Eddystone UID for now, this value can be set in HOP Devices with this firmware from the HOP IoT Platform (Homard) and its REST API.

-

Bluetooth Low Energy 4.2 will offer an universal support of IPv6 and Internet. HOP Ubiquitous is providing it since Bluetooth 4.0

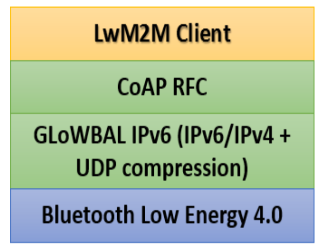
To improve the detection of broadcast messages from smartphones, the "Eddystone + iBeacon" firmware interspersed the different broadcasts as follows:

- Eddystone URL, Eddystone UID, iBeacon UID.
- Eddystone URL, Eddystone UID, iBeacon UID.
- Eddystone URL, Eddystone UID, iBeacon UID.
- Eddystone TLM type 0 (Telemetry).

Firmware 4: IoT sensor & actuator

Bluetooth Low Energy is also a powerful technology to create a new generation of sensors and actuators integrated into the Internet of Things. To achieve this, It is offered ne of the most actual and innovator IoT protocol: OMA Lightweight M2M. Its architecture allows achieve large scale deployments and the exposition of device generic resources makes easiest the creation of new products based on this technology.

To achieve provide IPv4/IPv6 to Bluetooth Low Energy devices we use a header compression method similar to the used by 6LoWPAN devices, which was a research objective for us in the past. GLoWBAL IPv6 over Bluetooth Low Energy creates and exchange unique hashes formed by the source IP, source port, destination IP and destination port. So a common IPv6 header of 40 bytes plus 8 bytes of UDP header is reduced to 7-8 bytes while is transported in the BLE connection

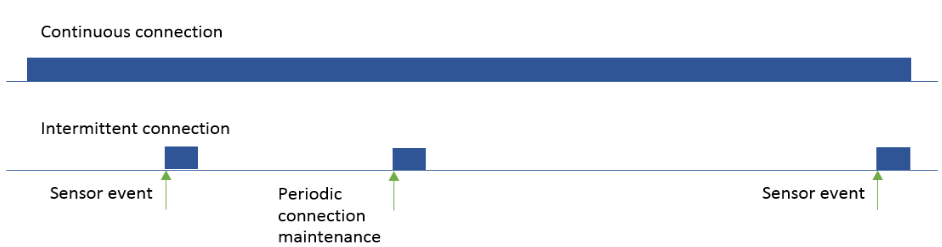


The use of OMA LwM2M and offering end-to-end IP connectivity enhances the scalability and performance of the solution with respect other alternatives based on proxies or brokers. For example, end-to-end connectivity reduces the impact and needs of synchronization methods that in case of requiring the use of a "connectable mode" (send a connectable advertisement) will impact over a 21% of the battery lifetime. In addition, OMA LwM2M enables the intelligence in the device to offer the subscription to events, anomalies detection and end-to-end integration between physical events with cloud-enabled processes.

In addition to end-to-end connectivity, the mapping and proxy of devices is also supported for telemetry and very low power use-cases, in fact the Ubibox is capable to map Bluetooth Low Energy Profiles to virtual LwM2M Clients.

Bluetooth Low Energy fits perfectly with the vision of OMA LwM2M for sleeping devices where devices does not need to stay constantly connected to internet to communicate with them. This allows to develop sensors which are usually "disconnected" from Internet until an event occurs such as detect gas leaks, water flood and temperature over a threshold.

For the use cases, where a continuous connection is required, such as gas or water valves manipulators, which should be closed immediately after detect a gas leak or a water flood. For that reason, constant / continuous or intermittent / periodic connections are supported to be used depending of the solution requirements. Next image shows the difference between both kinds of connection.



Check UbiBox documentation to learn more about continuous and intermittent connections management

A continuous connection can be maintained without a high consumption by sacrificing the connection speed (expanding the connection event interval). However, a continuous connection will consume a connection slot in the Ubibox gateway. Note that Ubibox is capable to manage 3 to 5 connections depending of the hardware and simultaneous connections shares the connection parameters.

Devices based on intermittent connections will carry out periodic connection maintenances against the OMA LwM2M server. The interval between these periodic maintenances can be set up in a range between 2 minutes to 8 hours. Intermittent connections will occupy a slot in the Ubibox gateway during less than 1 minute; thereby an Ubibox is capable to manage dozens of intermittent sensors at the same time, with a response time dependent of the number of devices and the maintenance interval of these.

For both, Intermittent and constant connections, the devices expose a set of objects, instances and resources which can be read, observed, modified or executed not only from the IoT Platform (Homard), but also from any platform or application via the a REST API.

Next table enumerates some available object/instances and a short description about his utility, extended information about objects and the complete list is found in the ANNEX 1.

Object name	Object description
Temperature	Exposes temperature values extracted from a SHT21 via I2C
Humidity	Exposes temperature values extracted from a SHT21 via I2C
Analog Input	Allow configure and use an Analog to Digital Converter (ADC), supports different resolutions (8,10,12,14) and voltage references (1,25, VDD), ... This object is useful to read values from sensors such as lux-meters, fluvial, …
Digital Input	This object allow receive interruptions signals over a digital input, useful for sensors such as motion sensors, door/window open/close detection, …
Digital Output	This object allow manage digital outputs to work with relays, electro-valves, …
HOP Configuration	This object exposes configurations such as the type of connectivity (constant or intermittent), 3 different broadcasts messages, interval between broadcast, …

Power consumption

Device consumption depends directly on the device and its configuration from the specific use case / application. Therefore, power consumption and lifetime estimation is defined and calculated in details based on the requirements of the solution and one of the more important questions is "Which is the desired device lifetime?", since based on this requirement different configurations of the device, duty cycles and battery size can be offered.

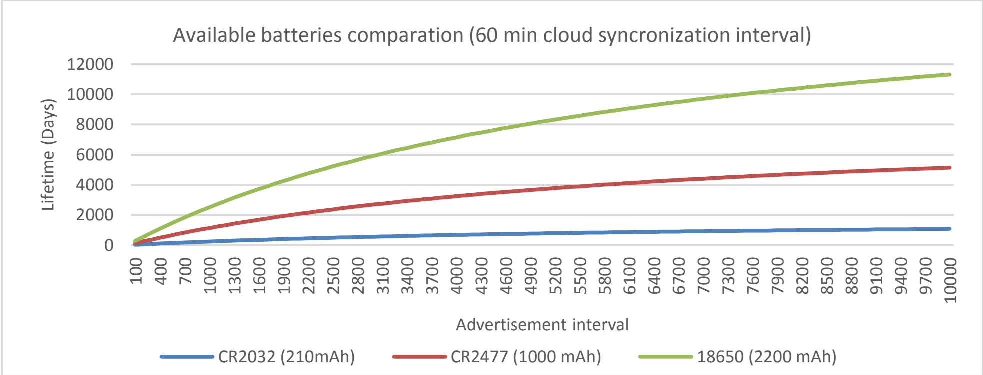
Use cases such as "panic button" achieves a huge lifetime (years) using a small coin cell battery such as the CR2032 (over 200mAh). The same battery could be also used for a proximity marketing solution (beacon), but if you desire use a high rate interval between advertisements (<250ms) in your marketing solution you will need a larger coin cell battery such as the CR2477 (over 1000mAh)

In case of sensing devices connected to internet, the coin cell battery CR2477 could be enough depending of the sensors imposed by the use case, but if the requirement specifies a long lifetime or use sensors with a high current consumption, an 18650 battery can be used.

Power consumption calculator is offered with the details of the sensors selected to enable an easy and ad-hoc module design

To facilitate design of the devices, HOP Ubiquitous offers a calculator which help to select the type of battery and the configuration needed to meet the solution requirements.

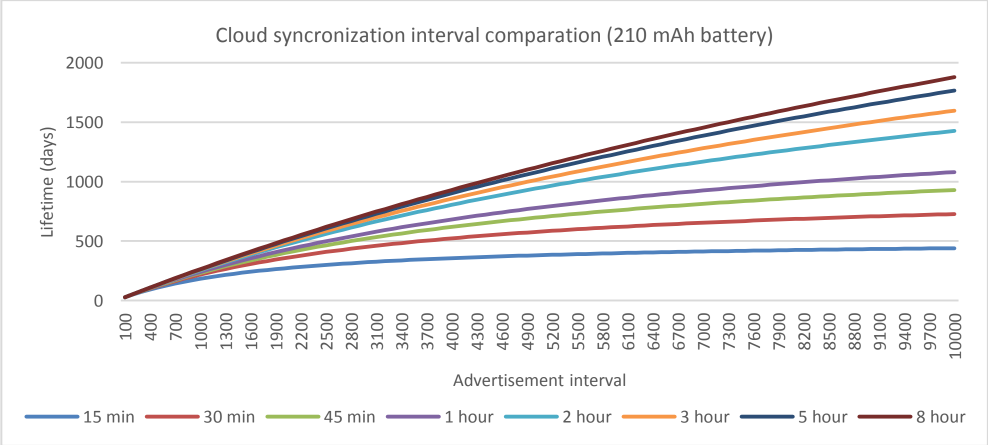
Taking in account the common divisor of all HOP Ubiquitous solution, the HOP Device main module (HOP Core) and the firmware with higher consumption demand, the IoT firmware, it is possible draw the longer lifetime which could be achieved with the different batteries available to create an end consumer solution.



The example shown in the figure illustrates the lifetime achieved depending of the interval between announcements of a device using a cloud synchronization interval of 60 minutes. This may help calculate the hardware/software configuration for connected proximity marketing solutions or as a reference (higher battery performance) for device with sensors.

The previous figure does not show the importance of a main configuration in IoT firmware's such as is the cloud synchronization interval. This is other important question if the device's goal is obtain a periodic update of a value.

Next figure is an example of how differs the lifetime of a device in function of the period for cloud synchronization.



Such as presented in the previous figure, the main influential parameter is the advertisement interval, which can be configured in an interval between 100ms and 10 seconds. While proximity marketing devices will use an interval between 100ms and 1,5 seconds, sensing devices could use larger values greater than 5 seconds to achieve a long lifetime.

It is also possible not to transmit advertisements (offering a very high increase of battery lifetime). However, advertisement transmission is an interesting feature for different purposes. In details, it helps to check the devices proximity during and after deployments to ensure the connectivity, it also allows to check the presence of all devices even when they are not connected to monitor reliability. Finally, we are also offering innovative ways to transmit data and information to the platform without connect the device to the cloud, in order to offer a higher performance, i.e., include useful data in the advertisements.

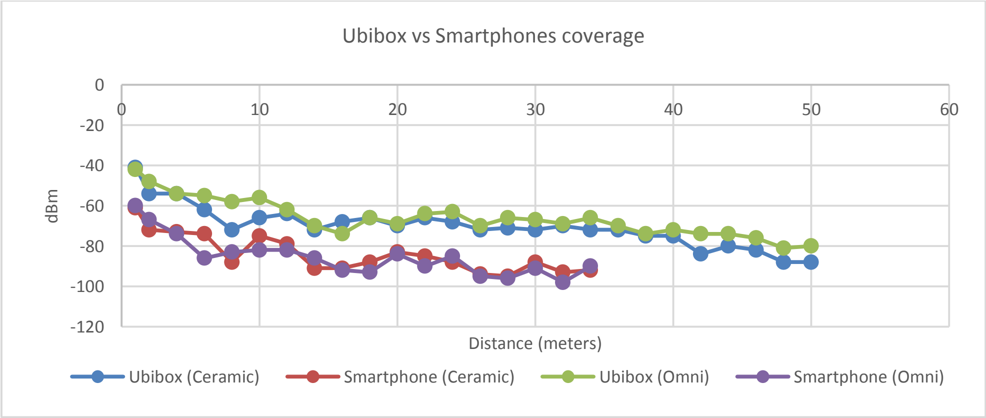
Coverage

Since each deployment scenario is different, the coverage is the main preoccupation on deployments, but in Bluetooth Low Energy the use of the advertisements in collaboration with the mobile tools offered by HOP Ubiquitous to analyse the connectivity between Ubibox and devices during and after the deployment.

Homard offers a scan function to analyse and check the visible devices by the gateway and quality of the link. Thereby, remotely can be checked the visibility of the deployed devices to ensure all devices are in range with the Ubibox. This means that is possible to check in real time (using the REST API) to the Ubibox which devices are in range and which is the quality of the link.

Since Ubibox will act as a gateway for Bluetooth Low Energy devices, it is mandatory to use a stable hardware with a good radiofrequency performance. This is because Ubibox uses an external antenna which increases the radio range and quality.

From the devices point of view, depending of the use case, will be recommendable use an integrated ceramic antenna (better performance compared with PCB antenna) or even an omnidirectional external antenna. A comparison between the receptions for both kind of antennas is shown on the next figure. It presents that external antenna performance is significantly better than the ceramic antenna.



Since there is no references of other Bluetooth Low Energy gateways, this figure also shows the coverage of the UbiBox with respect to one of the best smartphones in the market supporting Bluetooth Low Energy (Google Nexus 5). These results demonstrate the radio quality of the Ubibox gateway. In numbers, the signal benefit of the Ubibox router compared with a smartphone is around -20dBm higher, which offers around 20 additional meters.

ANNEX 1: LwM2M Objects

The exposition of resources in LwM2M Clients is as follows: the LwM2M Client expose objects, each object represent a group of resources related and the object could be unique instance (HOP configuration) or multiple instance (example: "Analog Input Object" where a device could expose several analogic input pins). Inside of each object/instance there are resources, which could be read/write/observed separately. It's also possible read a complete instance of an object.

In LwM2M, resources are represented by the tuple X-Y-Z where:

- X is the number of the object
- Y is the number of the instance
- Z is the number of the resource

For example, the resource 3202/1/5600 corresponds to the resource "Analog Input Current Value" in the instance "1" (P_04 pin for example) in the object "IPSO Analog Input".

There are two main organisms working on the production of objects:

- Open Mobile Alliance: They created the main core of objects needed for the bootstrap, provisioning, management and basic information such as connection statistics, location,...
- IPSO Alliance: They are defining new objects more related with sensors such as Temperature, Humidity, Digital Input, Digital Output, Analog Input,...

From HOP Ubiquitous we implement objects from both entities and also participate in the creation and definitions of objects together with both organizations. Next tables exposes the objects which are currently defined and implemented inside of the IoT firmware. Note, not all objects are visible for customers since they are managed by bootstrap servers (security object) and not all objects are needed for certain solutions.

HOP configuration Object

Creator	Object ID	Name	Multi-instance	Mandatory
HOPU	**9797**	HOP configuration	No	No

Resource ID	Name	Type	Operations	Instance Type	Description
1	Enable/Disable Advertising	Boolean	RW	single	Disable advertisements while device is in sleeping UQ
4	Advertisement Data	Opaque (0-31 bytes)	RW	single	Must contain AD Structures.
5	Scan Response Data	Opaque (0-31 bytes)	RW	single	Must contain AD Structures
6	Advertisement Interval	Integer	RW	single	Advertisement between announcements. (n * 0.625 mSec)
7	Advertisement Channel Map	Integer	RW	single	Mask -> Channel 37 = 0x01 | Channel 38 = 0x02 | Channel 39 = 0x04 | all = 0x07
8	TX Power	Integer	RW	single	TX Power selected where 0 = -23dBm, 1 = -6dBm, 2 = 0dBm
9	Current connection interval	Integer	R	single	(n * 1.25ms). Range: 7.5 msec to 4 seconds (0x0006 to 0x0C80)
10	Current slave latency	Integer	R	single	
11	Current timeout value	Integer	R	single	(n * 10ms). Range: 100ms to 32 seconds (0x000a - 0x0c80)
13	Battery	Integer	R	single	Battery percent available
14	Firmware version	String	R	single	Current firmware version
15	Secondary Advertisement Data	Opaque (0-31 bytes)	RW	single	Secondary optional advertisement. Set 00 will disable this advertisement.
16	Tertiary Advertisement Data	Opaque (0-31 bytes)	RW	single	Tertiary optional advertisement. Set 00 will disable this advertisement.

Analog Input Object

Creator	Object ID	Name	Multi-instance	Mandatory
IPSO	**5600**	Analog Input	Yes	No

Resource ID	Name	Type	Operations	Instance Type	Description
5600	Analog Input Current Value	Float	R	single	The current value of the analogue input.
5601	Min Measured Value	Float	R	single	The minimum value measured by the sensor since it is ON
5602	Min Measured Value	Float	R	single	The maximum value measured by the sensor since it is ON
5603	Min Range Value	Float	R	single	The minimum value that can be measured by the sensor
5604	Max Range Value	Float	R	single	The maximum value that can be measured by the sensor
5605	Reset Min and Max Measured Values	-	E	-	Reset the Min and Max Measured Values to Current Value
5750	Application Type	String	RW	single	The application type of the sensor or actuator as a string.
5751	Sensor type	String	R	single	The type of the sensor, for instance PIR type
5778	Select read reference	Integer	RW	single	0 = 1.25v | 1 = Analog IN 7 | 2 = AVDD
5779	Select read resolution	Integer	RW	single	Accepted ADC resolution values: 8, 10, 12 and 14
5780	Wake up from UQ threshold	Integer	RW	single	On UQ devices, a value higher than 0 will be used as threshold for a wake up condition. 0 disables this wake up condition.

Digital Input Object

Creator	Object ID	Name	Multi-instance	Mandatory
IPSO	**3200**	Digital Input	Yes	No

Resource ID	Name	Type	Operations	Instance Type	Description
5500	Digital Input State	Bool	R	single	The current state of a digital input
5501	Digital Input Counter	Integer	R	single	The cumulative value of active state detected.
5502	Digital Input Polarity	Bool	RW	single	The polarity of the digital input as a Boolean (0 = Normal, 1= Reversed)
5503	Digital Input Debounce Period	Integer	RW	single	The debounce period in ms.
5504	Digital Input Edge Selection	Integer	RW	single	The edge selection as an integer (1 = Falling edge, 2 = Rising edge, 3 = Both Rising and Falling edge)
5505	Digital Input Counter Reset	-	E	-	Reset the Counter value
5750	Application Type	String	RW	Single	The application type of the sensor or actuator as a string, for instance, “Air Pressure”
5751	Sensor Type	String	R	single	The type of the sensor, for instance PIR type
5780	Wake up from UQ	Bool	RW	single	On UQ devices, a value 1 will wake up device to connect with the LwM2M server.

Digital output Object

Creator	Object ID	Name	Multi-instance	Mandatory
IPSO	**3201**	Digital output	Yes	No

Resource ID	Name	Type	Operations	Instance Type	Description
5550	Digital Output State	Bool	RW	single	The current state of a digital output
5551	Digital Output Polarity	Bool	RW	single	The polarity of a digital output as a Boolean (0 = Normal, 1= Reversed)
5750	Application Type	String	RW	Single	The application type of the output as a string, for instance, “LED”
5777	Digital Output Timeout	Integer	RW	single	0 means manually manage turn off the GPIO 0. If seconds >0, then GPIO auto turn off the output after this certain seconds

Humidity Object

Creator	Object ID	Name	Multi-instance	Mandatory
IPSO	**3304**	Humidity	Yes	No

Resource ID	Name	Type	Operations	Instance Type	Description
5700	Sensor Value	Float	R	single	This resource type returns the Temperature Value in °C
5701	Units	String	R	single	Measurement Units Definition e.g. “Cel” for Temperature in Celsius.
5601	Min Measured Value	Float	R	single	The minimum value measured by the sensor since it is ON
5602	Max Measured Value	Float	R	single	The maximum value measured by the sensor since it is ON
5603	Min Range Value	Float	R	single	The minimum value that can be measured by the sensor
5604	Max Range Value	Float	R	single	The maximum value that can be measured by the sensor
5605	Reset Min and Max Measured Values	-	E	-	Reset the Min and Max Measured Values to Current Value
5620	Auto-Reader State	Bool	RW	single	Allow start and stop the auto-reader which will take measures internally along time.
5621	Auto-Reader Interval	Integer	RW	single	Read or edit the auto-reader interval between measures

Temperature Object

Creator	Object ID	Name	Multi-instance	Mandatory
IPSO	**3303**	Temperature	Yes	No

Resource ID	Name	Type	Operations	Instance Type	Description
5700	Sensor Value	Float	R	single	This resource type returns the Temperature Value in °C
5701	Units	String	R	single	Measurement Units Definition e.g. “Cel” for Temperature in Celsius.
5601	Min Measured Value	Float	R	single	The minimum value measured by the sensor since it is ON
5602	Max Measured Value	Float	R	single	The maximum value measured by the sensor since it is ON
5603	Min Range Value	Float	R	single	The minimum value that can be measured by the sensor
5604	Max Range Value	Float	R	single	The maximum value that can be measured by the sensor
5605	Reset Min and Max Measured Values	-	E	-	Reset the Min and Max Measured Values to Current Value
5620	Auto-Reader State	Bool	RW	single	Allow start and stop the auto-reader which will take measures internally along time.
5621	Auto-Reader Interval	Integer	RW	single	Read or edit the auto-reader interval between measures

Presented objects are the most useful for customers since they allow create their solutions by managing bluetooth features (such as advertisements), sensors, analogue and digital pins, ... However, there are more objects defined, but they are less important for customers since they contains device information, statistics, ...

The next set of objects are not a priority for customers and they show only the most important resources.

LWM2M Server Object

Creator	Object ID	Name	Multi-instance	Mandatory
OMA	**1**	LWM2M Server	Yes	No

Resource ID	Name	Type	Operations	Instance Type	Description
4	Disable	-	E	-	Disconnect a device from server during a time specified in resource 5.
5	Disable Timeout	Integer	RW	single	Specifies how many time device will be disconnected from LwM2M server
8	Registration Update Trigger	-	E	-	Request for a new registration update

Device Object

Creator	Object ID	Name	Multi-instance	Mandatory
OMA	3	Device	No	No

Resource ID	Name	Type	Operations	Instance Type	Description
0	Manufacturer	String	R	single	Human readable manufacturer name
1	Model Number	String	R	single	A model identifier (manufacturer specified string)
2	Serial Number	String	R	single	Serial number
3	Firmware Version	String	R	single	Current firmware version
4	Reboot	-	E	-	Reboot the LwM2M device
5	Factory Reset	-	E	-	Perform factory reset of the LwM2M Device to have the same configuration as at the initial deployment
6	Available Power Sources	Integer	R	multiple	0: DC Power 1: Internal Battery 2: External Battery 4: Power over Ethernet 5: USB 6: AC (Mains) power 7: Solar
7	Power Source Voltage	Integer	R	multiple	Present voltage for each Available Power Sources
8	Power Source Current	Integer	R	multiple	Present current for each Available Power Source
9	Battery Level	Integer	R	single	Current battery level as a percentage. This value is only valid when the value of Available Power Sources resource is 1.
10	Memory Free	Integer	R	single	Estimated current available amount of storage space which can store data and software in the LwM2M Device (kilobytes)
11	Error Code	Integer	R	multiple	0: No error 1: Low battery power 2: External power supply off … 4: Low received signal strength … 5: Out of memory …
12	Reset Error Code	-	E	-	Delete all error code
13	Current Time	Time	RW	single	Current UNIX time of the LwM2M Client
14	UTC Offset	String	RW	single	Indicates the UTC offset
15	Time zone	String	RW	single	Indicates in which time zone is located
16	Supported Binding and Modes	String	R	single	Indicates which bindings and modes are supported by the LwM2M Client.

Connectivity Monitoring Object

Creator	Object ID	Name	Multi-instance	Mandatory
OMA	4	Connectivity Monitoring	No	No

Resource ID	Name	Type	Operations	Instance Type	Description
0	Network Bearer	Integer	R	single	0~20 are Cellular Bearers 21~40 are Wireless Bearers 22: Bluetooth Network 41~50: are Wireline Bearers
3	Link Quality	Integer	R	single	This contains received link quality

Location Object

Creator	Object ID	Name	Multi-instance	Mandatory
OMA	6	Location Statistics	No	No

Resource ID	Name	Type	Operations	Instance Type	Description
0	Latitude	String	RW	single	The decimal notation of latitude
1	Longitude	String	RW	single	The decimal notation of longitude
2	Altitude	String	RW	single	The decimal notation of altitude in meters above sea level
3	Uncertainty	String	RW	single	The accuracy of the position in meters
5	Timestamp	Time	R	single	The timestamp of when the location measurement was performed.

Statistics Object

Creator	Object ID	Name	Multi-instance	Mandatory
OMA	**7**	Connectivity Statistics	No	No

Resource ID	Name	Type	Operations	Instance Type	Description
2	Tx Data	Integer	R	single	Indicate the total amount of data transmitted during the collection period (kilobyte)
3	Rx Data	Integer	R	single	Indicate the total amount of data received during the collection period (kilobyte)
4	Max Message Size	Integer	R	single	The maximum message size that is used during the collection period (byte)
5	Average Message Size	Integer	R	single	The average message size that is used during the collection period (bye)

Ubibox configuration Object

Creator	Object ID	Name	Multi-instance	Mandatory
HOPU	**9798**	Ubibox configuration	No	No

Resource ID	Name	Type	Operations	Instance Type	Description
0	Authorized devices	String	RW (only authorized users can write)	single	Allow set devices whitelist
1	OMA server	String	R	single	Destination OMA server
2	Ubibox IPv6	String	RW	single	Shows the ubibox address where prefix is get
3	Ubibox IPv6 netmask	String	RW	single	Shows the ubibox netmask
4	Ubibox device history resource	String	R	single	Shows the Ubibox device RSSI history resource list

.

Version history

David Fernández Ros (davidfr@hopu.eu)	v0.1 – Document creation
David Fernández Ros (davidfr@hopu.eu)	v0.9 – Document drafting
Antonio J. Jara (jara@hopu.eu)	V1.0 – Final review