IoT Practical Instruction
Office Lighting System

Leila F. Rahman
Eindhoven, 15 December 2016
Agenda

• Protocol description
• Tutorials run-through
System Deployment

User App

HTTP or CoAP

Broker (Area Controller)

LWM2M and MQTT
LWM2M and MQTT

Light Device
Sensor Device

User (Open Office Area)

Building Network Infrastructure

Building Management Server

Building Manager App

HTTP

Cloud Service

HTTP or CoAP

HTTP or CoAP

Cloud Service

Cloud Service
LWM2M Overview

LWM2M Server

Interfaces
- Bootstrap
- Client Registration
- Device Management & Service Enablement
- Information Reporting

Stack
- Efficient Payload
- CoAP Protocol
- DTLS Security
- UDP or SMS Bearer

LWM2M Client

Objects

M2M Device

LWM2M Client

Object 0, Instance 0
- Resource 1: R, W, E
- Resource 2: R

Access Control List
- ACL: Server1=R, W
LWM2M in Broker and End Devices

Broker (Laptop)

- LWM2M Server (Wakaama, Leshan or mbed)
- MQTT Broker (Mosquitto)
- mDNS/DNS-SD Server (Avahi)

Light Device (Raspberry Pi + SenseHat)

- LWM2M Client (Wakaama, Leshan or mbed)
- MQTT Client (Paho)
- mDNS/DNS-SD Client (Avahi)

Sensor Device (Raspberry Pi + Camera)

- LWM2M Client (Wakaama, Leshan or mbed)
- MQTT Client (Paho)
- mDNS/DNS-SD Client (Avahi)

Lighting Service
- Configurations
- Update Service

Face Detection Service
- Configuration

Hardware Abstraction Layer (Camera Python Libraries)
LWM2M Objects

- LWM2M Objects can be acquired from:
  - LWM2M objects defined by OMA (Appendix E of LWM2M Technical Specification)
  - Private objects by using free range id 10241 – 32768 ([http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry](http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry))
OMA LWM2M Objects

Registered with Open Mobile Naming Authority

<table>
<thead>
<tr>
<th>Object</th>
<th>Object ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWM2M Security</td>
<td>0</td>
</tr>
<tr>
<td>LWM2M Server</td>
<td>1</td>
</tr>
<tr>
<td>Access Control</td>
<td>2</td>
</tr>
<tr>
<td>Device</td>
<td>3</td>
</tr>
<tr>
<td>Connectivity Monitoring</td>
<td>4</td>
</tr>
<tr>
<td>Firmware</td>
<td>5</td>
</tr>
<tr>
<td>Location</td>
<td>6</td>
</tr>
<tr>
<td>Connectivity Statistics</td>
<td>7</td>
</tr>
</tbody>
</table>

E.6 LWM2M Object: Firmware Update

Description
This LWM2M Object enables management of firmware which is to be updated. This Object includes installing firmware package, updating firmware, and performing actions after updating firmware. A reboot of the device must occur for taking into account the new successfully installed firmware.

After reboot of the device:
- the “State” Resource must be at Downloaded state (2) if the “Package” Resource contains a valid Package which has not been successfully installed yet, or at Idle state (0) otherwise.
- the Update Result must maintain the relevant value it has before Device reboot.

Object definition

<table>
<thead>
<tr>
<th>Name</th>
<th>Object ID</th>
<th>Instances</th>
<th>Mandatory</th>
<th>Object URN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Update</td>
<td>5</td>
<td>Single</td>
<td>Optional</td>
<td>urn:oma:lwm2m:oma.5</td>
</tr>
</tbody>
</table>

Resource definitions

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Operations</th>
<th>Instances</th>
<th>Mandatory</th>
<th>Type</th>
<th>Range or Enumeration</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Package</td>
<td>W</td>
<td>Single</td>
<td>Mandatory</td>
<td>Opaque</td>
<td></td>
<td></td>
<td>Firmware package</td>
</tr>
<tr>
<td>1</td>
<td>Package URI</td>
<td>W</td>
<td>Single</td>
<td>Mandatory</td>
<td>String</td>
<td>0-255 bytes</td>
<td></td>
<td>URI from where the device can download the firmware package by an alternative mechanism. As soon the device has received the Package URI it performs the download at the next practical opportunity.</td>
</tr>
<tr>
<td>2</td>
<td>Update</td>
<td>E</td>
<td>Single</td>
<td>Mandatory</td>
<td>none</td>
<td>no argument</td>
<td></td>
<td>Updates firmware by using the firmware package stored in Package, or, by using the firmware downloaded from the Package URI. This Resource is only executable when the value of the State Resource is Downloaded.</td>
</tr>
</tbody>
</table>
Object and Resource Examples

- /3/0/4 Device, Reboot
- /3/0/5 Device, Factory Reset
- /3/0/9 Device, Battery Level
- /5/0/1 Firmware Update, Package
- /5/0/2 Firmware Update, Update
- /6/0/0 Location, Latitude
- /7/0/2 Connectivity Statistics, Tx Data

- IPSO examples (see references)
Object and Resource IDs with the IPSO alliance

- IPSO alliance: IP for Smart Objects
- IPSO Smart Objects Starter Pack
  - Set of 18 smart objects for use with CoAP
  - Based on LWM2M object model
  - Not dependent on full LWM2M framework
    - Use CoAP directly
- IPSO Smart Objects Expansion Pack
  - 16 Common Template sensors
    - Pressure, Power, Distance, …
  - 6 Special Template sensors
    - Energy, Color, GPS Location, …
  - 5 Actuators
    - Buzzer, Display, …
  - 6 Control switch types
    - Up/Down, Push button, Multiple axis joystick, …

Johan J. Lukkien, j.j.lukkien@tue.nl
TU/e Informatica, System Architecture
10. IPSO Object: Temperature

Description: This IPSO object should be used with a temperature sensor to report a temperature measurement. It also provides resources for minimum/maximum measured values and the minimum/maximum range that can be measured by the temperature sensor. An example measurement unit is degrees Celsius (°Celsius).

Object Info:

<table>
<thead>
<tr>
<th>Object</th>
<th>Object ID</th>
<th>Object URN</th>
<th>Multiple Instances?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPSO Temperature</td>
<td>3203</td>
<td>urn:oma:lwm2m:ext.3203</td>
<td>Yes</td>
<td>Temperature sensor, example units = °C</td>
</tr>
</tbody>
</table>

Resources:

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Access Type</th>
<th>Multiple Instances?</th>
<th>Mandatory</th>
<th>Type</th>
<th>Range or Enumeration</th>
<th>Units</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Value</td>
<td>5700</td>
<td>R</td>
<td>No</td>
<td>Mandatory</td>
<td>Float</td>
<td></td>
<td></td>
<td>Last or Current Measured Value from the Sensor</td>
</tr>
<tr>
<td>Units</td>
<td>5701</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>String</td>
<td></td>
<td></td>
<td>Measurement Units Definition e.g. “°C” for Temperature in Celsius.</td>
</tr>
<tr>
<td>Min</td>
<td>5601</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>Float</td>
<td>Same as</td>
<td>Same as</td>
<td>The minimum</td>
</tr>
</tbody>
</table>

IPSO Smart Objects Expansion Pack

Table 1 summarizes the objects defined by this Technical Guideline.

<table>
<thead>
<tr>
<th>Type</th>
<th>Object ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>3316</td>
</tr>
<tr>
<td>Current</td>
<td>3317</td>
</tr>
<tr>
<td>Frequency</td>
<td>3318</td>
</tr>
<tr>
<td>Depth</td>
<td>3319</td>
</tr>
<tr>
<td>Percentage</td>
<td>3320</td>
</tr>
<tr>
<td>Altitude</td>
<td>3321</td>
</tr>
<tr>
<td>Load</td>
<td>3322</td>
</tr>
<tr>
<td>Pressure</td>
<td>3323</td>
</tr>
<tr>
<td>Loudness</td>
<td>3324</td>
</tr>
<tr>
<td>Concentration</td>
<td>3325</td>
</tr>
<tr>
<td>Conductivity</td>
<td>3326</td>
</tr>
<tr>
<td>Power</td>
<td>3328</td>
</tr>
<tr>
<td>Power Factor</td>
<td>3329</td>
</tr>
<tr>
<td>Rate</td>
<td>3346</td>
</tr>
<tr>
<td>Distance</td>
<td>3330</td>
</tr>
<tr>
<td>Energy</td>
<td>3331</td>
</tr>
<tr>
<td>Direction</td>
<td>3332</td>
</tr>
<tr>
<td>Time</td>
<td>3333</td>
</tr>
<tr>
<td>Gyrometer</td>
<td>3334</td>
</tr>
<tr>
<td>Color</td>
<td>3335</td>
</tr>
<tr>
<td>GPS Location</td>
<td>3336</td>
</tr>
<tr>
<td>Positioner</td>
<td>3337</td>
</tr>
<tr>
<td>Buzzer</td>
<td>3338</td>
</tr>
<tr>
<td>Audio Clip</td>
<td>3339</td>
</tr>
<tr>
<td>Timer</td>
<td>3340</td>
</tr>
<tr>
<td>Addressable Text Display</td>
<td>3341</td>
</tr>
<tr>
<td>On/Off Switch</td>
<td>3342</td>
</tr>
<tr>
<td>Push Button</td>
<td>3347</td>
</tr>
<tr>
<td>Level Control</td>
<td>3343</td>
</tr>
<tr>
<td>Up/Down Control</td>
<td>3344</td>
</tr>
<tr>
<td>Multistate Selector</td>
<td>3348</td>
</tr>
<tr>
<td>Multiple Axis Joystick</td>
<td>3345</td>
</tr>
</tbody>
</table>

6.6 IPSO Object: Multiple Axis Joystick

Description: This IPSO object can be used to report the position of a shuttle or joystick control. A digital input is provided to report the state of an associated push button.

Object Info:

<table>
<thead>
<tr>
<th>Object ID</th>
<th>Object URN</th>
<th>Multiple Instances?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3345</td>
<td>urn.oma.1.wm2m.m.334 5</td>
<td>Yes</td>
<td>Used for a 3-axis (shuttle) control, 2 axis control, or 3 axis control</td>
</tr>
</tbody>
</table>

Resources:

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Access Type</th>
<th>Multiple Instances?</th>
<th>Mandatory Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Input State</td>
<td>5500</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>Boolean</td>
</tr>
<tr>
<td>Digital Input Counter</td>
<td>5501</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>Integer</td>
</tr>
<tr>
<td>X Value</td>
<td>5702</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>Float</td>
</tr>
<tr>
<td>Y Value</td>
<td>5703</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>Float</td>
</tr>
<tr>
<td>Z Value</td>
<td>5704</td>
<td>R</td>
<td>No</td>
<td>Optional</td>
<td>Float</td>
</tr>
<tr>
<td>Application Type</td>
<td>5750</td>
<td>R,W</td>
<td>No</td>
<td>Optional</td>
<td>String</td>
</tr>
</tbody>
</table>

http://www.ipso-alliance.org/ipso-community/resources/smart-objects-interoperability/
## Object Instances in Light Device

<table>
<thead>
<tr>
<th>Object</th>
<th>Object ID</th>
<th>Object Instance ID</th>
<th>Notes</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light Profile</td>
<td>10250</td>
<td>0</td>
<td>Represent the Light Device's static and dynamic information</td>
<td>Private object</td>
</tr>
<tr>
<td>Firmware Update</td>
<td>5</td>
<td>0</td>
<td>Represents the Update for the Lighting Behavior Component of the Light Device</td>
<td>OMA LWM2M Objects</td>
</tr>
<tr>
<td>Firmware Update</td>
<td>5</td>
<td>1</td>
<td>Represents the Update for Configuration File about Ownership Priority of the Light Device</td>
<td>OMA LWM2M Objects</td>
</tr>
</tbody>
</table>
## Resources in the *Light Profile* Object
(Object Id: 10250) (cont’d)

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Operation</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Type</td>
<td>1</td>
<td>R, W</td>
<td>String</td>
<td>“Light Device”</td>
<td>The device type, in this case a “Light Device”</td>
</tr>
<tr>
<td>Light State</td>
<td>2</td>
<td>R, W</td>
<td>String</td>
<td>“USED” or “FREE”</td>
<td>Represents the state of the Light Device, whether it is in the “USED” or the “FREE” state</td>
</tr>
<tr>
<td>User Type</td>
<td>3</td>
<td>R, W</td>
<td>String</td>
<td>“USER1”, “USER2”, “USER3”</td>
<td>Represents the type of user that is using the Light Device, whether it is “USER1”, “USER2” or “USER3”</td>
</tr>
</tbody>
</table>
| Light Color   | 5           | R, W      | String   | “(r, g, b)”                                                          | Represents the color on the Sense Hat’s LED matrix. “(r, g, b)” is a string which represents a tuple containing the RGB (red, green, blue) values of the color. Each element must be an integer between 0 and 255. Examples of LED color represented in (r, g, b):
Off = (0, 0, 0)
Red = (255, 0, 0)
White = (255, 255, 255)                                                                                           |
# Resources in the *Light Profile* Object (Object Id: 10250)

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Operation</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Light</td>
<td>6</td>
<td>R, W</td>
<td>Boolean</td>
<td>True, False</td>
<td>Represents the current color intensity on the Sense Hat’s LED matrix. True = Low light mode on, False = Low light mode off</td>
</tr>
<tr>
<td>Group No</td>
<td>7</td>
<td>R, W</td>
<td>Integer</td>
<td></td>
<td>Group Number that the Light Device belongs to. Each desk is assign to a group of light(s) and a sensor.</td>
</tr>
<tr>
<td>Location X</td>
<td>8</td>
<td>R, W</td>
<td>Float</td>
<td></td>
<td>X location of the Light Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
</tr>
<tr>
<td>Location Y</td>
<td>9</td>
<td>R, W</td>
<td>Float</td>
<td></td>
<td>Y location of the Light Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
</tr>
<tr>
<td>Room ID</td>
<td>10</td>
<td>R, W</td>
<td>String</td>
<td>“Room-No”</td>
<td>The ID of the room where the Light Device is located. For example “Room-1”, “Room-2”, etc</td>
</tr>
<tr>
<td>Behavior Deployment</td>
<td>11</td>
<td>R, W</td>
<td>Boolean</td>
<td>“Broker” or “Distributed”</td>
<td>Stating which lighting behavior deployment is used, broker deployment or distributed deployment. The default value is “Distributed”. When the system decides to use behavior deployment in the broker, the value is changed to “Broker”, and the behavior deployment in the Light Device is deactivated.</td>
</tr>
</tbody>
</table>
Resources in the *Firmware Update* object (Object Id: 5)

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Operation</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>0</td>
<td>W</td>
<td>Opaque</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>PackageURI</td>
<td>1</td>
<td>W</td>
<td>String</td>
<td>0-255 bytes</td>
<td>The URL of the software update. Download and execute the update after value is changed.</td>
</tr>
<tr>
<td>Update</td>
<td>2</td>
<td>E</td>
<td>none</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>State</td>
<td>3</td>
<td>R</td>
<td>Integer</td>
<td>0-3</td>
<td>Indicates current state with respect to this firmware update. This value is set by the LWM2M client. 0: Idle 1: Downloading 3: Updating (Refer to LWM2M TS for more detail information)</td>
</tr>
<tr>
<td>Update Supported Objects</td>
<td>4</td>
<td>RW</td>
<td>Boolean</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>Update Result</td>
<td>5</td>
<td>R</td>
<td>Integer</td>
<td>0-6</td>
<td>Contains the result of downloading or updating the firmware. (Refer to LWM2M TS for more detail information)</td>
</tr>
<tr>
<td>PkgName</td>
<td>6</td>
<td>R</td>
<td>String</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>PkgVersion</td>
<td>7</td>
<td>R</td>
<td>String</td>
<td></td>
<td>Not used</td>
</tr>
</tbody>
</table>
## Object Instances in Light Device

<table>
<thead>
<tr>
<th>Object</th>
<th>Object ID</th>
<th>Object Instance ID</th>
<th>Notes</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Profile</td>
<td>10350</td>
<td>0</td>
<td>Represent the Sensor Device's static and dynamic information</td>
<td>Private Object</td>
</tr>
</tbody>
</table>
## Resources in the **Sensor Profile** Object (Object Id: 10350)

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Operation</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Type</td>
<td>1</td>
<td>R, W</td>
<td>String</td>
<td>“Sensor Device”</td>
<td>The device type, in this case a “Sensor Device”</td>
</tr>
<tr>
<td>Sensor State</td>
<td>2</td>
<td>R, W</td>
<td>String</td>
<td>“USED” or “FREE”</td>
<td>Represents the state of the Sensor Device, whether it is in the “USED” or the “FREE” state</td>
</tr>
<tr>
<td>Group No</td>
<td>4</td>
<td>R, W</td>
<td>Integer</td>
<td></td>
<td>Group Number that the Sensor Device belongs to. Each desk is assign to a group of light(s) and a sensor.</td>
</tr>
<tr>
<td>Location X</td>
<td>5</td>
<td>R, W</td>
<td>Float</td>
<td></td>
<td>X location of the Sensor Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
</tr>
<tr>
<td>Location Y</td>
<td>6</td>
<td>R, W</td>
<td>Float</td>
<td></td>
<td>Y location of the Sensor Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
</tr>
<tr>
<td>Room ID</td>
<td>7</td>
<td>R, W</td>
<td>String</td>
<td>“Room-No”</td>
<td>The ID of the room where the Sensor Device is located. For example “Room-1”, “Room-2”, etc</td>
</tr>
</tbody>
</table>

\[System Architecture and Networking Group\]
Use Case Diagram (Building Manager)

Lighting System

Commission the system
Set user account
Set Identity and Binding
Set Priority Ownership
Execute the System

Operate the system
Observe State
Request lighting usage and desk occupancy report
Update Priority Ownership
Update light behavior

Maintain the system

Lights and Sensors

Request lighting usage and desk occupancy report
Update Priority Ownership
Update light behavior

Execute the System
Use Case Diagram (Office Worker)

Lighting System

- Experience adaptive lighting based on presence
- Adjust lighting

Office Worker

Lights and Sensors
“Execute the System” use case

Registration of Light Device and its objects to the Server

Light Device (LWM2M Client)

Register ep=Light-Device-GroupNo

</10250/0>, </5/0>, </5/1>

2.01 Created

Broker (LWM2M Server)

Sensor Device (LWM2M Client)

Register ep=Sensor-Device-GroupNo

</10350/0>

2.01 Created
“Set Identity and Binding” use case

Light Device (LWM2M Client)

- Change the value of the Group No Resource in the Light Profile Object
  - Write /10250/0/7
    - Group No
    - Success

- Change the value of the Location X Resource in the Light Profile Object
  - Write /10250/0/8
    - Location X
    - Success

- Change the value of the Location Y Resource in the Light Profile Object
  - Write /10250/0/9
    - Location Y
    - Success

- Change the value of the Room ID Resource in the Light Profile Object
  - Write /10250/0/10
    - Room ID
    - Success

Broker (LWM2M Server)
"Set Identity and Binding" use case

Sensor Device (LWM2M Client)

- Change the value of the Group No Resource in the Sensor Profile Object
  
  Write /10350/0/4
  
  Group No
  
  Success

- Change the value of the Location X Resource in the Sensor Profile Object
  
  Write /10350/0/5
  
  Location X
  
  Success

- Change the value of the Location Y Resource in the Sensor Profile Object
  
  Write /10350/0/6
  
  Location Y
  
  Success

- Change the value of the Room ID Resource in the Sensor Profile Object
  
  Write /10350/0/7
  
  Room ID
  
  Success

Broker (LWM2M Server)
“Set Priority Ownership” use case

Light Device (LWM2M Client)

Write URL of configuration file PriorityOwnership.json to the PackageURI resource in the Firmware Update object

Write /5/1/1

URL of PriorityOwnership.json

Success

Broker (LWM2M Server)
OwnershipPriority.json

```json
[
    {
        "user_type": "USER1",
        "user_id": "Office-Worker-1",
        "light_color": "(255, 255, 255)",
        "low_light": false,
        "user_location_x": 2,
        "user_location_y": 1
    },
    {
        "user_type": "USER2",
        "user_id": "Office-Worker-3",
        "light_color": "(255, 0, 0)",
        "low_light": true,
        "user_location_x": 4,
        "user_location_y": 1
    },
    {
        "user_type": "USER3",
        "user_id": "Office-Worker-20",
        "light_color": "(0, 0, 255)",
        "low_light": false,
        "user_location_x": 2,
        "user_location_y": 3
    },
    {
        "user_type": "USER3",
        "user_id": "Office-Worker-25",
        "light_color": "(0, 0, 255)",
        "low_light": false,
        "user_location_x": 4,
        "user_location_y": 3
    },
    {
        "user_type": "USER3",
        "user_id": "Office-Worker-40",
        "light_color": "(0, 0, 0)",
        "low_light": false,
        "user_location_x": 2,
        "user_location_y": 5
    },
    {
        "user_type": "USER3",
        "user_id": "Office-Worker-30",
        "light_color": "(0, 255, 255)",
        "low_light": true,
        "user_location_x": 4,
        "user_location_y": 5
    }
]
```
“Observe State” use case (cont’d)

1. Observe the value changes of Light State Resource in the Light Profile Object.

   - GET /10250/0/2 Observe
   - 2.05 Content Observe
   - ... Notify
   - USED | FREE

2. Observe the User Type Resource in the Light Profile Object.

   - GET /10250/0/3 Observe
   - 2.05 Content Observe
   - ... Notify
   - USER1 | USER2 | USER3

3. Observe the Light Color Resource in the Light Profile Object.

   - GET /10250/0/5 Observe
   - 2.05 Content Observe
   - ... Notify
   - (r, g, b)
“Observe State” use case

Light Device (LWM2M Client)  
Observe the Low Light Resource in the Light Profile Object

GET /10250/0/6 Observe

2.05 Content Observe

... Notify

True | False

Sensor Device (LWM2M Client)  
Observe the value changes of Sensor State Resource in the Sensor Profile Object

GET /10350/0/2 Observe

2.05 Content Observe

... Notify

OCCUPIED

... Notify

FREE
“Update Priority Ownership” use case

Light Device (LWM2M Client)

Write URL of configuration file PriorityOwnership.json to the PackageURI resource in the Firmware Update object

Write /5/1/1

URL of PriorityOwnership.json

Success

Broker (LWM2M Server)
“Update Light Behavior” use case

Light Device (LWM2M Client) 

Write URL of the update to the PackageURI resource in the Firmware Update object

Broker (LWM2M Server)

Write /5/0/1

URL of the update

Success
“Adaptive Lighting Based on Presence” for centralized behavior deployment

Light Device (LWM2M Client)

<table>
<thead>
<tr>
<th>Change the value of the Light Color Resource in the Light Profile Object</th>
<th>Write /10250/0/5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r, g, b)</td>
<td>Success</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change the value of the Low Light Resource in the Light Profile Object</th>
<th>Write /10250/0/6</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change the value of the User ID Resource in the Light Profile Object</th>
<th>Write /10250/0/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>Success</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change the value of the Light State Resource in the Light Profile Object</th>
<th>Write /10250/0/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>USED</td>
<td>FREE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change the value of the User Type Resource in the Light Profile Object</th>
<th>Write /10250/0/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>USER2</td>
</tr>
</tbody>
</table>

Broker (LWM2M Server)
“Adaptive Lighting Based on Presence” for centralized behavior deployment

Observe the value changes of Sensor State Resource in the Sensor Profile Object

GET /10350/0/2 Observe

2.05 Content Observe

... Notify

OCCUPIED
... Notify

FREE
"Adjust Lighting" use case

<table>
<thead>
<tr>
<th>Light Device (LWM2M Client)</th>
<th>Broker (LWM2M Server)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the value of the Light Color Resource in the Light Profile Object</td>
<td>Write /10250/0/5 (r, g, b)</td>
</tr>
<tr>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>Change the value of the Low Light Resource in the Light Profile Object</td>
<td>Write /10250/0/6 True</td>
</tr>
<tr>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>Change the value of the User ID Resource in the Light Profile Object</td>
<td>Write /10250/0/4 User ID</td>
</tr>
<tr>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>Change the value of the Light State Resource in the Light Profile Object</td>
<td>Write /10250/0/2 USED</td>
</tr>
<tr>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>Change the value of the User Type Resource in the Light Profile Object</td>
<td>Write /10250/0/3 USER1</td>
</tr>
<tr>
<td></td>
<td>Success</td>
</tr>
</tbody>
</table>
Broker Architecture

Broker (Laptop)

HTTP and/or CoAP Server
- User App API
- Cloud Service API
- LWM2M Client
- Visualization (UI for Web App)

- User Authentication
- Available Light Discovery
- Storage

Lighting and Sensor Services
- LWM2M Server (Wakaama or Leshan)
- MQTT Broker (Mosquito)

mDNS/DNS-SD Server (Avahi)

+ Centralized Behavior Deployment
User App Architecture

User App or Web Browser (Mobile Phone or Laptop)
  UI (for mobile app)
  HTTP or CoAP Client

HTTP or CoAP

Broker (laptop)
  HTTP or CoAP Server
  Broker Lower Stack
Cloud Service Architecture

Cloud Service

- HTTP Server
  - Building Manager API
  - Visualization / Dashboard
- Commissioning Services
- Monitoring Services
- Update Services
- Storage Services
- Analytics / Reporting Services
- HTTP or CoAP Client
- LWM2M Server
- mDNS/DNS-SD Client (Avahi)

Broker

- HTTP and/or CoAP Server
- mDNS/DNS-SD Server (Avahi)
- Broker Lower Stack
Building Manager Web App (Laptop)

- UI
  - HTTP Client

HTTP

Cloud Service

- HTTP Server
  - Cloud Service Lower Stack
Tutorials

http://www.win.tue.nl/~lrahman/iot_2016/tutorial/

- Tutorial for Raspberry Pi with Sense Hat
- Tutorial for Raspberry Pi with Camera module
- Tutorial for txThings, an open source implementation of CoAP in Python.
- Tutorial on Avahi, an implementation of mDNS-SD, a service discovery protocol
- Tutorial for Eclipse Leshan, an open source programming framework for developing LWM2M server and client in Java.
- Tutorial for Wakaama, an open source programming framework for developing LWM2M server and client in C.
- Tutorial for mbed, a semi open source programming framework for developing LWM2M server and client in C++
- Tutorial for Paho and Mosquitto, open source implementation of MQTT client and server. Paho is available in several programming language.
- Protocol description between the end devices and the broker, which will be based on the LWM2M and MQTT specification.