CONNECTED HIVE
Final report

BESNIER Benjamin, LEVESQUE Théo, WEILL William

Enseignant : Nicolas PALIX
Année universitaire 2017-2018

Table of contents

Project presentation

As Einstein said *“If the bee disappeared off the surface of the globe then man would only have four years of life left. No more bees, no more pollination, no more plants, no more animals, no more man.*

If the bee disappears from the surface of the earth, man would have no more than four years to live” [1]

So the bees are one of the most important animal species for us if it isn’t the most important one.

The main goal of this project is to have sensors placed in the hive to monitor its environment. Therefore, the project was initially divided in two parts, one who deals with the sensor (Thanks to the IESE) and the other one with the data recovery and visualization (The RICM part).

- First part: Embedded system
  This first subpart is composed of two aspects. Firstly, the IESE had to place different sensors inside the hive without bothering the bees. They managed themselves to put sensors for temperature, weight, humidity and pressure. As there are no bees for now, it’s easy to put/remove the sensors. Then, once the sensors were placed, the next step was to send the sensor’s information using LoRa protocol[2].

- Second part: Data processing and visualisation
  The data send by the LoRa card are gathered by a gateway connected to a Raspberry Pi. The unrelated packages are dropped and the sensor id is extracted before forwarding the message to the corresponding server. The server receives the message and decode all the sensor values. It stores all data to a time series database. It provides also an HTTP server to visualise all the data stored in real time, filtered by date and time.

[1] <https://quoteinvestigator.com/2013/08/27/einstein-bees/>

[2] <https://en.wikipedia.org/wiki/LoRa>

Description of tools used

1. LoRa gateway

The LoRa card located in the hive send LoRa packets retrieved by the gateway. This one is connected to a Raspberry Pi. The gateway software[3] transmit packets to the next application, Node-RED.

2. Node-RED

We have used two Node-RED server, this first is used when the packets forwarded by the LoRa gateway, they are retrieved and computed with the help of Node-RED[4]. On Node-RED, it's possible to create flow management scenario for connected objects. Here, our flow deals with the ID of the incoming packet, as we have used our own protocol. This ID is used to know to which MQTT topic sends the data. As the LoRa gateway collects all the LoRa packets surrounding, the flow had to remove these packets.

After that the packets are sent to the MQTT topic, the second Node-RED server is used. This one gets the packet from MQTT, uncode them to get the information from the sensors and send it to our InfluxDB server.

3. Mosquitto

Mosquito[5] is an MQTT broker. MQTT is a publish subscribe messaging protocol. The client can send messages to a topic, and all clients who have subscribed to this topic receive the message. In our case, the Raspberry Pi sends the packets retrieved by the gateway to topic that correspond to the sensors, ID and the server can store the sensor values in the corresponding database.

4. InfluxDB

The information collected is stored in a time series database such as InfluxDB. This kind of database store input data according to the date and time of arrival. It's designed to optimise metric/date graphic.

[3] <https://github.com/Lora-net/picoGW_hal> <https://github.com/Lora-net/picoGW_packet_forwarder>

[4] <https://nodered.org/>

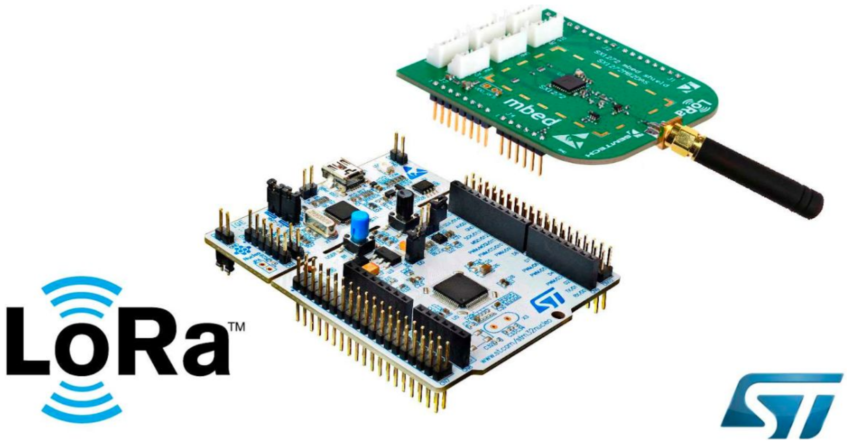
[5] <https://mosquitto.org/>

5. Grafana

The visualization of data is possible thanks to grafana. Grafana is an open platform allowing to do dashboard using InfluxDB metric. With a simple SQL[6] query, its show real time graphic.

6. Equipment

The sensors are connected to an ESP32. It is a system on a chip microcontrollers where we put a SX1272, a LoRa transmitter and receiver. The Gateway is connected to a Raspberry Pi, a small and cheap computer. The database and the web server are on a server provided by the IM2AG.



Functioning

As the future user of this connected hive will mostly be beekeepers, they are not accustomed with this kind of technology. As we also have 4 servers to start (Mosquitto, Node-RED, InfluxDB and Grafana) we have chosen to put all these server in Docker[7]. Thanks to docker, the user will only have to start the Docker using the following command:

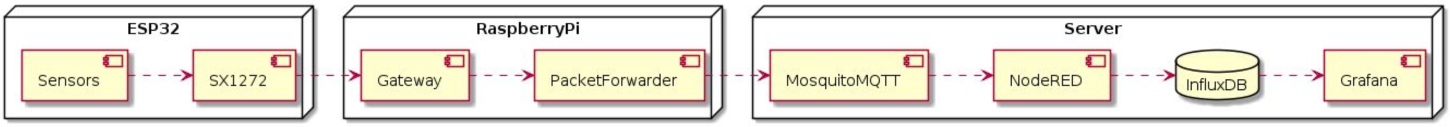
```
docker-compose -d up
```

[6] <https://en.wikipedia.org/wiki/SQL>

[7] <https://www.docker.com/>

In a terminal where both `docker-compose.yml` and `Dockerfile` are placed. Once the docker has started, the user only have to go to http://localhost:3000 to visualize the graphic on grafana.

Simultaneously, there is also a docker on the Raspberry Pi used to launch the LoRa gateway software and the first Node-RED server.



Future developments

As there is no bee in the hive for now, it could be really easy to add another example, for example, it’s possible to add a laser used to count the number of bees in the hive. Moreover, if the user have several hive, we can use Eclipse SmartHome[8] to detect where each hive are placed. Finally, as for now we use our own protocol we can change it to use LoRaWan[9].

Conclusion

To conclude, thanks to docker, our project is usable on every device as long as docker is installed on it. This project was very interesting and permit us to learn new technology (Node-RED, LoRa, docker). As our first project working with IoT it was a discovery for us and it we mostly enjoyed all the work we have done.

Our project is avaible on Gitlab : https://gricad-gitlab.univ-grenoble-alpes.fr/Projet-RICM4/17-18/10

[8] <https://www.eclipse.org/smarthome/>

[9] <http://jensd.be/755/network/lorawan-simply-explained>