SmartOrchard

INTERNET OF THINGS

Project Report

Student

Arianna Gavioli

590761

A.Y. 2019/2020

The Scenario

The project considers a scenario where smart devices monitor and interact with a set of orchids. In particular, each orchid resides in a greenhouse and the considered scenario contemplates a set of specific sensors and actuators for each of them, as shown in table 1. The sensor nodes provide measurements taken within the orchid

Sensor Nodes	***Measurements***
Soil Monitoring	Soil Temperature (°C) Soil Humidity (%) Soil pH level Soil Salinity (ds/m)
Air Monitoring	Air Temperature (°C) Air Humidity (%)
Luminosity	Luminosity (lux)
Actuator Nodes	***Actions***
GreenHouse Heating	Controls the heating system of the GreenHouse
GreenHouse Roller Shutter	Controls the Roller Shutter of the GreenHouse
IIrrigator	Controls the irrigation system

Table 1: The Nodes

greenhouse, while actuator nodes provide the possibility to control some devices inside of the greenhouse. In particular, it is possible to activate the greenhouse heating to prevent frostings, open and close the greenhouse roller shutter, based on the temperature and on the lighting conditions, and finally, it is possible to activate the irrigating system of the orchid.

Once the sensor and actuator nodes are deployed in place, the general idea is to setup a cloud application able to retrieve the sensor information and to interact with the orchid by controlling the actions of the actuator nodes.

The Implementation

The implementation of the whole system has been carried out using CoAP communication protocol in order to exchange data between the Low Power and Lossy Network and the cloud application.

The LLN nodes were simulated through the cooja simulator, running with the contiki-ng operative system, while the cloud application was developed in java, exploiting the Californium library.

Please note that, to launch and execute the cooja simulation, the default contiki-ng rpl border router was used.

The Registration Phase

The cloud application offers an interface through which the motes can enroll themselves into the system: this is done by instantiating a CoAP server offering a resource `"/registration"`, on which the motes can invoke a post operation, specifying their features in the message payload. The latter is a JSON message of the form:

```
{"Functionality":mote_functionality,"ID":id,"IP":mote_ip}
```

The motes, when booting up, activate their resources and wait to be reachable from the border router and retrieve their own ip. Once the needed information has been obtained, they perform a post on the registration resource of the server, specifying their features in a JSON message and wait for a reply from the server. From this point on, they are enrolled and monitored by the system, ready to interact with it.

The Motes Resources

Once the motes register onto the cloud application, they keep on running and act as CoAP servers, offering their resources to the cloud. Here's the offered resources from each node.

Soil Monitor

- `/soil`: an observable resource which specifies soil measurements.
  - GET: returns a JSON message containing soil temperature, humidity, pH and salinity.
  - POST: modifies the resource fields based on the message payload, allowing to increase the humidity, increase or decrease the pH levels, increase the salinity or change the growth trend of the temperature. This possibility was introduced for simulation purposes.

Air Monitor

- `/air`: an observable resource which specifies air measurements.
  - GET: returns a JSON message containing air temperature and humidity.
  - POST: modifies the resource fields based on the message payload, allowing to change the temperature growth trend. This possibility was introduced for simulation purposes.

Luminosity Monitor

- `/luminosity`: an observable resource which specifies the luminosity measured.
  - GET: returns a JSON message containing the measured luminosity.
  - POST: modifies the resource fields based on the message payload, allowing to change the luminosity growth trend. This possibility was introduced for simulation purposes.

Irrigator

- `/irrigator`: a resource which allows to activate a physical irrigator. The status (on/off) of the irrigator is simulated using red and green mote leds.
  - GET: returns a JSON message containing the irrigator status (on/off).
  - POST: activates or deactivates the irrigator based on the message payload.

GreenHouse Heating

- `/gh_heating`: a resource which allows to control the greenhouse heating system. The status (on/off) of the heating system is simulated using red and green mote leds.
  - GET: returns a JSON message containing the heating system status (on/off).
  - POST: activates or deactivates the heating system based on the message payload.

GreenHouse Roller Shutter

- `/gh_roller_shutter`: a resource which allows to control the greenhouse roller shutter. The status (open/closed) of the shutter is simulated using red and green mote leds.
  - GET: returns a JSON message containing the shutter status (open/closed).
  - POST: opens or closes the heating system based on the message payload.

The Cloud Application & the User Interface

When the motes issue a post on the registration resource, the cloud application creates an instance of a specific client based on the specific node. In particular, it subscribes to the observable resources and it shows them using a GUI. Each new orchid will create a new Tab in the GUI. Image 1 shows what the GUI looks like.

As we can see, all of the sensed information is showed in the top part of the interface: each time a new measurement arrives, the interface field is updated accordingly. The

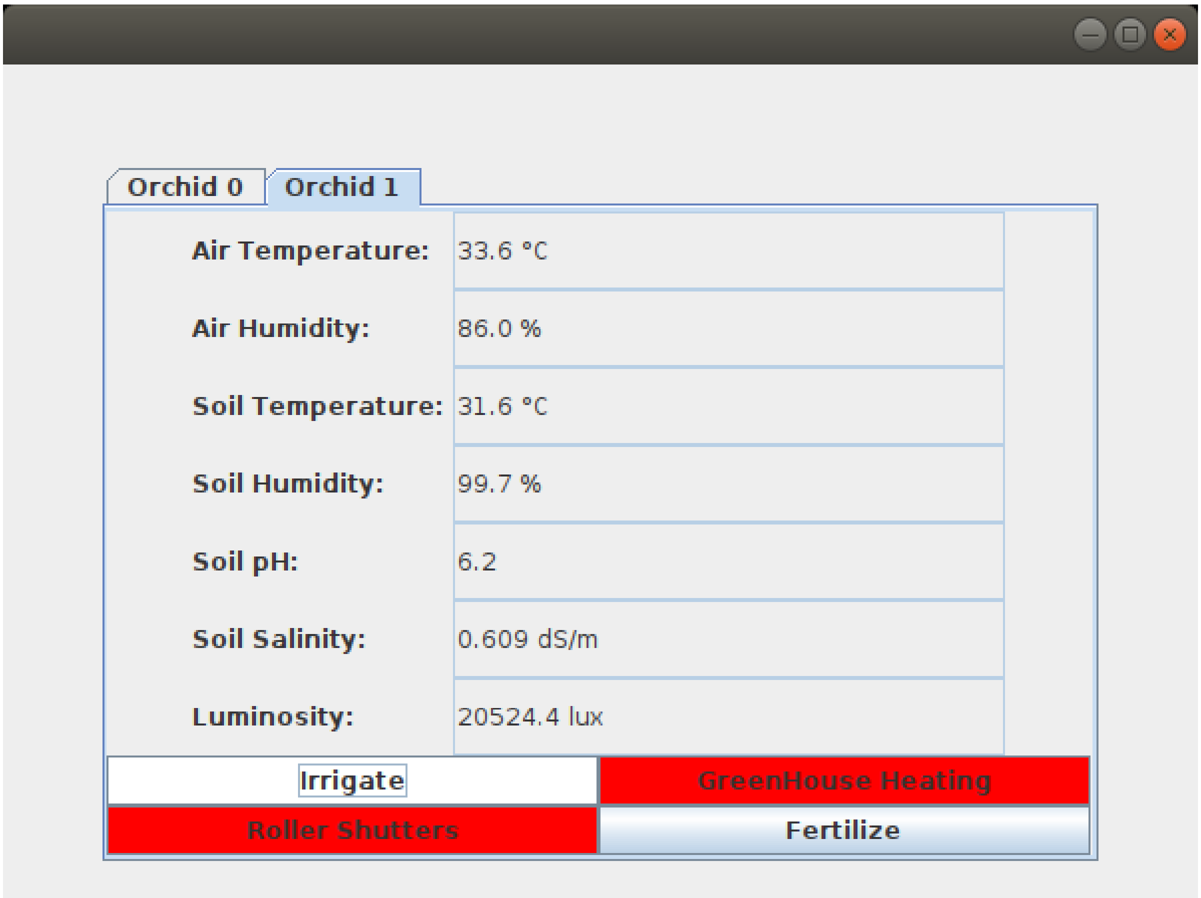

Figure 1: The Graphic User Interface

four buttons on the bottom give the user the possibility to interact with the orchid environment:

- Irrigate: it activates the irrigator mote of the corresponding orchid, while signaling to the soil resource to increase the soil humidity, for simulation purposes. If the irrigator is already active, the button turns blue and the subsequent pressing on it will deactivate the irrigator.
- GreenHouse Heating: it activates the greenhouse heating system, while signaling to the air temperature to start increasing. If the heating system is already active, the button turns green and the subsequent pressing on it will deactivate it.
- Roller Shutters: it opens the roller shutter of the greenhouse, while signaling the luminosity to turn down, for simulation purposes. If the roller shutter is already open, the button turns green and the subsequent pressing on it will close it.
- Fertilize: this button is not linked to any specific actuator node: it aims to simulate the human action of putting fertilizer onto the ground. When pressing the

button, the soil resource will be signaled in order to increase the soil salinity of 1 dS/m and to get the soil pH closer to 6.

Simulation Limits

The simulation was developed in order to emulate at its best a real situation, but of course it was not possible to replicate real actions. In particular, for the simulation sake, the observation periods of the sensed resources are very short, while in practice they would be significantly higher. The second important difference from the reality is that, in order to emulate the actuator's consequences on the sensed measurements, the sensors offered the possibility to change the measurements through POST operations on their resources. This would not be needed in a real environment, where the sensor would automatically acquire data according to the actions performed by actuators and to the external environment.

The measurements and their ranges were designed following the sources listed in the bibliography, in order to stick to the truth as much as possible.

Bibliography

- IOT LAB slides, University of Pisa, 2020
- FAO - Food and Agricolture Organization of the United Nations - http://www.fao.org/3/x5871e/x5871e04.htm
- Agriculture Victoria - http://agriculture.vic.gov.au/agriculture/farm-management/soil-and-water/salinity/testing-and-interpretation-of-salinity-and-ph
- National Optical Astronomy Observatory - Recommended Light Levels - https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf