A.A. 2020-2021

IN SUPREMÆ DIGNITATIS

1343

UNIVERSITÀ DI PISA

INTERNET OF THINGS

SMART FRUIT FRIDGE

GITHUB: https://github.com/lorebianchi98/SmartFruitFridge

PRESENTED BY:
BIANCHI LORENZO
BICCHIERINI IACOPO

Introduction

Smart Fruit Fridge is an automation application for the smart management of a fruit fridge. It could be useful in a logistics company or for greengrocers to manage the conservation and ripening of fruit.

Sensors & Actuators

Oxygen detector (sensor): measures the percentage of oxygen in the fridge.

Oxygen emitter(actuator): if active, increase the percentage of oxygen in the fridge, it is simulated through terminal print.

Temperature detector (sensor) : measures the temperature of the fridge.

Chiller (actuator) : if active, decrease the temperature in the fridge, it is simulated through terminal print.

Ethylene sensor (sensor) : measure the level of ethylene in a division of fruit.

Ripening notifier (actuator) : it shows using LEDs the status of the fruit based on the level of Ethylene. RED LED very ripe, YELLOW LED ripe, GREEN LED unripe.

Deployment

MQTT Network deployed using 3 sensors :

1 device deployed as **Border Router**

1 device deployed both as **Temperature detector** and **Chiller**

1 device deployed both as **Oxygen detector** and **emitter**

CoAP network is simulated with Cooja, we decided to deploy the sensor and the actuator related to the ethylene control, which are the **Ethylene sensor** and the **Ripening notifier**, plus the **Border Router**.

Collector & Control Logic

The Collector is responsible of receiving updates from the MQTT Broker and accepting connection of CoAP devices . In addition, we execute some control logic that allows:

- **Automatic management** of actuators (e.g. automatic management of temperature regulators and oxygen emitter)
- **Manual modifications** to the actuators through intuitive commands

We provide the user with a CLI with the following commands available:

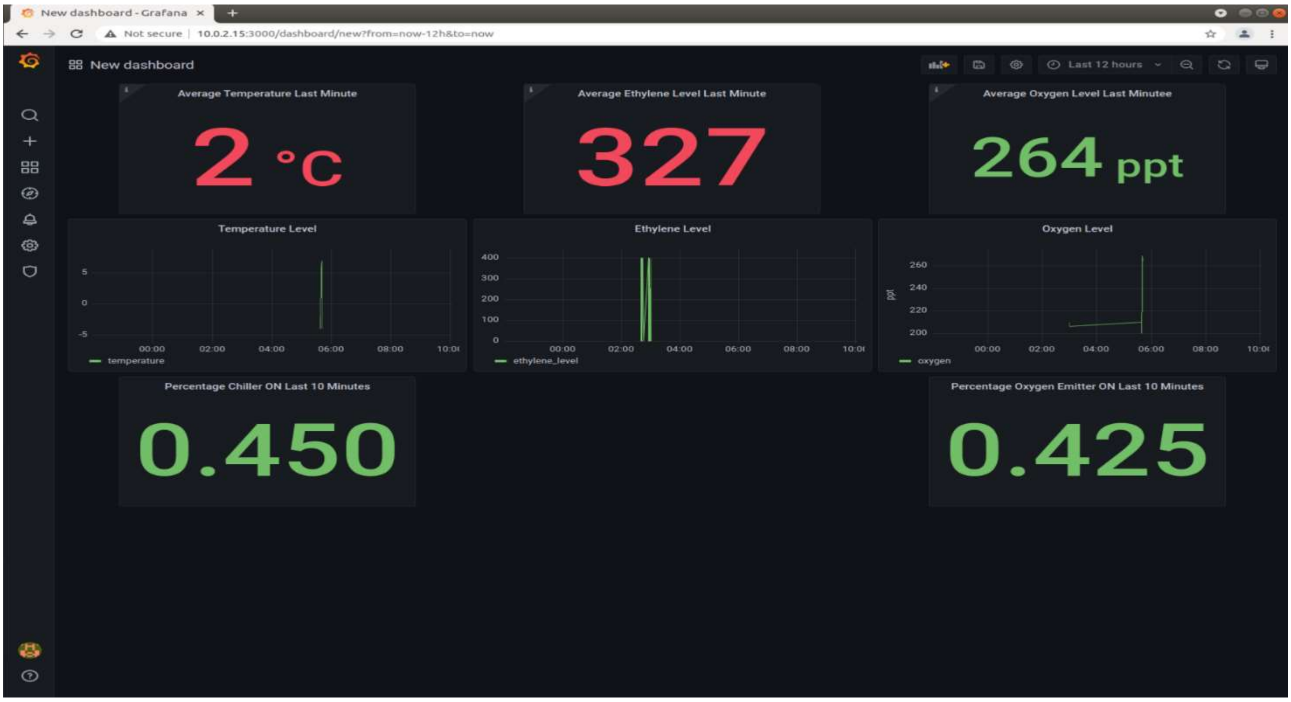
- !exit
- !commands
- !checkTemp
- !checkOxygen
- !triggerRipening
- !conservation
- !checkEthylene
- !getSensor
- !getNotifier

The 2 commands !triggerRipening and !conservation will send messages to the actuators to ensure that they keep the fruit stored through a very low temperature in the fridge and avoiding adding oxygen, or they will trigger the ripening through the emission of oxygen and the increase in temperature that will take place by switching off the cooler. Obviously there will be a closed control circuit in order not to bring these 2 values out of range (for example the temperature in the fridge is too high or the oxygen level is too low and so on).

Implementation Choices

We decided to implement the Collector and Control Logic using Java. The data encoded chosen is JSON since it is a lightweight data encoding language and our application is not a critical application and so a more structured language such as XML is not necessary. For the messages exchanged on the CoAP network we decided to use a simple text encoding, since they are single attribute messages (the only information exchanged is the level of ethylene) and there is no need to give a specific structure to the data.

Grafana



New dashboard - Grafana

Not secure | 10.0.2.15:3000/dashboard/new?from=now-12h&to=now

New dashboard

Last 12 hours

Average Temperature Last Minute: 2 °C

Average Ethylene Level Last Minute: 327

Average Oxygen Level Last Minutee: 264 ppt

Temperature Level

Ethylene Level

Oxygen Level

Percentage Chiller ON Last 10 Minutes: 0.450

Percentage Oxygen Emitter ON Last 10 Minutes: 0.425

The dashboard implemented using Grafana will show the 3 main values of our system in their evolution in time: **Temperature**, **Ethylene level** and **Oxygen level**.

Above each graphic it is shown the average value about the last minute.

In the case of the **Cooler** and the **Oxygen Emitter**, the percentage of time the actuator has been switched on in the last 10 minutes is also shown.