Remote Automation Using Optical Fibre Communication

Requirements Analysis

By Anthony Baudinette & Jarred Paola, 2019

Table of Contents

Table of Contents 2
Document Control 3
Contributors and Approvals 3
Introduction 4
Context and Objectives 4
Scope 4
Types of Requirements 4
Gantt Chart – Timeline 4
Functional Requirements 5
Caveats 5
Assumptions 5
Non-Functional Requirements 6
Security 6
Cost 6
Use-Case Scenarios 6
Appendix A 7

Document Control

Date	*Task*	*Contributors*
12/03/19	General overview and layout of requirements analysis	JP
16/03/19	introduction, Objectives, Context Scope, and project overview	JP
18/03/19	Filling, listing, and sorting of requirements	AB, JP
19/03/19	Further functional requirements and non-functional requirements	JP, AB
25/03/19	Added Gantt chart, Table of contents	JP
02/04/19	Clarified wording	JP

Contributors and Approvals

This document has been written collaboratively by Anthony Baudinette & Jarred Paola. Bill Corcoran has acted as the supervisor for the duration of 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 essential to reliable and efficient wireless technologies and require further research and optimisation to perform as intended, and as expected. Communication over long distances is currently easily created, but there are more opportunities yet to be explored.

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 from a remote computer to an embedded PC that is in communication with a network of facilities and devices, facilitating 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

On a small scale, we wish to be able to connect to an external device and communicate on basic terms, before attempting to communicate with another PC. As we scale up, communicating to a connected device on an embedded PC from a remote PC will provide the foundation of the project. From here, testing communication with different connected devices, and different types of connections, on the embedded PC will prove fundamental in establishing simple and clear instructions to replicate across different remote PCs.

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 project, against those which create convenience and polish.

	First semester	**Second Semester**
Core requirements	1. Establish communication tunnel with external device via Python 2. Adapt tunnel between local and remote PCs via Python 3. Remote control of signal generator, oscilloscope, etc. via 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. Sending files and executing on embedded PC, returning results. 2. Executing files on remote PC and sending requests to embedded PC to receive results concurrently. 3. Reading of oscilloscope data into .csv files from remote computer 4. Passing oscilloscope data into other applications i.e. Matlab on remote computer 5. Executing	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)
- Possibly creating executable files or other explanations to facilitate ease of use for cross-OS integration.
- Limited by speed of Python’s 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 via tunnel.

Assumptions

- Serial link is large enough to transfer necessary data from connected device to embedded PC, then communication tunnel is capable to forward to remote computer.
- The connected devices are open to 3rd party application interaction, allowing access via Python, Matlab, and any 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. Hence, as a proof of concept, an insecure connection tunnel is viable. Moving forward there may be a strong desire for security, so an SSH tunnel may become a better implementation for a general or “sellable” product.

Cost

- Nominal travel fees
- Making use of existing hardware and embedded PC’s at no cost
- Using two Raspberry Pi development kits for further testing and extrapolation

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.

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

Appendix A

ACTIVITY	Description	Start	Duration
Literature review	Research and explore previous works regarding optical fibre and remote communications.	1	2
Risk and Requirements docs	Documents to outline predicted and forseeable risks and project milestones.	2	2
Communicate with black box	Ability to establish link and send/receive information with a device.	3	1
	Talk to local port on computer.	2	1
	Receive information from black box at local port.	3	1
SSH tunneling method	Method of establishing a tunnel between one computer and another.	4	3
	Research possible methods of tunnel connection.	4	2
	Decide and implement tunnel connection.	5	2
Use device through SSH tunnel	Setting specified instructions to a device to obtain desired data back from device through communication tunnel.	7	4
	Send/receive information from device on other side of tunnel.	7	2
	Run commands and standard use of device through tunnel.	9	2
Automated scripts via SSH tunnel	Running scripts through tunnel on another computer for target device.	11	7
	Set up script to run multiple times.	11	2
	Set up script to save all results from multiple runs.	13	2
	Implement automated scripts via ssh tunnel	15	3
Handoff material	Develop user manual to handoff for potential users to implement and use system for remote automation	20	3
	Pull github code on anonymous computer to replicate work	20	1
	User manual for how to use and implement such technique on personal system	21	2
FYP material	Thesis paper, poster, video, and presentation to academic team, culminating in Spark Night	16	9