Scalable IoT Systems based on OpenRemote
Project plan

Vladislav Serafimov

September 2, 2025

Contents

CHAPTER 1

Introduction

This chapter introduces the foundational concepts behind the assignment.

1.1 Background

The assignment was created by the Ambient Intelligence (AmI) research group in order to expand on the current knowledge in the Internet of Things (IoT) research area. AmI has worked on a number of IoT projects related to sports, safety and industry. Their aim is to make environments more interconnected, by monitoring them, processing the data and feeding it to the user.

The student thesis project is a part of NOWATT, which is a collaboration between academia and industry. The NOWATT project involves both AmI and the company behind the OpenRemote [1] platform. Because the multi-organization team would like to use OpenRemote and integrate it into their future work.

1.2 Purpose of the assignment

The assignment is designed with the intention of expanding on what AmI has already been researching. Its main purpose is to develop scalable IoT systems using OpenRemote and to do so with clear visualization.

There are three main parts that make up the assignment. The first one is deploying an OpenRemote instance and simulating IoT devices that communicate with it. An MQTT [2] setup needs to be realized, over which the data needs to be transferred from the mock devices to OpenRemote. The second part of the assignment is edge computing optimizations. They are another crucial part of the assignment, as the processing that is not done on the cloud needs to be as fast and resource-efficient as possible. Lastly, in order to test and verify the proper working of the devices, the sensor data needs to be extracted and validated by generating graphical representations of it. The data gathered throughout the project will also need to be analyzed using statistical models. These two tasks will be a part of the testing activity (see Chapter3.3).

If there is time left after finalizing the core of the project, data prediction using machine learning can also be implemented. Firmware for IoT devices is another feature that can also be developed, if there is enough time.

1.3 Assignment specifications

Figure 1.1 shows a diagram of all the functionalities offered by OpenRemote. In brief, OpenRemote is a platform for managing IoT devices. The most important features for this project are the MQTT agents (functioning as brokers/clients) and the built-in data visualization options.

The languages used for this project will be Python and C/C++. The extensive and varied catalog of Python packages makes the language better for tasks where simplicity is valued over

[1] OpenRemote is an open-source device management platform; for more information see this link

[2] MQTT stands for Message Queuing Telemetry Transport **mqtt˙wiki**. It is a communication protocol, mainly used for purposes relating to IoT.

Figure 1.1: OpenRemote application diagram **openremote˙intro**

speed. Because of this, during the project, Pyhton will be used for modeling and visualization of the data. The MQTT communication can be done using both, which means they need to be evaluated during the project to see which one suits it better.

Data visualization and modeling will both be done, at least partially, in Python. ‘matplotlib‘ is one of the most widely-used packages for data visualization and because of this it was chosen for this project as well. Because OpenRemote has built in features for data plotting, it will also be used for some visuals. Data modeling, using statistical models, will be done in order to make sure the final product is configured in the optimal way. The optimal configuration will balance performance and service availability. Python's relevant features are the reason it was chosen for this task as well.

The extra features will use the same languages. C/C++ is more suited to low-level applications, so it will be used for the firmware. Python has a plethora of features and packages that make it good specifically for machine learning, so data prediction will be done in Python.

1.4 Methodological approach

The waterfall model (see Figure 1.2) was selected for this project. It is a straightforward model that is suited for assignments where the requirements do not undergo any significant changes. This is the case in this project, because the output needs to align with the goals of the overarching project AmI is working on. Another relevant strength of this model is that adhering to the methodology does not require much effort to implement once the groundwork is laid. Other models, like Scrum, require a lot of time and energy to be implemented properly, but in the waterfall model every stage follows up on what the previous set up. What is more, the chosen model is excellent for documentation. The straightforward approach to the project means there will be no retroactive editing of existing documentation. This leads to a situation where the documentation product of each phase will lead into the next.

Figure 1.2: Project phases using the waterfall model

CHAPTER 2

Products and project objectives

This chapter describes all the required products and based on them sets up the objectives for the project.

2.1 Products

There are two types of products that need to be delivered: professional products and learning products.

The professional products consist of several deliverables, which need to be provided to AmI after completion of the project. The main deliverable of the project is the firmware code. Just as important are the data extraction and visualization scripts. The edge computing optimizations should also be documented and reported as a part of the final product. The final report is also an important product, as future work will require some foundation to base further research on.

The learning products are there to ensure the student (see Table 6.1) has completed the learning goals for the project. The final report is also an important part of the learning products, as it demonstrates knowledge and the ability to communicate that knowledge. Another important deliverable is the learning report, which is an introspection about the project and its highlights. The last deliverable is the final presentation PowerPoint file, which is not as important as the oral presentation. However, it is still a demonstration of the acquired knowledge and skills during the project.

2.2 Objectives

As the chief goal of the project is to expand AmI's pool of knowledge, the project objectives need to align with that goal. It is also important to note that the project is done as a graduation assignment, which means there are learning objectives to be completed in parallel with the professional objectives.

The SMART[1] method for setting goals cab be applied to this project. The goals can be made even more specific by using the mandatory deliverables for the project. SMART will simplify the bookkeeping of the whole project.

- Goal #1: Complete the learning objectives of the project.
- Goal #2: Complete the professional objectives of the project.

2.2.1 Learning objectives

1. Project plan: outline of the project
2. Technical report: technical documentation of the project
3. Learning report: reflection on the project as a learning activity

[1] SMART in this case stands for Specific, Measurable, Achievable, Relevant, Time-bound

4. Final presentation (incl. presenting): communication of the highlights of the project

SMART

- Specific: The deliverables associated with the learning objectives all have to be completed by the student and submitted for review to Saxion. The only exception is the final presentation, which includes both the digital presentation and the in-person presenting done by the student. The learning objectives are set in place so that the student can demonstrate sufficient knowledge and skills in order to obtain a diploma.
- Measurable: Every deliverable has a corresponding grading rubric, which quantifies the proficiency of the student.
- Achievable: The learning objectives build on top of the knowledge that the student has acquired during their study at Saxion. In order to navigate any setbacks that concern the learning objectives, the student can request feedback from the Saxion coach. All other work can be completed by the student with the use of the internet and all publicly available tools and resources.
- Relevant: Goal #1 is aligned with the degree of the student. Learning objectives are set in place to showcase to what level the student has mastered the knowledge and competences acquired during the study.
- Time-bound: Every submission has its own due date, consult Chapter 7 for more information.

2.2.2 Professional objectives

1. Code: main deliverable; includes everything from firmware to scripts
2. Visualizations: graphs corroborating the findings of the project
3. Documentation: a document covering the findings of the project and how to work with the code

SMART

- Specific: The professional objectives were discussed with the company coach before the technical work on the project was started. The main goal of the project is to test the scalability of OpenRemote in order to prove whether or not it can be used in future (industrial) IoT projects carried out by AmI. The auxiliary goals, set in place to ensure the findings of the project are as truthful and useful as possible, are discussed in Chapters 1.3 and 4
- Measurable: The project should result in clear visualizations and numbers. These outcomes should show how the speed and stability of OpenRemote scale with the number of devices. The analysis that can be performed on this data should result in an answer to the question "Is OpenRemote scalable enough?".
- Achievable: The student has the skillset to conduct the technical part of the project successfully. The physical tools that the project requires will all be provided by AmI. AmI also provides technical support in the form of the company coach.
- Relevant: The project is a part of a bigger project within AmI called NOWATT. The results of the professional objectives will directly utilized by the NOWATT team.
- Time-bound: The timeline for the professional objectives needs to align with that for the learning objectives. This is so, because the project is done as a graduation assignment and the graduation deadlines are mandated by Saxion.

CHAPTER 3

Project activities

For the sake of simplicity the project can be broken down into 3 phases as seen in Figure 1.2. Those phases are the start phase, realization phase and finalization/end phase. This simplification helps with the scheduling of the tasks. There is a task, which needs to be done throughout all phases and that task is documentation. Due to the use of the waterfall model (see Chapter 1.4), every phase of the documentation activity seamlessly flows into the next. Because of this, it is indicated as separate in Figure 1.2.

3.1 Start phase

The start phase includes the first 2 blocks of the waterfall model (see Figure 1.4). The familiarization involves getting acquainted with AmI's technical requirements and what technology they expect to be involved in the project. What is even more crucial in this period is getting familiar with the assignment and making sure the plan meets AmI's expectations. After concluding the process of familiarization, the research phase can start. It involves exploration of material that can help with building the solution and possibly looking into similar problems and their solutions.

There are also several documentation activities during this phase. These activities include scheduling, risk analysis and the project plan.

3.2 Realization phase

This phase is the longest and most important of the three. The assignment should be mostly completed during this phase. The first part of realization involves implementing the different parts of the final product (firmware, visualizations of mock data, etc.). The second part is about combining the separate implementations into a single, mostly-finished, product.

Alongside the formerly mentioned tasks, the technical report needs to be worked on as well. The learning report should also be started before the end of the realization phase. The technical report will be prioritized over the learning report, as it requires more time and effort. The code will also require some documentation, in order to aid future work that extends the functionality of the products of this project.

3.3 End phase

While some amount of testing will be done during realization, holistic tests can only be performed once the integration activity has concluded. After testing and fixing the product, the finalization activity can begin. During this activity, all the minor details that need attention will be addressed (e.g. code formatting).

During the end phase, and especially during the finalization, a lot of documentation will need to be done. Two separate presentations, one for AmI and one for Saxion, will need to be made and presented. The technical and learning report, as well as all the code documentation, will need to be finalized.

CHAPTER 4

Project boundaries

The project boundaries were initially brainstormed based on the contents of Chapters 1.2 and 1.3. After that, they were discussed with the client to ensure both parties were aware of the other's expectations.

Must have

- Sensor data extraction
- OpenRemote integration

Should have

- Visualizations using sensor data
- MQTT configuration

Could have

- Physical device implementation
- Data prediction using machine learning
- Firmware implementation for emulated devices

Will not have

- Cross-platform compatibility
- Several implementations for different hardware platforms
- Cloud-based processing

CHAPTER 5

Quality assurance

The quality of the project and the resulting products is of the utmost importance.

To guarantee the quality of the technical product, a lot of testing needs to be done. As discussed in Chapter 3.3, tests will be conducted during implementation and integration, but the main testing activity will be conducted during the end phase of the project. This method of working was chosen in order to maximize the chance of spotting issues with the products.

In order for the student to remains productive, several safeguards need to be put in place. Frequent communication between the student and the company coach (as discussed in Chapter 6.2) is one of the most vital safeguards. It guarantees both parties are well-informed and might help resolve issues if any arise during the project. Another important factor for the student's performance is the work environment. Because of the preferences of both the student and the client, most of the work hours will be done on location in one of AmI's rooms. Most of the project work can also be done remotely, if the need arises.

CHAPTER 6

Project organization

The project organization chapter discusses the different parties involved in this project and how they are expected to interact.

6.1 Organization

Name	Work email	Role
Vladislav Serafimov	v.serafimov@saxion.nl	Student
Yanin Kasemsinsup	y.kasemsinsup@saxion.nl	Saxion coach
Eyuel Ayele	e.d.ayele@saxion.nl	Company coach

Table 6.1: People involved in the project

The student is the person doing the assignment. They are responsible for the deliverables. Furthermore, they should be the initiator and proactively seek out the Saxion and company coach, if necessary. The Saxion coach is in charge of grading and is responsible for supporting the learning needs of the student. The company coach has similar tasks, with a few key differences. Both coaches participate in the grading, but their responsibilities vary depending on the deliverable. The exact differences are not relevant and will not be discussed in this report. The most important difference between the two coaches is that while the Saxion coach provides learning support, the company coach is responsible for the technical support of the student. The company coach also serves as a supervisor and as such makes sure the student is working on the project in a manner similar to the agreements made in the contract and the project plan. There will also be meetings with the Saxion coach to ensure the learning goals are met and that the products required by Saxion will be delivered in a timely manner.

6.2 Communication

The communication will mainly be done online. This enables easier sharing of files and data. However, physical meetings between the student and the company coach are crucial for the success of the project. In order to ensure the quality and consistency of the student's work, these meetings will be held often in the beginning of the project. As time goes on, less of these meetings will be needed to maintain the same level of quality.

CHAPTER 7

Planning

This chapter discusses the scheduling of the project. The schedule was created using a Gantt chart (see Figure 7.1), which contributes to the established structure of the project.

The deliverables are also a part of the planning. The project plan should be done a week before the end of the start phase. Using in, the functional and technical design chapters of the technical report can be completed. The report can then be submitted before the start of the finalization activity, aong with the code, visualizations and all other technical deliverables. The final presentation needs to be delivered within the first week of the the finalization activity. To ensure delays do not cause problems with meeting deadlines, all the dates have a buffer time of about a week.

Figure 7.1: Gantt chart

CHAPTER 8

Risk analysis

This chapter discuses the most common risks that might occur during a project like this and ways to address them. For a more detailed risk analysis, see the attached MS Excel file.

The most critical issues that need to be prevented are those relating to hardware. This is because they have the biggest impact on the project, as seen in Table 8.1. In order to lessen the impact of a hardware issues and supply shortages, it was decided that the software should work on a simulation before being tested on a real piece of hardware. Simulating on different levels of abstraction can help simplify the implementation of the MQTT connection and later when a working connection has been established, a more realistic and intricate hardware simulation can be done. This way, if there is a supply shortage and there are no alternative distributors, the code can still be verified more accurately. In case of hardware issues, having simulations and an expected result will mean these issues can be diagnosed early on. This has the potential of saving a lot of time later in the project.

Risk	Description	Frequency/ Probability	Severity	Importance	Risk mitigation
Software issues	Bugs might get through testing	Low	Medium	High	See Chapter 5
Hardware issues	Hardware might malfunction	Low	High	Medium	Results of the tests with the physical devices need to be compared with the simulations
Time management	Unexpected events might cause delays	Low	Low	Medium	Every event in the schedule has a buffer in order to mitigate the impact of delays
Hardware availability	Required hardware components might not be available	Low	High	Low	Using alternative suppliers and/or components makes the likelihood of this being a serious hurdle low
Deployment issues	The OpenRemote instance and the Python code might interact in an unexpected way on the server	Low	Medium	Low	Docker needs to be used along with good networking practices
Unrealistic simulations	The real hardware might behave differently from the simulations	Medium	🟩	Medium	Different simulations with different levels of abstraction can solve this issue

Table 8.1: Risk analysis table