ECE4094
Project A

Progress Report

	First Name	Last Name	ID	Email
Supervisor	Bill	Corcoran	-	bill.corcoran@monash.edu
Student 1	Anthony	Baudinette	26949415	ajbau1@student.monash.edu
Student 2	Jarred	Paola	25958674	jpao1@student.monash.edu

Table of contents

Introduction

Our aim for the first half of the project was to establish the functionality to operate an AWG remotely, at the time of this reports publishing we have succeeded in communicating with the AWG and demonstrated uploading a file to the instrument, with this progress we are confident our initial version will be operational at the start of the next teaching period. And as such our added functionality will be built upon this in the remaining half of the project.

As the weeks have passed, we have monitored not only our progress in the project, but also our workflow, time dedication, and relative contribution. Compared to the start of semester, we believe we have taken reasonable consideration for the timing and throughput of our milestones, allowing reasonable goal setting and achievement of functionality. There have been some shifting and adjustments in terms of specific target dates of goals, but overall they have balanced out for an on-time delivery for the end of semester projection.

It is now our aim to reflect on the progress of the past few months. Here we will take into consideration our initial ideas and conceptualization of the problem and desired goals, our skillset, abilities, and newfound understanding of the components and functionalities, as well as our collaborative and creative process when contributing to the project. This is so that we can re-evaluate the remaining potential directions for the project and decide what is not only feasible but necessary and desirable given our reformed knowledge, in an effort to better prepare ourselves in utilizing the remaining time to deliver the best project we can.

Objectives

Our initial goal was to explore the ability of utilizing optical fibre communications to remotely control another computer. This goal was set to send remote commands to a communications test-bed, streamlining academic research by minimising required presence in a lab and allowing potential for automation.

The scope for this goal included setting up basic connection and communication between two computers, which was then to be extended to send commands from one computer to be executed on the other computer. This would functionally mimic the “Master-Slave” set up, with a computer in the lab connected to all of the equipment able to receive and execute commands, and return the results.

We opted to setup a GitHub Repository within the starting period of the project. This allowed us the ability to maintain version control, so that when we have a component of code or an aspect of the project working, we can return to it should it go wrong at a later time. Likewise, GitHub implementation across coding programs allows collaborative coding, so that we could both be working on the same files without interruption from the other, and collaboratively contribute together.

Progress and Outcomes

In the early stages, we looked in to the concept and tried to find previous works regarding this project, to little success. The concept is not a difficult one to grasp, and there are elements of various parts in different works, but the particular combination and execution within the context of lab equipment such as oscilloscope or arbitrary function generator where not available.

Exploring some of these components lead to research in the Python coding language, and the various libraries and capabilities it can provide. We hypothesised that Python would be able to provide a simple means of interaction and communication between two computers on the same network, over the likes of C or Java for example, as it is a widely used language for its more malleable coding.

This lead us to researching the various library packages python has available, and with regards to inter-computer connections, the ‘Fabric’ library seemed apt. There were many recommendations for this library as a way to establish the desired secure connection, but for us this proved difficult to execute. Considering our time limit and other factors to explore, we opted to use a generic TCP connection through a Python inbuilt library; something much simpler to execute and provided the necessary functionality to move forward with the project.

Likewise, we also explored the ‘PySerial’ library for Python to interact and use serial communication. This communication is what is used amongst instruments in the laboratory, and something we hoped to tap into and utilise as a part of our communications testbed. Although we were able to use this on one of the instruments, it appeared to create a bottleneck due to timeout limitations, as well as providing a similar problem in executing the library with various other instruments, so we opted to leave this library behind as well.

Since the serial implementation proved difficult, the next option was to use Visa commands through the ‘PyVisa’ library, as it is just an abstraction of the physical layer that drives the instruments and combines TCP, serial, PXI ext devices onto the one standard interface. This approach was successful in our testing with smaller instruments, and gives us some more options in terms of our implementation. Now that we have a method of sending commands to a certain machine and executing, and another method of sending messages from one computer to another in a wireless connection, our next step was to put together these two main components to establish our testbed of wireless execution of instruments from a remote computer.

With a prototype testbed connection established, we then attempted to implement this using the intended lab equipment, but encountered an unusual problem with the embedded computer; the instrument connected to the embedded computer does use visa commands, but they are through a PXI interface that the instrument also sits on, and the PyVisa library that we are using does not support this interface. But upon further research, the soft front panel that talks to the PXI interface has its own visa interface that acts as a translator to the

PXI, which we can then use and communicate from our established TCP connection with our PyVisa implementation.

Parallel to this, we noted that many of the machines operate using Matlab and on the assumption that the user would be “fluent” in Matlab so to speak, it would be easiest for an engineer or lab technician to interface with one software or programming language. With this in mind we explored the use of Python with Matlab, delving into the interactions between the two programming languages, and found relative ease in using Python through Matlab.

Works to be Completed

Looking ahead to the remaining time for the project, we aim to flesh out our testbed on the in-lab computer in order to begin testing and establish the functionality for the process with the relevant equipment. This will act as a catalyst for our project in terms of testing and execution, verifying that our initial conceptualization and prototyping of each element can cohesively work together and operate to achieve the goal of remote execution of the lab equipment.

Furthermore, we will be looking to explore what other devices may connect in the lab to provide further functionality. We will also be looking at exploring how well we can automatically execute on many files, leaving room for finding an optimal setting automatically rather than manually. Towards the final weeks of the semester, we will also need to finalise our work, collating it into a poster, video, and a final report, each to summarise and exhibit the usefulness of our hard work for this project.