UNIVERSITÀ DI PISA

Artificial Intelligence and Data Engineering

IOT SMART IRRIGATION SYSTEM

Gianmarco Petrelli Lorenzo Tonelli

Anno accademico 2021/2022

1. Introduction

This project work presents a demo of a SMART Irrigation System whose aim is to manage and regulate the water flow on a distributed network of automatic sprinklers, exploiting a network of temperature and humidity sensors.

Once several observations of temperature and humidity (for each specific area) have been captured and aggregated, the system is able to estimate a proper actuation plan in order to avoid waste of water.

The user who wants to monitor IOT system evolution will have a Web application available, which allows him to monitor the overall status of the irrigation system and the humidity and temperature conditions for each area or agricultural plot, guaranteeing effective maintenance of the entire system once deployed.

2. Requirement analysis

In order to build our system it will be necessary to implement an IOT architecture made of sensors, actuators and different server applications (deployed on virtualized environment) having the following characteristics and functionalities:

- Sensors dedicated to measuring soil moisture on a percentage scale (for each specified zone).
- Surrounding temperature sensors, positioned in a non-random way, near the humidity sensors.
- A binded actuator for each zone, whose aim is activating/deactivating sprinklers when needed.
- An IF-THEN business logic that defines the behavior of sprinkler actuators according to the temperature and humidity read by the respective sensors.
- A Collector component for the collection, aggregation and processing of data referring to different fields and coming from the devices that make up the LLN.
- A database for storing data coming from sensors in order to support the application and related telemetry system.

3. Actors/Systems

TEMPERATURE SENSOR

The temperature sensor measures the temperature of the surrounding environment on a °C scale at regular intervals of 60 seconds.

Each reading is then published at the broker node under the topic "TEMPERATURE".

WATER CONTENT SENSOR

The volumetric moisture content sensor measures the moisture value of the earthy substrate in percentage scale at regular intervals of 60 seconds.

The values measured generally are between 10% and 35%.

Each reading is then published at the broker node under the topic "HUMIDITY".

IRRIGATION SYSTEM ACTUATOR

The system is activated in critical water conditions when the environmental conditions of temperature and humidity require it.

The sprinklers activation will be emulated through the use of the LEDs integrated in the boards prepared for the realization of the demo.

The system also manages a manual activation mode of the area sprinklers through the use of the physical buttons set on the board.

COLLECTOR

The business logic core of the system: it's in charge of parsing, gathering and collecting data coming from sensors and actuators managing two different communication protocols (coap and mqtt).

This software component is developed as a Java server application and integrates the minimum application logic needed to trigger per-zone actuators according to specified logic.

4. IT Systems – Logical Architecture

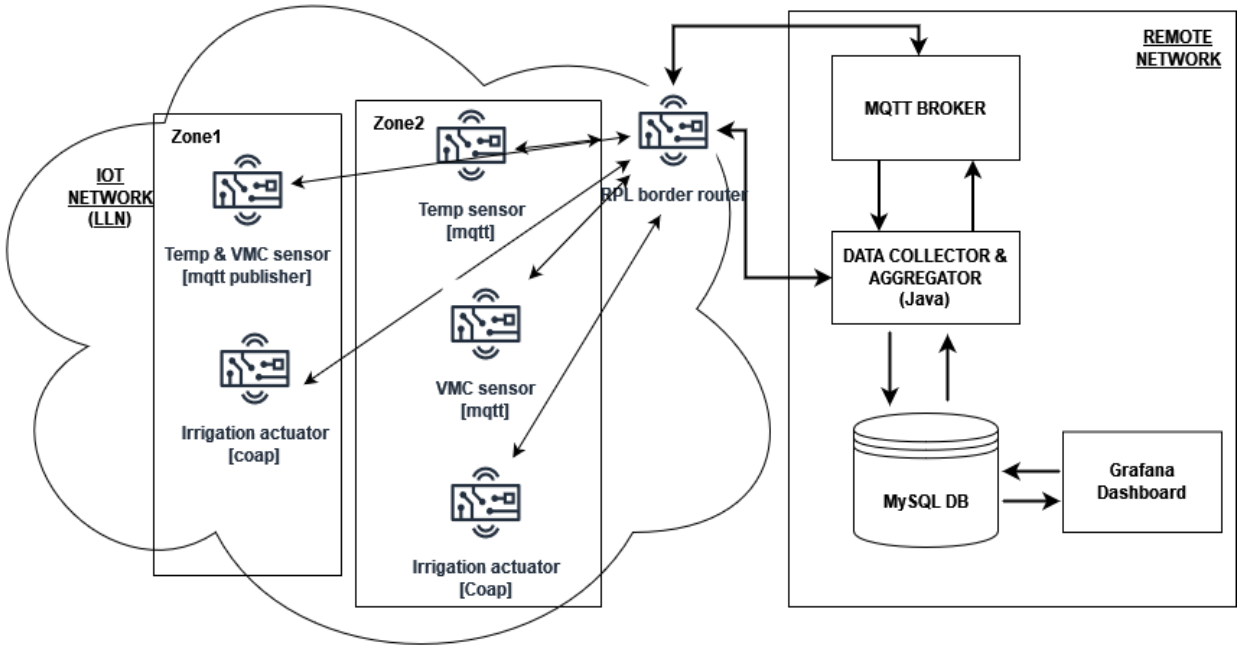
Below are shown main software components building up the logical architecture for IOT platform:

- Mqtt client for Temperature and WMC sensing [MQTT PUBLISHER].
- Mqtt broker to collect data streams exploiting a PUBLIC/SUBSCRIBE communication model.
- Coap actuator to trigger irrigation system.
- RPL border router.
- Java Collector to implement BL and data processing logic either for mqtt or coap protocol.
- MySQL server for time-series and application data storage.
- Grafana instance to provide user with a graphical interface implementing a dashboard to monitor the IOT system and environment.

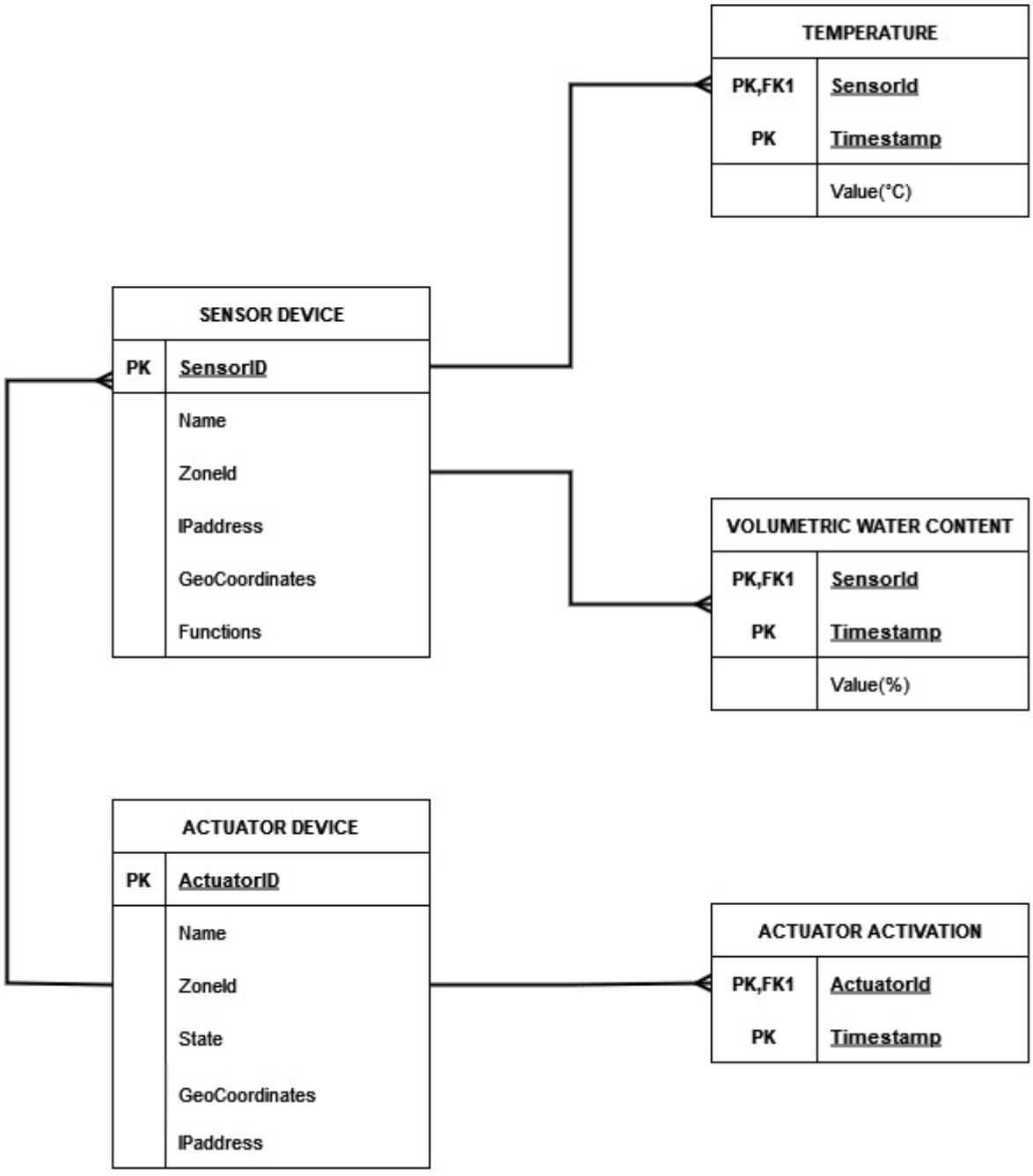
As a testing platform, before deploying the IOT application on real boards (CC2650 LaunchPad), we verified the correctness and effectivity of each software component exploiting Cooja simulation framework.

5. Physical Architecture

In this section we introduce the physical architecture diagram to show how different software components are deployed and coexist within the infrastructure.



6. Data Model


TEMPERATURE
PK,FK1
SensorId
PK
Timestamp
Value(°C)
SENSOR DEVICE
PK
SensorID
Name
ZoneId
IPaddress
GeoCoordinates
Functions
VOLUMETRIC WATER CONTENT
PK,FK1
SensorId
PK
Timestamp
Value(%)
ACTUATOR DEVICE
PK
ActuatorID
Name
ZoneId
State
GeoCoordinates
IPaddress
ACTUATOR ACTIVATION
PK,FK1
ActuatorId
PK
Timestamp

7. Development

General considerations

- The optimal level of relative humidity of the soil varies according to the type of cultivation; assuming agricultural land for generic crops (cereals, fruit trees,…) it lies between 21% and 40% (Volumetric Moisture Content).
- An irrigation session lasts about 15 minutes:
    - IF it is very hot it is better to increase session frequency and reduce duration.
    - ELSE a longer morning session is enough.
- The best time to irrigate the fields is early in the morning or late in the evening.

Environment Simulation

In order to simulate temperature and MWC sensor response to irrigation cycle, we defined some logic on sensor code in order to adapt environment variables properly every time a binded actuator triggers the activation of a session. To do this, we defined a topic "SIMULATION_SensorId": each related sensor is a SUBSCRIBER whereas the Collector application is the PUBLISHER.

The observable behavior consists in the increase of m.w.c. level and a soft mitigation of environmental temperature.

Instrumentations and technologies

- Java language for Collector and business logic component, following exploited libraries and tools:
    - PAHO: The Paho Java Client is an MQTT client library written in Java for developing applications that run on the JVM or other Java compatible platforms.
    - Californium: Californium is a powerful CoAP framework targeting back-end services communicating with smaller Internet of Things devices. Stronger Internet of Things devices may use Californium as well. It provides a convenient API for RESTful Web services that support all of CoAP's features.
    - JDBC for MySQL
    - GSON, Java.time, …

8. Grafana Web Interface

The exposed Web interface allows users to select different areas to observe thanks to Grafana variables that are always kept updated with the database current state.

The upper part of the window shows time-series values for temperature and humidity measurements.

Bottom-left part shows the actuator state for the selected area and to its right the corresponding ctivation session monitor.

Data are collected by the Java Collector and once available the dashboard is real-time updated.

9. DEMO

The real time span in which the demo takes place is 5 minutes.

Over the real time period, the simulated time corresponds to 24 hours.

To create the demo we implemented a TimeManager class in order to be able to map real time and simulated time at will.

In the demo we will show the various actuator operations, both automatic and manual, simulating a typical day.

Through the telemetry interface it will be possible to observe the evolution of environmental variables and the different activation cycles carried out in each area (for convenience, an example will be built in which only 2 agricultural areas are managed).