Remote Automation Using Optical Fibre Communication

By Anthony Baudinette & Jarred Paola, 2019

# Table of contents

[Table of contents 2](#_Toc4531685)

[Document Control 3](#_Toc4531686)

[Contributors and Approvals 3](#_Toc4531687)

[Introduction 4](#_Toc4531688)

[Context and Objectives 4](#_Toc4531689)

[Scope 4](#_Toc4531690)

[Types of requirements 4](#_Toc4531691)

[Gantt Chart – Timeline 4](#_Toc4531692)

[Functional requirements 5](#_Toc4531693)

[Caveats 5](#_Toc4531694)

[Assumptions 5](#_Toc4531695)

[Non-functional requirements 6](#_Toc4531696)

[Security 6](#_Toc4531697)

[Use-Case Scenarios 6](#_Toc4531698)

[Appendix A 7](#_Toc4531699)

# Document Control

|  |  |  |
| --- | --- | --- |
| ***Date*** | *Task* | *Contributors* |
| **12/3/19** | General overview and layout of requirements analysis | JP |
| **16/3/19** | introduction, Objectives, Context Scope, and project overview | JP |
| **18/3/19** | Filling, listing, and sorting of requirements | AB, JP |
| **19/3/19** | Further functional requirements and non-functional requirements | JP, AB |
|  |  |  |

### Contributors and Approvals

This document has been written by Anthony Baudinette and Jarred Paola. Bill Corcoran has acted as the supervisor for this project, providing Anthony and Jarred with requests, direction and guidance, and has contributed and provided approvals as such.

# Introduction

Fibre optic cables are at the cutting edge of high speed and high volume data transfer. They are central to reliable and efficient wireless technologies, and require much research and optimisation to perform as intended, and as expected.

### Context and Objectives

This project aims to explore the ability to utilise optical fibre connections to remotely control another computer to perform specified actions. The desire is to be able to send remote commands to facilitate a communications test-bed in order to streamline academic research, without the need for a desktop screen or alternatively installed programs other than the coding language Python.

### Scope

Initial steps include setting up connections between a computer and a device using the Python language. From here, being able to communicate to the device from another computer through the first will provide a major hurdle. Translating this to a pre-existing set up of computers will take some time to adapt and execute. From here, having a running method of communication between device and local computer allows for the testing of remote access and scripting.

### Types of requirements

* Core requirements - Minimum viable product specifications which define the product in its simplest usable prototype state.
* Further extensions- developments that will add value to the end product but not fundamental to its functionality.
* Assumptions - existing technologies to be used in the project which require no development for this project.
* Caveats - limitations in place by the approach taken to the project that will have to be designed around.

### Gantt Chart – Timeline

[See Appendix A.]

# Functional requirements

Requirements are divided by semester 1 and semester 2, giving sense of timeline for the short term and long term goals for the lifetime of the project. These are furthermore distinguished via necessity, separating into those which are more integral to the ethos of the project, against those which create convenience and polish.

|  |  |  |
| --- | --- | --- |
|  | **First semester** | **Second Semester** |
| *Core requirements* | 1. Establish communication with external device – Python 2. Set up SSH tunnel between local and remote PCs – Python 3. Remote control of signal generator, oscilloscope, etc. via SSH tunnel | 1. Processing of oscilloscope data to display into other applications in real time 2. Automated sweep functionality via .csv or .txt files |
| *Possible extensions* | 1. Reading of oscilloscope data into .csv files 2. Passing oscilloscope data into other applications i.e. matlab on local device | 1. Build tutorial for usage of our methodology and functionality 2. GUI via python or a MATLAB app designer 3. (Any possible extensions from First Semester worth pursuing) |

### Caveats

* Desire for no extra programs to control embedded PC (i.e. desktop or screen sharing)
* Limited by Python’s speed on existing systems to link to devices. This may become an issue depending on age of embedded PC, and type of data we are transmitting

### Assumptions

* Serial link is large enough to transfer necessary data from oscilloscope to remote computer
* The oscilloscope is open to 3rd party application interaction, allowing access via python and external modules such that we don't have to use manufacturer's own program

# Non-functional requirements

### Security

* Since we are developing a method for remote automation, there is minimal personal risk regarding sensitive information, as we simply wish to establish the methodology.
* As per our assumption, there may be security measures when trying to access the device from an external PC

### Cost

* Nominal travel fees
* Making use of existing hardware and embedded PC’s at no cost

# 

# Use-Case Scenarios

Establishing a simple, easy to use method for remote access and automation of laboratory equipment paves the way for optimisations not only across the electrical engineering field, but various academic and professional areas of research.

1. The ability to script and automate testing without the need for physical presence to reset values and machinery
2. Operating and running tests for optimisation purposing, running near same-time experiments for like-conditions and best-case discovery

# Appendix A

