|  |
| --- |
| SOLENT UNIVERSITY |
| Project Initiation Document |
| Transmitter Power Measurement |
| **Joshua Alsop-Barrell**  **Matthew Brook**  **Klea Cengu**  **Matthew Dear**  **Kieron Gillingham**  **Mark Hartop** |
| **April 2021** |
| **Tutor: Craig Gallen**  **Project Sponsor: Arqiva** |

Contents

[1. Introduction 1](#_Toc69289410)

[1.1. Background 1](#_Toc69289411)

[2. Objectives and Benefits 2](#_Toc69289412)

[3. Requirements and Scope 3](#_Toc69289413)

[3.1. Functional Requirements 3](#_Toc69289414)

[3.2. Non-Functional Requirements 3](#_Toc69289415)

[3.3. Scope 3](#_Toc69289416)

[3.4. Deliverables 4](#_Toc69289417)

[3.5. Stakeholders 4](#_Toc69289418)

[4. Team 5](#_Toc69289419)

[4.1. Management 5](#_Toc69289420)

[5. Plan 6](#_Toc69289421)

[5.1. Approach 6](#_Toc69289422)

[5.2. Milestones 6](#_Toc69289423)

[5.3. Assumptions 6](#_Toc69289424)

[5.4. Constraints 7](#_Toc69289425)

[5.5. Risk Assessment 7](#_Toc69289426)

[6. Implementation 8](#_Toc69289427)

[6.1. Proposed Solution 8](#_Toc69289428)

[6.2. Chosen Technologies 8](#_Toc69289429)

[6.2.1. Back End 9](#_Toc69289430)

[6.2.2. Communication 9](#_Toc69289431)

[6.2.3. Front End 10](#_Toc69289432)

[6.2.4. Infrastructure 10](#_Toc69289433)

[6.3. Development of the System 11](#_Toc69289434)

[7. Evaluation 12](#_Toc69289435)

[7.1. Evaluation of System 12](#_Toc69289436)

[7.2. Further Work 12](#_Toc69289437)

[7.3. What have we learnt? 13](#_Toc69289438)

[8. Conclusion 14](#_Toc69289439)

[9. References 15](#_Toc69289440)

[10. Appendices A](#_Toc69289441)

[A. Key Contributions A](#_Toc69289442)

[B. Risk Assessment C](#_Toc69289443)

# Introduction

This report details the transmitter temperature measurement project which will consist of presenting the problem and illustrating the requirements and the scope of the project. This will also include specifics on planning, implementing, testing the system. This allows the team to accurately evaluate the intended software to conclude the report.

## Background

Arqiva operate the terrestrial broadcast TV and radio network of transmitters in the UK. These transmitters are controlled and monitored from a central location in Yorkshire. High power VHF transmitters are tested by connecting them to a 'test load' instead of an aerial. The transmitter functions at full power, but the radio frequency energy is sunk in a test load instead of being transmitted. The power output of the transmitters needs to be monitored, and the power is measured by applying it to a test load to adhere to the regulations. Currently the temperature of the test load is calculated using a thermostat locally, more specially using glass thermometers. The power is calculated by reading the difference in the temperature before and after the test load.

# Objectives and Benefits

We will create a system that will monitor temperatures of transmitters digitally and store that data on server/cloud to be accessed remotely. This will Initiate a process which will modify measuring changes in the power measurement at high power VHF sites. The indicted will be accomplished on the Test Loads, using two-input digital thermometers, thermo-pockets and suitable temperature probes. This will reduce the need for engineers to travel to site as well as improve the accuracy of readings compared to analogue. It will provide an accessible history of readings couple with automated alerts. As a result, it will generally be cheaper to replace/repair current system.

# Requirements and Scope

As a team we discussed the requirements grouping them as functional and non-functional allowing them to be organised, based on priority, for inclusion in the project sprints. The requirements were then used to aid the selection of in scope and out of scope tasks.

## Functional Requirements

1. Gather temperature readings from digital sensors.
2. Gather flow rate reading from the flow rate digital sensors.
3. Power should be calculated from digital sensor data.
4. Visualise the sensor readings for each transmitter.
5. On-site hardware should be able to be individually calibrated.
6. On-site hardware should be secure.
7. On-site hardware should transmit sensor readings/calculations securely.
8. A remote user should be able to view stored sensor readings for each transmitter.
9. All online systems should be secured behind HTTPS.
10. All online systems should be secured behind login credentials.
11. Temperature sensors should be able to withstand temperatures of up to 100˚C.
12. Temperature sensors should have a differential accuracy of 0.05˚C.
13. Temperature sensors should be tested and proven to be immune from VHF interference.
14. Hardware should cost no-more than £200 per test load.

## Non-Functional Requirements

1. Display historical sensor data for each transmitter.
2. Temperature sensors should be able to be simulated for testing purposes.
3. There should be a test harness.
4. The system should detect and raise alerts for anomalies in power/temperature readings, including response timeouts.

## Scope

The features included in the scope can be split into three sections software for on-site hardware, communications and front-end. The software for on-site hardware will include sensor readings and contain transmission of messages, which feeds into the next feature of communication to receiving messages and data storage. The front-end of the program will be built within the user interface which will give the users the platform to interact with the software.

Given the current restrictions by the government it was not possible to visit the site to implement or select the required hardware for the project and therefore has not been included within the scope.

|  |  |
| --- | --- |
| **IN SCOPE** | **OUT OF SCOPE** |
| Software for Sensor Readings | Installing hardware |
| Messaging Architecture | Hardware selection |
| System security | Hardware budget |
| Data storage |  |
| User interface for monitoring sensor readings |  |
| Alerts |  |

## Deliverables

An overview of the aspects being delivered by the team includes the deployable docker container and their configuration files. The project will also include documentation such as a user guide and a developer’s guide. The user guide will be intended for engineers on site to assemble and configure whereas the developers guide will cover the technical aspect of deployment. A test plan has also been provided which was followed during evaluation to confirm the system requirements were met.

The link provided below will provide you access to the documentation of our Wiki page on GitHub:

Wiki: <https://github.com/mattdear/temperature_transmitter/wiki>

## Stakeholders

|  |  |
| --- | --- |
| Name | Role |
| Peter Katic | Sponsor |
| Craig Gallen | Tutor |
| Temperature Monitoring Attendant | The end user who will monitor the incoming messages through the provided visualisation software. |
| On-Site Engineer | An engineer who may be required to recalibrate the hardware on-site. |
| System Administrator | An administrator who will need unrestricted access to any component of the system including data storage. |
| Governing Body | An organisation that requires the system to adhere to safety guidelines and policies. |

# Team

## Management

The team consists of six people, with each member assigned a role at the beginning of the project. These roles were reconsidered during development, originally including a separate product owner and scrum master which were later combined into a single Project Manager when it was noticed that more active developers were required. The Testing Lead role was also reassigned to Joshua from Matthew during development as the Infrastructure development required more focus.

Diagram

Description automatically generated

The team was able to communicate through many streams including: GitHub, Discord and Microsoft Teams. We have used group meetings, breakout rooms, screen sharing, working from combined central locations of files with version control. The team had two weekly group meetings on Tuesdays and Thursdays, in addition the team frequently met with each other outside of scheduled meetings. Our team also had meetings with our tutor once a week and created presentation for the sponsor at the end of each sprint to provide updates on the project.

# Plan

## Approach

During the project we used an Agile Scrum methodology which is a flexible model consisting of sprints and selecting priority tasks at the start of each sprint. The team received feedback from stakeholders after each sprint. The difference that we took from the methodology is we had outlined and set the requirements from the start of the project whereas the methodology usually does it throughout the project depending on the feedback.

To evaluate the overall success of the project, the test plan has been carried out to confirm which requirements have been achieved in relation to the scope. The results of testing have highlighted what areas will require further work.

We evaluated the project management through milestones in each sprint. GitHub issues were assigned to each milestone representing the tasks to be completed. The success of each milestone was measured by tracking the state of its issues. A project review survey has been completed to assess each team member’s opinion of the project management.

Our project register of issues can be accessible with the link provided below:

<https://github.com/mattdear/temperature_transmitter/issues?q=is%3Aopen+is%3Aissue>

## Milestones

The project has been broken down into 3 sprints. The first sprint focused on requirement gathering; the second sprint focused on proof of concept and the final sprint produced the minimum viable product. Each team member was assigned issues to complete in each sprint, at the end of each sprint outstanding issues were considered and reassigned for next sprint when necessary.

|  |  |  |
| --- | --- | --- |
| **Gather Requirements** | **Initial Proof of Concept (PoC)** | **Minimal Viable Product (MVP)** |
| **Sprint 1** | **Sprint 2** | **Sprint 3** |
| Functional Requirements  Non-functional Requirements  Initial Solution Design  Project Plan  Creation of GitHub Issues  Present requirements to Sponsor | Working PoC  System Documentation  Initial Testing Methodology  Review of GitHub Issues / Milestones  Present PoC to Sponsor | Create MVP  Finalise system documentation  Review of GitHub Issues / Milestones  Testing and Evaluation/Results  Present MVP to Sponsor |

## Assumptions

Some assumptions for the project include:

* The team will be available for the required period 01/02/2021 - 21/05/2021.
* Key stakeholders will be available for the required oversight/governance.
* Team have the necessary technical skills.
* Any hardware sensors will be compatible with a raspberry pi.

## Constraints

Some limitations are as following:

* Completion of the project.
* Working remotely, we cannot physically attend site.
* No access to Arqiva server/hardware/database.

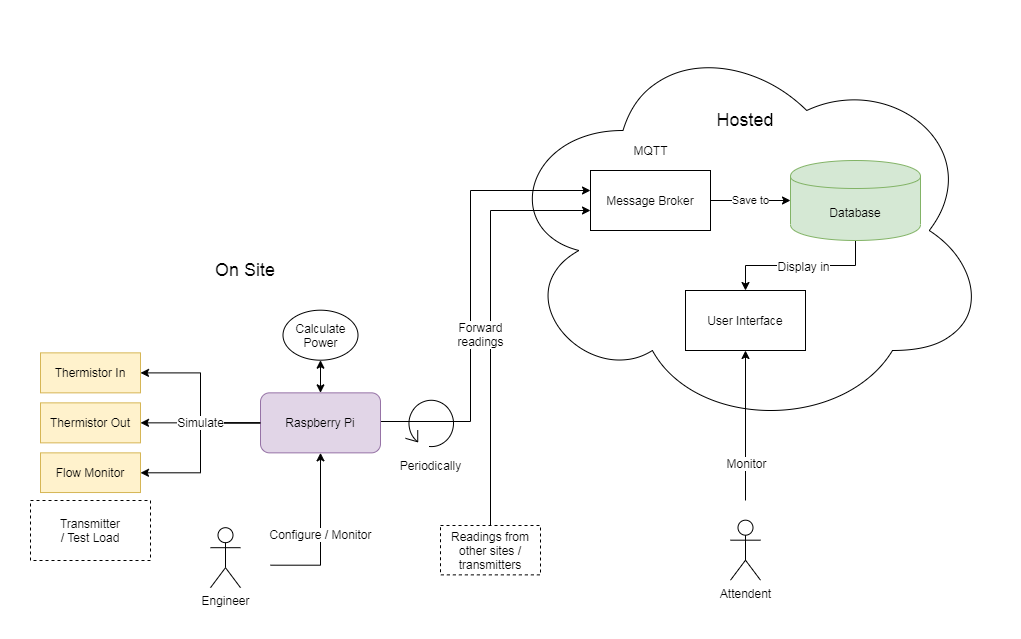
## Risk Assessment

Risks to project completion were evaluated and summarised (APPENDIX ?) in order to create relevant plans to mitigate them.

# Implementation

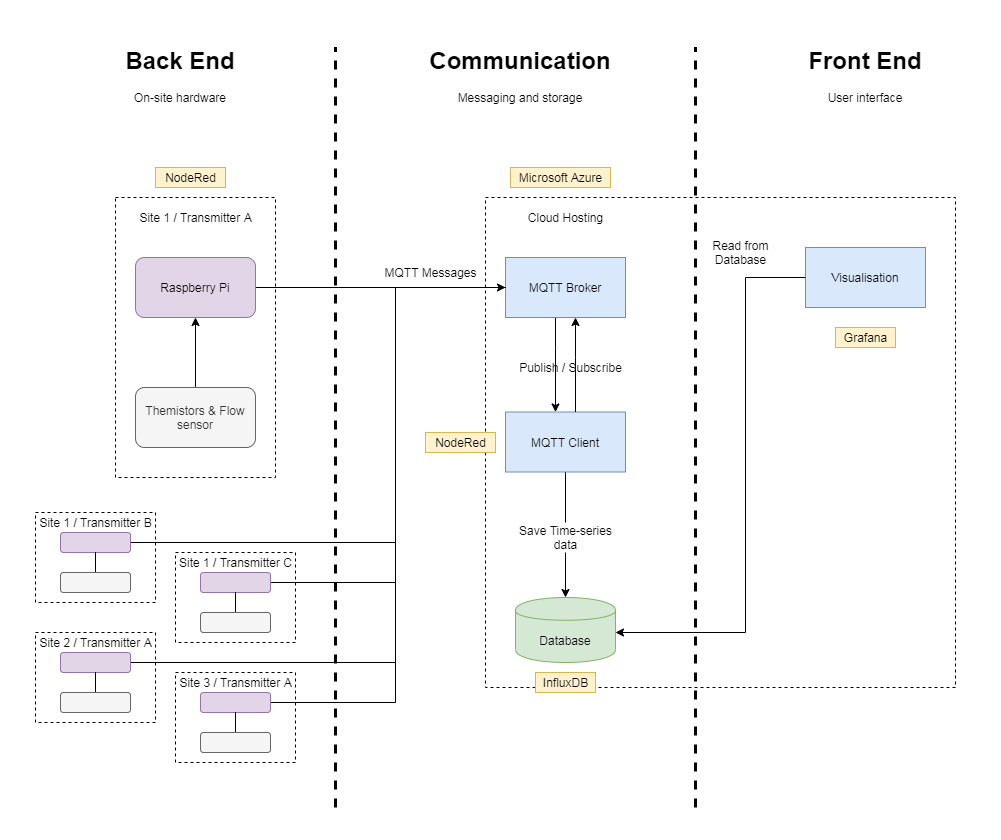
## Proposed Solution

The vision for the project is to create a system to allow the power to be measured from using two thermistors and flow monitors connected to a raspberry Pi, which will replace the analogue thermometers. This setup will be connected to test load to measure the temperature differential and flow rate to calculate power output. These reading can then be transmitted to a remote database to be monitored.



## Chosen Technologies

The architecture diagram includes the technologies selected for each component of the system and divides the system into three main areas: back end, communication, and front end.



### Back End

The technologies we chose to use for the back end were Node Red and Raspberry Pis to create a user interface for configuration on site. We chose to use this as it is simple and useful for creating rapid protype and a Raspberry Pi is both versatile and cost efficient.

We had considered temperature sensor which was a requirement that was out of scope in our initial requirement. However, it was considered to get an understanding of the calculations required and the communications from digitals sensors to Raspberry Pi. We had researched into thermocouples and other different types of temperature sensors but after speaking with Arqiva the thermistors worked better with the Raspberry Pi as it allows us to set the co-efficient and the resistance to calculate the power.

### Communication

For communications, the technology selected was MQTT, Active MQ, Influx DB and Node Red. The reasons for these are because MQTT is good for IOT Devices and the team has a good knowledge of the technology. In addition, Node Red was used to create a MQTT client for the consistency with the backend. Active MQ was used as the team had experience of the software and a good user interface. Influx DB is an open-source time series database that was initially selected as its cloud solution was cost effective however a self-hosted container was ultimately used.

The other options considered included Mosquito for MQTT however its interface caused difficulties during setup. OpenNMS has been successfully tested as an alternative to the client although due to time constraints could not be incorporated into the final system. CosmosDB and Cassandra were considered for the database however CosmosDB did not offer cost effective storage and Cassandra is not a time series database.

### Front End

For the front end of the system, we used Grafana as it is easy to use, open-source interactive visualisation web application that is flexible and customizable. It also works well with InfluxDB which was our chosen database to use for the project. Influx DB has its own built-in dashboard however these are less robust. Additionally, Grafana has plenty of plugins to extend it capabilities such as a map panel to visualize location co-ordinates.

Grafana can display alerts based on the imported data to the dashboards, additionally email alerts can be sent via an SMTP server which could integrate with the sponsors existing system.

### Infrastructure

To develop the project infrastructure, we used Docker and Docker Compose allowing us to leverage a microservice architecture alongside containerization. Docker containers sandbox each microservice making the system more secure by limiting their access to the rest of the system allowing them to be maintained and deployed independently. Containers also provide deployment flexibility because other than networking and compute they have no reliance on the host operating system. Docker compose was used to orchestrate the deployment and running of containers due to its simplicity and because like Docker it is free to use under the Apache 2.0 licence. We also used the Microsoft Azure cloud service to host the platform for development because we had free access at the time however, the system can be deployed on any alternative cloud platform like Amazon Web Services without changes.

Alternatives to Docker and Docker Compose were investigated with Virtual Box, Hyper-V, and QEMU being potential replacements for Docker but, were not used due their higher resource requirements and heavier reliance on a host operating system subsequently reducing the systems portability. Kubernetes was also investigated as an alternative to Docker Compose but was not used due to its enterprise focus and steep learning curve.

## Development of the System

The difficulties that were raised in the development was that we were unable to implement HTTPS, TLS, SMTP due to time constraints. The sponsor was looking to host themselves meaning it would be unnecessary to implement this and work completed would be superfluous.

We were initially unable to display points on the Grafana map panel until geohashes were implemented for the location data.

The solution does not incorporate physical hardware instead the system simulates the values that the hardware would be produce. These can be set through the user interface to represent the values changing in a production environment. This interface is also used to set the necessary coefficients for calculating power.

Automated deployment has been partially achieved as not all containers were able to be automated and some manual setup is required; this setup has been outlined in the developer guide.

The positive outcomes of the development include being ahead of schedule according to the milestones and the meetings with the sponsor was successful and well received. The twice-weekly team meetings worked effectively to complete tasks and assigning issues on time, we showed a good collaboration in problem solving.

# Evaluation

## Evaluation of System

Using the test plan, we evaluated the system developed during this project. The test plan and its results can be seen in (APPENDIX ?). Following the evaluation, the requirements have been grouped into: addressed, partially addressed, and not addressed.

|  |  |  |
| --- | --- | --- |
| **Requirements addressed** | **Requirements partially addressed** | **Requirements not addressed** |
| 3 | 1 | 6 |
| 4 | 2 | 7 |
| 5 | 11 | 9 |
| 8 | 12 | 13 |
| 10 | 14 | 17 |
| 15 |  |  |
| 16 |  |  |
| 18 |  |  |

Requirements 6, 7, 13 were not addressed due to being out of scope of the project. The hardware requirements below have not been addressed but were considered, such as types of sensors and potential costs. Due to challenges met when implementing HTTPS requirement 9 was not met. Requirement 17 was not addressed due to the time constraints on the project. Although false data can be manually input to see the response from the system.

The requirements 1 and 2 have been partially address as the data is not being read from the digital sensors, however this data is instead simulated. Although requirements 11 and 12 have not been met different types of sensors have been researched to create accurate simulations. In addition, requirement 14 is not part of the scope but has been considered when suggesting the use of Raspberry Pis.

## Further Work

Some of the work that would be continued/taken to the next step was security. This consisted of considering HTTPS, Apache Reverse Proxy, TLS. Security of the MVP was not completed as it was decided that any security protocols would need to be changed for the actual deployment. Other work considered was the selection of hardware to be used (sensors/flow rate) and an automated test harness that follows the test plan.

## What have we learnt?

Upon completion, the project review survey (APPENDIX ?) revealed that overall, the team had a positive experience. All team members agreed that Scrum was the correct methodology to use and were generally happy with the roles and tasks assigned.

# Conclusion

To conclude the transmitter temperature measurement project, this report has covered every aspect and approach of the project. From the start of the project where we identified the requirements and the scope of the project to aligning roles and responsibilities and tasks within the group. The majority of the project has been very successful with the main key requirements being fulfilled to their fullest potential which were put through testing and evaluating the project.

# References

ACTIVEMQ, 2021. *ActiveMQ* [online] [viewed 29 Mar 2021]. Available from: https://activemq.apache.org/components/classic/documentation

NODE RED, n.d. *Documentation : Node-RED* [online] Available from: https://nodered.org/docs/

RASPBERRY PI FOUNDATION, 2019. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online] Available from: https://www.raspberrypi.org/

# Appendices

## Key Contributions

|  |  |
| --- | --- |
| Name (Role) | Key Contributions |
| Mark Hartop  (Project Manager) | Organised team meetings  Lead team meetings twice a week  Created and delivered end-of-sprint presentations working with team members to gather information  Communicated with tutor and sponsor  Allocated tasks to team members to be completed  Assisted with front-end development  Issue management and updating on GitHub  Updated with team members for any issues/problems |
| Matthew Dear  (Infrastructure Lead) | Implemented Scrum  GitHub repository management  Researched and implemented Conventional Commits  Implemented and managed cloud system  Developed all Docker Compose files  Developed automated system deployment  Secured Node-RED  Researched HTTPS (Apache, NGINX)  Tested flows on physical Raspberry Pis  Researched Microsoft IoT hub  Created team structure diagram  Created Scrum to GitHub conversion diagram  Created Docker diagram |
| Matt Brook  (Front End Lead) | Created requirements list  Designed the wireframes  Implemented an architecture diagram  Ensured all data is visualised  Added a map plugin for map visualisation  Created dashboard alerts and researched SMTP  Populated data  Assisted in HTTPS  Created Grafana install guide |
| Kieron Gillingham  (Communication Lead) | Co-ordinated MQTT messages from Back End  Assisted with Back End flows  Created client to manage MQTT messages  Researched and set-up database  Assisted in displaying database data in Front End  Created developer guide  Researched securing MQTT messages  Dummy data population  Assisted with automated system deployment |
| Klea Cengu  (Documentation Lead) | Collated and reviewed documentation  GitHub Wiki  Created the Project Initiation Document  Created risk assessment table |
| Joshua Alsop-Barrell  (Back End Lead / Testing Lead) | Created Back End flows and user interface  Simulated a thermistor  Populated data  Example message  Use-cases  Requirements list  Test plan |

## Risk Assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Impact | Probability | Total Risk | Actions |
| 1 | Loss of work if power went down | 2 | 1 | 3 | As we are working remotely, other team members should be able to cover the loss of work and they should be backing up their work regularly |
| 2 | Team members sick | 2 | 3 | 5 | Ensuring there are multiple people on the team to cover whenever necessary |
| 3 | Issues implementing technologies | 4 | 5 | 9 | There should be a plan in place which may involve using alternative technologies |
| 4 | Unclear commit messages | 1 | 1 | 2 | The team should use the conventional commit standard |
| 5 | Loss of internet access | 1 | 2 | 3 | other members should be able to gain internet access and be able to cover the work |
| 6 | Team turnover | 1 | 1 | 2 | If a team member leaves the project before the complete date, other members should be qualified to cover/complete |
| 7 | Requirements change | 2 | 2 | 4 | New requirements should be evaluated to be adapted into the project and milestones |
| 8 | Scheduling problems | 1 | 1 | 2 | Unanimously adjust the schedule to fit the availability of the team |
| 9 | Inaccurate information/data | 2 | 1 | 3 | If the data ranges change, it should be adjusted in the program |
| 10 | Compromising/ignoring features of the scope | 3 | 2 | 5 | Reinitiating the requirements of the scope of the project |