Tecnologie per IoT
Lab Software Part 3

Exercise 1. Become confident with the Mosquitto clients:

- Use mosquitto_pub to publish json messages with a certain topic
- Use different (more than one) mosquitto_sub to receive the messages published by mosquitto_pub. Play with the wildcards ('+' and '#') when you subscribe for the topic.

You can use message broker provided by mosquitto (test.mosquitto.org) available at the port 1883.

Exercise 2. Develop an MQTT subscriber to receive the temperature values sent by the Arduino yun (refer to Exercise 3.3 Lab Hardware – part 3 for topics and dataformat).
Test this software by using the Arduino yun, which must be registered on Catalog (refer to Exercise 1 Lab Software – part 2).
During its boot, the MQTT subscriber must invoke web services provided by the Catalog:

i. to register itself as a new service,
ii. to retrieve information about the message broker to use (i.e. address and port)
iii. to retrieve information about the end-point (i.e. the topics in this case) used by the Arduino yun to communicate.

SUGGESTION: Exploit the Mosquitto clients (both mosquitto_pub and mosquitto_sub) to debug the communication among your applications.

Exercise 3. Develop an MQTT publisher to send actuation commands to switch on and off a led managed by the Arduino yun (refer to Exercise 3.3 Lab Hardware – part 3 for topics and dataformat).
Test this software by using the Arduino yun, which must be registered on Catalog (refer to Exercise 1 Lab Software – part 2).
During its boot, the MQTT publisher must invoke web services provided by the Catalog:

i. to register itself as a new service,
ii. to retrieve information about the message broker to use (i.e. address and port)
iii. to retrieve information about the end-point (i.e. the topics in this case) used by the Arduino yun to communicate.

SUGGESTION: Exploit the Mosquitto clients (both mosquitto_pub and mosquitto_sub) to debug the communication among your applications.

Exercise 4. Develop the remote version (i.e. running on a remote server) of the Smart home controller defined in Exercise 2.1 Lab Hardware – part 2.

This Remote Smart home controller must implement MQTT functionalities to work as publisher and subscriber. The main task to implement are:

i. register itself on the Catalog as a new service via REST,
ii. retrieve information from the Catalog about the message broker to use (i.e. address and port),
iii. retrieve information from the Catalog about the end-point (i.e. the topics in this case) used by the Arduino yun to communicate,
iv. Implement the functionalities described in bullets 1 - 6 and 8 in Exercise 2.1 Lab Hardware – part 2

The Arduino sketch must be modified accordingly to perform the following tasks:

i. register itself on the on Catalog (refer to Exercise 1 Lab Software – part 2).
ii. send information about temperature, presence and noise via MQTT
iii. receive actuation commands via MQTT
iv. receive messages to be displayed on the LCD monitor via MQTT

In the report, compare the two version of the Smart home controller and highlighting possible pro and cons.