Sensing and Actuation Networks and Systems [2022-2023]

Project (Part II) – Process and visualize IoT data

Introduction

This assignment aims to create a full IoT solution from data collection to processing and visualisation. Students will have to program MQTT publishers that are sensing environmental data, process the data gathered, store it in a database, and finally visualize it on a dashboard.

Objectives

Students successfully concluding this work should be able to:

- Manipulate real-time data collected by sensors connected to a Raspberry Pi
- Transfer and collect real-time data using MQTT
- Process data coming from sensors and save it in a database
- Deliver alarms to a surveillance console
- Display real-time data in a dashboard

Support Material

- Raspberry Pi 2 Model B with different sensors attached are available for remote access
- Python files to collect data from the sensors
- Files and examples from previous classes
- Python JSON: https://www.w3schools.com/python/python_json.asp

Network configuration

The network configuration to use is depicted in Figure 1. It is composed of **5** different parts:

1. **Data extraction**: consists of a group of different sensors, each connected to a specific Raspberry Pi device; the data gathered will be published to a MQTT broker.
2. **MQTT broker**: provides the connection between publishers and subscribers and is located in a Raspberry Pi; a Mosquitto MQTT broker will be used.
3. **Processing unit**: located in the students' computer, this unit processes data and saves it to an InfluxDB database.
4. **Real-time surveillance console**: client program that receives alerts whenever sensor values are outside the ones established as healthy.
5. **Storage**: all the data will be stored in an InfluxDB time series database (a InfluxDB Cloud free plan, located at https://www.influxdata.com/, will be used).
6. **Control dashboard**: the information will be visualized in a Grafana dashboard (a Grafana Cloud free plan, located at https://grafana.com, will be used).

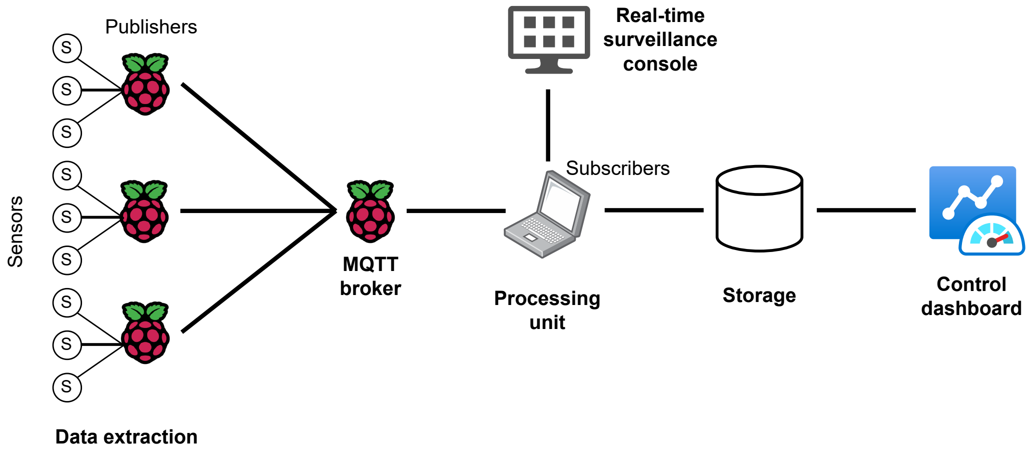

Figure 1. Network configuration

Each group of students (team) will have access to a specific Raspberry Pi in which a publisher will be created. The Raspberry Pi where the MQTT broker is located will only be accessible using MQTT, no other access is available to students.

To access the Raspberry Pi inside DEI premises, it is necessary to be connected to the "DEI" network. **Do not use the "eduroam" network!** To access the Raspberry Pi from outside DEI premises, you will need to connect to the department's VPN (Virtual Private Network). Please refer to the instructions provided by the helpdesk to connect to DEI's network using the VPN[1]. Both InfluxDB Cloud and Grafana Cloud are available with an Internet connection (registration in both sites will be required).

[1] https://helpdesk.dei.uc.pt

Available sensors

The sensors available for data collection are listed in Table 1. A python file to collect the associated data is also provided.

Table 1. Available sensors

Sensor	Data collected	Unit of data collected
Temperature	Temperature	C
Barometer	Temperature	C
	Pressure	Pa
Photoresistor	Luminosity	lux

Will not be used

To read the data from the sensors, you can use as reference the files provided in Part I of the project and from previous assignments.

Description of work

Each team will have to create a publisher and a subscriber, develop a data processing program, store the data in a time-series database, and finally display the results in a dashboard.

Publisher

Each team will have a specific **groupID** that will give access to a specific account in one specific Raspberry Pi. The access information for the Raspberry Pi is as described in Table 2.

Table 2. Access information for the Raspberry Pi

Group ID	Username	Password	IP address	DNS name
01-08	group**ID**	group**ID***.	10.6.1.8	srsa-pi-7.dei.uc.pt
09-16	group**ID**	group**ID***.	10.6.1.9	srsa-pi-8.dei.uc.pt
17-24	group**ID**	group**ID***.	10.6.1.10	srsa-pi-9.dei.uc.pt

Example: Group 01 will have the username ***group01*** and password ***group01*.*** .

To obtain a groupID, a previous registration via email, sent to the PL teacher, is required.

After obtaining a groupID, students must connect to the respective Raspberry Pi (according to Table 2) and change the password, to protect each team files. To change the password, use the command “`passwd`” in the Raspberry Pi.

Next, it is necessary to create a folder inside the user default path (`/home/<username>/project`).

Virtual environments **must always** be used during the development process.

Each team must define as many topics as needed for data exchange using MQTT. The name of the topics must follow this format: ***GroupID_topic***

For example, if group ***03*** wants to create a topic to share ***luminosity*** data, the topic name must be “***03_luminosity***”.

If data of the same nature is collected from different sensors, the topics must have different names. Example: ”***03_temperature_1***”, ”***03_ temperature_2***”.

MQTT broker

A Raspberry Pi with a Mosquitto MQTT broker will be available. To access this broker use IP 10.6.1.4 (srsa-pi-3.dei.uc.pt) and port 1883. This broker will hold all the students’ topics. Additionally, three topics named ***sensor_data_hum_1***, ***sensor_data_hum_2***, and ***sensor_data_energy***, will receive synthetic data, generated locally (students will have access to the data but are not responsible for its generation).

Note:

- Both ***sensor_data_hum_1*** and ***sensor_data_hum_2*** will deliver float measurements of humidity in the air.
- ***sensor_data_energy*** will report the energy consumption, both in peak hours and off-peak hours, since the beginning of the last accounted period (new periods are created from time to time without student intervention). The data is sent using JSON (JavaScript Object Notation).

  Example:

  ```
  energy = {
       "peak": 10.3,
       "off_peak": 2.5
  }
  ```

Processing unit

The processing unit will be located in each team computer. The processing unit is a server program that has the following functionalities:

- Subscribes to each team own topics, and also to the three additional *sensor_data* topics; data from other teams can also be collected.
- Reads a configuration file with healthy sensor intervals (`healthy_intervals.cfg`); the healthy values are only defined for the team own sensors and to the two additional *sensor_data*.
- Generate alarms when the values received from sensors, through the MQTT broker, are outside the healthy interval defined; these alarms are sent by UDP to the Real-time surveillance console.
- Generates an alarm cancellation whenever a sensor returns to its normal range.
- Calculates the average of all temperatures measured; the value should be in Celsius and Fahrenheit.
- The *sensor_data_energy* represents the energy consumption since the beginning of the last accounted period, measured in kWh, both for the peak hours and off-peak hours; the price for peak hours is 0.20 eur /kWh, and for off-peak hours is 0.10 eur/kWh.
- Saves data to an InfluxDB database:
  - Individual sensor values (no individual temperature values will be stored);
  - Temperature average;
  - All other processed values found necessary for the dashboard.

Real-time surveillance console

This console receives alarms from the Processing unit, using UDP sockets. The alarms are displayed in the console screen, showing info about the sensor, location (the Raspberry where it was collected), the date/hour when they were received, and the first time they were received (in case an alarm is repeated). To avoid continuously repeating the same alarms in screen, repeating alarms are only showed each 20 seconds (unless if they are cancelled).

```
04/05/2023 19:59:45 Temperature – Rasp 09 – Value: 38 (NEW ALARM)
04/05/2023 20:00:05 Temperature – Rasp 09 – Value: 38 (started 04/05/2023 19:59:45)
04/05/2023 20:00:25 Temperature – Rasp 09 – Value: 39 (started 04/05/2023 19:59:45)
04/05/2023 20:00:30 Temperature – Rasp 09 – Value: 35 (ALARM CANCELLED)
```

Storage

An InfluxDB database will be used to store data. This database must be created in InfluxDB Cloud (https://www.influxdata.com/). The registration of a free plan is required (registration has already been made in previous classes, students can use the same one). The database should hold all the necessary values to enable the following analysis of the data in the Control dashboard.

Control dashboard

Grafana will be used to create a control dashboard. This dashboard must be created in Grafana Cloud (https://grafana.com). The registration of a free plan is required (registration has already been made in previous classes, students can use the same one).

Each team must prepare a dashboard to display the data collected from the sensors and processed in the processing unit. The data analysis should include three separate sections:

Section 1: Summary dashboard

- Students should select the most important information to present in the dashboard and use the best Grafana charts and graphs.

Section 2: Detailed dashboard

- Last values from sensors;
- Healthy intervals;
- Values history;
- Moving averages.

Section 3:

- This section will focus the energy consumption that was originally reported by ***sensor_data_energy***. The variation along time, and the cost associated with it, must be visualized.
- Data visualized will help the management understand when more energy is consumed.

Project delivery

This section describes the delivery conditions and submission policy for the project.

Delivery conditions

The project should be done in groups of two (2) students. While the project is a group work, each student will have its own grade, corresponding to the individual performance in the final defence. Students must use MQTT for data transmission between the Raspberry Pi and their dashboard, and UDP sockets for the Real-time surveillance console. Python programming language must be used.

Project submission

Each group of students must create a zip file containing all the necessary files for their project to work. Include all source and configuration files needed. Do not include any files not needed for the execution of the program. Also add a report (maximum 1-page A4 in pdf format) that includes information about the data sent to the MQTT broker, saved in the database, and the analysis made to the data. The final zip file must be submitted to Inforestudante (only one zip file per group). **Submissions via email will not be accepted!**

Submission date: 29/May/2023

Defence date: 1 and 2/Jun/2023

Evaluation

This practical assignment is worth five (5) points of the final grade. The evaluation will consist of the categories listed in Table 3. The final grade will be weighted according to the individual defence of the project Not being present for the defence will result in a grade of 0 for that student. Students may be asked, during defence, to alter some of the graphics shown in the dashboard.

Table 3. Evaluation criteria

Module	Value
Data extraction	20%
Processing unit	30%
Real-time surveillance console	20%
Storage	10%
Control dashboard	20%
Total	100%

Plagiarism

There will be no tolerance for plagiarism or any other type of fraud. Attempts to do so will be graded ZERO and will imply the failure of the course. Depending on the severity, the situation may lead to disciplinary proceedings. Code will be subject to inspection and comparison between all students.