University of Pisa

Master's Degree in

Artificial Intelligence and Data Engineering

Internet of Things

Energy consumption and Air Conditioning control system

Candidate

Lorenzo Massagli

ACADEMIC YEAR 2021/2022

1. INTRODUCTION

The goal of the project is to simulate some IoT devices that can be found in a smart home. In particular, create a system that controls the energy consumption and allows the user to remotely control the air conditioning system. The application that has been developed is capable of:

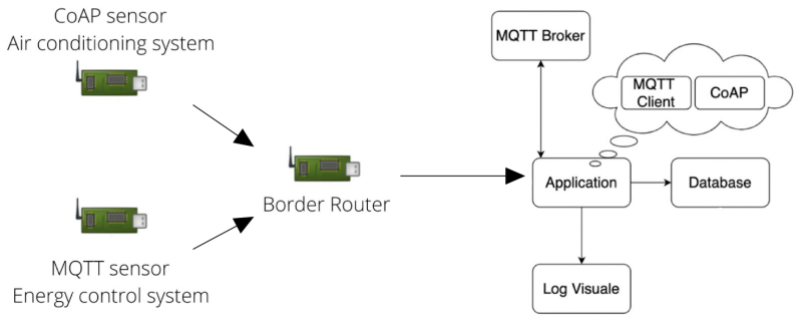
- Remote-control of air-conditioner actuators;
- Collect sensors data (energy consumption and temperature) in a database;
- Allow manual function with buttons;
- Use leds to give information, in real time, about the data collected from the sensors.

Half of the devices in the network use the CoAP protocol to expose their resources, while the other half use the MQTT protocol.

A Python collector has been developed to observe the resources of the devices and control their actuators. The collector has a command line interface capable of making the system's behavior dynamic, as variables that control the actuators can be changed live.

The project uses three Launchpad CC2650, one for the border router, one for the CoAP device and the last one for the MQTT device.

2. DEPLOYMENT



From now on we will consider the deployment on real devices.

All the devices use their LEDs to communicate the current state of the sensor they're controlling. The CoAP device uses the leds to communicate about the status of the environment temperature respect to the air conditioner temperature. If the environment temperature is different from the air conditioner ones, the led of the device is red, otherwise its green. The MQTT device uses the leds to communicate about the

status of the energy consumption respect to the limit of 3000 Watt of energy (Italian assumption). If the energy consumption is below 3000W, the led of the device is green, otherwise its red.

2.1 Energy consumption system

The energy consumption system is supported by the MQTT device.

It senses the energy consumption every 5 seconds and send the data to the MQTT collector.

The MQTT collector analyze the data received and with an actuator function manage the leds of the device.

Pressing the button of the device, is possible to activate the sound alarm function, which simulates a sound alarm warning about the energy consumption when it is above 3000 Watt.

2.2 Air conditioning

The air conditioning control is supported by a CoAP device exposing three resources: a temperature sensor, an air conditioning system and a led control system.

The functioning temperature of the air conditioning system can be changed remotely by the Python collector through a PUT request on the air_conditioner resource. By default, the AC temperature is set to 22.

When the temperature is over or under the air conditioner temperature and the air conditioner is turned on, it will increase/decrease the temperature of the environment while it will not be equal to the ac temperature.

While the environment temperature it's not equal to the air conditioner temperature, the device's led will be red. Otherwise the device's led will become green.

Pressing the button of the device will manually switch ON/OFF the air conditioning (depending on the current state of the AC).

3. DATA MANAGEMENT AND VISUALIZATION

3.1 Data Encoding

Given the well-known constraints of our devices in terms of packet size (802.15.4 has a MTU of 127 bytes), I have decided to encode the data in the JSON format avoiding fragmentation of the IPv6 datagram and less traffic in the network. If I had decided to use an XML encoding, would have resulted in fragmentation for sure, cause of the payload size.

```
CAOP JSON
{
  "node": %d,
  "temp": %d,
  "time": %lu,
  "ac_temp": %d,
}
```

```
MQTT JSON
{
  "node": %d,
  "energy_consumption": %d,
  "timestamp": %lu,
  "sound_alarm_on": %d,
}
```

3.2 Python collector

The Python collector, besides managing the data coming from the sensors and the logs, features a CLI which allows the system's manager to quickly get information about the last samples measured by the sensors and regulating the de/activation of the actuators. In the case of the air conditioner, the collector also allows the system's manager to set the temperature of the AC system.

```
----------------- COMMANDS AVAILABLE -----------------

Energy Management:
1. Get the energy consumed right now
2. Get which led is activate right now and the reason why

Air Conditioning Management
5. Get the enviroment temperature right now
6. Activate the air conditioner
7. Deactivate the air conditioner
8. Get which led is activate right now and the reason why
9. Change the air conditioner temperature

0. EXIT
```

3.3 Database

Each data sample has been saved in a MySQL database. Two table for the samples has been created, one for the MQTT data and one for the CoAP data:

```
CREATE TABLE mqtt (
  measureID int AUTO_INCREMENT,
  node_id INT NOT NULL,
  energy_consumption INT NOT NULL,
  timestamp varchar(255) NOT NULL,
  sound_alarm_status varchar(255) NOT NULL,
  PRIMARY KEY (measureID)
);
```

```
CREATE TABLE coap(
  measureID int AUTO_INCREMENT,
  node_id INT NOT NULL,
  temperature INT NOT NULL,
  timestamp varchar(255) NOT NULL,
  ac_temperature INT NOT NULL,
  PRIMARY KEY (measureID)
);
```

3.4 Data Visualization

The data can be visualized with two python programs, one for MQTT and one for CoAP, that calls a SELECT query every 5 second to retrieve all the data in the respective tables.

```
osboxes@osboxes:~/contiki-ng/examples/IoT_Project/collector$ python3 coap_show_data.py
+-------------+-----------+---------------+-------------+-----------------+
|   measureID |   node_id |   temperature |   timestamp |   ac_temperature |
+=============+===========+===============+=============+=================+
|           1 |         3 |            30 |          18 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           2 |         3 |            30 |          23 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           3 |         3 |            30 |          28 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           4 |         3 |            30 |          38 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           5 |         3 |            30 |          43 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           6 |         3 |            30 |          48 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           7 |         3 |            30 |          53 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
|           8 |         3 |            30 |          58 |              22 |
+-------------+-----------+---------------+-------------+-----------------+
```

```
^Cosboxes@osboxes:~/contiki-ng/examples/IoT_Project/collector$ python3 mqtt_show_data.p
+-------------+-----------+----------------------+-------------+----------------------+
|   measureID |   node_id |   energy_consumption |   timestamp | sound_alarm_status   |
+=============+===========+======================+=============+======================+
|           1 |         2 |                  918 |          35 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           2 |         2 |                 1005 |          40 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           3 |         2 |                 1231 |          45 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           4 |         2 |                 1644 |          50 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           5 |         2 |                 2023 |          55 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           6 |         2 |                 2372 |          60 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           7 |         2 |                 2716 |          65 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
|           8 |         2 |                 3024 |          70 | OFF                  |
+-------------+-----------+----------------------+-------------+----------------------+
```