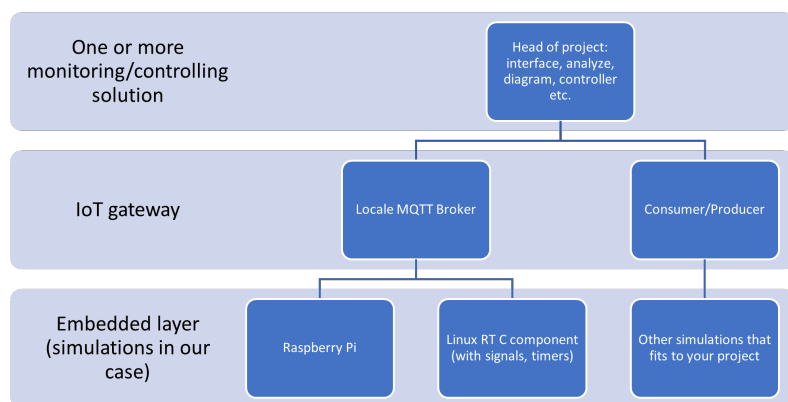


Practice requirements

A final project

You will get a grade for a big homework. You need to do it in **groups** (2-3 people). The main goal is to create a **complex real-time system**, which includes the technologies and concept that we learnt during the semester. The system's components are those application, that we created together.

It is really important, that your team needs to **design** the scope of the project, and its **architecture** as well. The following diagram shows the general structure, that you need to implement.



You can **extend the bottom layer** with any simulators which is not definitely the learnt Linux C RT application. For example, if you would like to do something in Python that simulates a device, feel free to do that. The keywords are careful, accurate, nice. **If you miss a component from the following list, you cannot get more than the half of the points:**

1. Linux C RT component
2. Gathering something from the real world
3. Local MQTT broker
4. Cloud service and database (Firebase)
5. Monitoring/Controlling solution

Grades and points

Criteria	Description	Points
The system is dynamic	The system can be easily extended with a new feature or device, or one can be removed without affecting the system.	5

Monitoring
Solution

Real-Time **data visualization** feature.
Real-Time **device control**. Easy to use, **user friendly, nice** 10

The system is able to gather information from the **environment**, and not only one type sensor data. Those values can be **monitored** as well. The data gathering contains some **error handling**.

Data gathering
Solution

Some of the components will be just simulators, so not all the components will use “real life data”. On the other hand, you cannot get the maximum points, if all of the components are simulations (i.e.: you need to use the Raspberry Pi, or your own external device). 10

At least one component of the system is:

- **subscribed** to a topic
- **publishes** something

The *MQTT topics* are well defined:

Real-time
communication

- The structure of them is a good **representation** of the system 15
- The system can be understood by studying the **topics**

No useless communication, too many connections or **bandwidth** wasting. The database is well defined.

Simulation of a real problem 5

Demonstration

The **demonstration** shows all the available features well. It is clear, quick, understandable and illustrative. It is clear, who did what (in the team) and both team members were active 5

Real-Time C component

Do not forget that you can not create the real-time C component on a Raspberry Pi, because the Raspbian is not a real-time operating system.

You have 2 options for the embedded (real-time) layer:

1. Use the ELTE server
 1. Note: You can not install and manage a mosquito broker on the server (because you have no permission to do so)
2. Use your local computer
 1. Note: If you have Windows, install WSL + Ubuntu (you can install mosquito as well in WSL)