Summary of Comments on Practical Work Report

Student Name..... **Markers Initials**_____

	Ranking			
Structure of the Report (compliance issue) All instruction about format have been followed, and all the prescribed sections are present	5	4	3	2 OR 1
Quality of the Report Grammar and spelling are good. The language is fluent, and flows well, no proof reading errors.	5	4	3 OR 2	1
Presentation of the Report The report is well focussed, balanced, interesting and contains an appropriate amount of detail. Illustrations are useful.	5	4 OR 3	2	1
Scope of Report Well-chosen and relevant to a Practical Work Report	3	2	1	0
TOTAL VALUE OF MARKS				

OVERALL Grade for this Report

Total Marks greater than 14	A
Total Marks greater than 10	B
Total marks greater than 6	C
You are required to resubmit this report.	D

It is recommended that you contact the Student Learning Unit for advice on the writing of this report

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

CERTIFICATE OF PRACTICAL WORK

Student Family Name		First Name	
Student ID Number		Specialisation	Software Engineering
Part Completed	I ☐ II ☐ III ☐ IV ✓		
Full Company Name	Teknique LTD		
Company Physical Address	Level 1, 104 Rosedale Road, Albany.		
Company Postal Address	PO BOX 300622		
Company Website Address	www.teknique.com		
Company Phone Number	+64 9 282 3132	Company Fax Number	N/A.
Supervisor Name and Designation		Supervisor e mail address	
Period Worked	From 14th November 2016	To	3rd March 2017.

Nature of Work (To be classified showing the hours for each classification)	Hours General	Hours Sub Prof
Mechanical and electronics assembly and configuration	200	0
Webservice Development.	0	100
Automate Testing.	50	50
Machine Learning.	0	300
Continuous Integration and deployment.	50	50
Total Hours Worked	300	500.

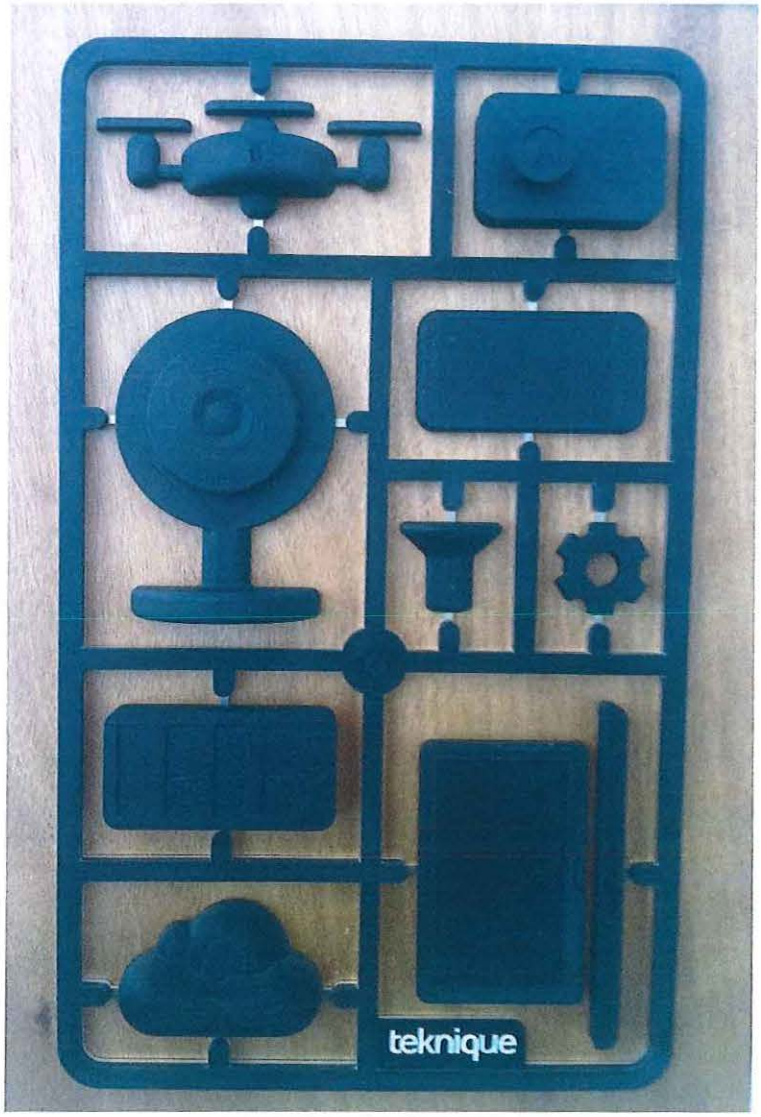
Report of Engineer in Charge

Signature		Date	27/02/17
Student Progress	Excellent	Student Diligence	Excellent
Student Attendance	Excellent	Remarks	Made excellent contributions to a number of internal projects, very impressed by his ability to tackle challenging problems.

FOR OFFICE USE

Name of Marker				
Reports Approved	General Hours		Sub Professional Hours	
Hours Credited	General Hours		Sub Professional Hours	
Grade ☐	Signature		Date	

Practical Work Report



Software Engineering (Department of Electrical & Computer Engineering)
March 5, 2017

Teknique
L1, 104 Rosedale Rd PO Box 300622 Albany 0632, Auckland New Zealand
Dates of work period: 14th Nov 2016 - 3rd March 2017

Summary

Teknique is a growing high technology company that is base in Auckland who specialise in creating camera solutions. The company have expertise in camera integration, image quality, assured manufacturing and software solutions for the overall camera's experience.

Throughout my time at Teknique, I was under the guidance of many different team leads who are more senior and experience to undertake many diverse and in depth projects. The projects ranges from physical prototyping, deployment automation to developing artificial intelligence for image recognitions. These projects are undertaken with the other three interns together as a team and generally closely with a team lead.

The projects are all heavily research and development based. They also require all of the stages of the full cycle of the production process. This starts with problem elicitation to designing prototypes to testing prototypes and to mass production. These projects require me to use mechanical system such as drills and 3D printers, electrical skills such as wiring and soldering the peripherals and many software development and general testing.

The projects I worked on are generally all internal projects. However, many of the artefacts will still be deploy in the commercial system such as the manufacture where cameras are produce, automate testing of camera quality and software to be deploy onto the camera.

The experience provides many opportunity to build on the skills that I gained from time at University but it also expanded in to many new areas that was not cover in my degree such as the hardware aspect and automation of the deployment of the products.

Overall, the projects are challenging but rewarding and interesting which made my time at Teknique both educational and enjoyable.

Acknowledgment

Teknique have work very hard to make sure that the experience are of high standards for the internship. Essentially I have my gratitude for everyone who worked there as they are all very talented and approachable. However, some went above and beyond to make the experience very educational and enjoyable.

First and foremost, I would like to thank my main technical supervisor software Engineering and team lead. Even though under the incredibly busy schedule for his projects he still spend a great amount of time designing projects, checking up on the status and provide technical help for the interns. I am also thankful for the conversations and foosball games at lunch time.

The team leads J , devops engineering who I work closely with throughout the projects for the expertise and experience they pass onto me. They never failed to assist me with any problems I have to help me become more experience in producing higher quality work.

Thirdly, I would like to thanks Lead UX Designer and I Office Manager for designing the culture and experience for the company. In particular, also design the experience and guide me through my first few days of the internship. on the other hand organises many parties such as the thanks giving and Christmas party and tiding up for the engineers making us as productive as possible.

I will also like to thank y for providing the internship opportunity and also the immense vision he has for technology.

Lastly, I will like to thank the other interns for making many of the project possible through our teamwork and different perspectives they bought into discussion. Especially to or our tag team and foosball games.

Contents

Table of Figures

1 Introduction

This report covers my practical works experience at Teknique over the period November 2016 to March 2017. This report will give the details of the structure and staff organisation of Teknique, a full description of the type of work I undertook during the internship and the skills and knowledge I gained. The report will conclude with the evaluation of my learning as well as the performance and organisation of Teknique.

Teknique was established in 2004 by Ben Bodley and Will Bodley. Since then Teknique has been helping clients in generating creative, efficient, profitable and problem solving solutions in a wide range of works. From the company's establishment the business has taken a turn from many general problem solving toward a focus in camera technology. Teknique works closely with many manufacturers, the clients and the users to provide camera experiences of the highest quality. Over the years Teknique have won numerous awards in technology including recently rank 8th in Delcitte Fast 50. Teknique now provides over ten oversea clients with a wide range of camera solutions ranging from ip cameras, sport cameras to drones.

I was employed at Teknique in an internship position. I spent the summer over 2016 and 2017 to work on many projects that required large amount of skills from a wide range of areas from mechanical to software. During the projects I will be focusing on a project initially till meeting a blocker such as waiting for parts to arrive and switch to working on other projects. Hence, many project are in progress simultaneously. The work include product reviews, mechanical assembly and configuration, setting up electrical system and research and development of software. The technology used including basic tools such as drills, screws, saw and fills for mechanical system. When setting up electrical system tools such as solder is used. Other hardware knowledge is gained for raspberrypi, labjack, various sensors and PCB. I have also learnt to use Sketch up and Cura software to make 3D printing models. For software development a wide range of technology is used which includes Python, Ruby, C++, Qt, bash, Jenkins, docker and TensorFlow. I have also learnt and improve on many general skills such as problem framing and presentation skills through out the projects.

2 Company Profile

2.1 Layout of Office

The Teknique office is located at 104 Rosedale Road, Albany, Auckland. The companies is close to the industrial park and local food stores which provides a large selection of possible eatery options. The team moved into the lot about four years ago. The building consists of a level 1 and a level 2. The majority of the area is available at level 1 where as level 2 is much smaller and only have about 1/10 of the area of level 1. As the building is relatively new, Lead UX designer have design the layout of the buildings to improve experience. There are lots of windows so there is many natural lights. They are all located in views with many tree to allow a lot of green to rest eyes when tired. The walls are designs to have a common super hero theme. There are many plants located all around the office. Tables are arranged into four tables for each section. There are about 9 sections which are roughly split my the teams. The sections are separate by shelves. The sections and tables are all open to encourage the team to communicate with each other. Other areas include three meeting rooms, one hardware section, the optics laboratory and a testing room and a large kitchen and dinning area. The only main downside of the buildings I thought is that there is no windows.

2.2 Staff Organisation and structure

Teknique approaches culture with a very anti-corporates style. They don't believe in textbook or rule based way to interaction. Essentially the management is very flat. The staffs are split into teams.

- product evangelist and the founder of the company, who is also the boss and led the company.
- Other technical management roles include project manager com-mercial manager VP of Engineering . They manage the deadlines and business side of the products.
- People management team includes executive assistant, office manager and , Lead UX Designer. The people management team is a mix of HR, finance and providing best experience for both the workers and the customers.
- There are several product development team at Teknique. Each team are focus in working on a product and have about four members. These teams are lead by
- A specialise team is the hardware team led by
- The testing and quality team is led by

2.3 Technical Facilities and Amenities

The office have a brilliant kitchen, dinning and relaxation space. The kitchen include many useful facilities such as toaster, coffee machine, toast maker and a kettle for people to make breakfast, lunch or snacks. There also is a fridge to kept our lunches, milk and

drinks. The company also provides many kitchenware, utensils and spices for the worker to used. There are two tables for dinning and a couch area with magazines for the lunch time to read. There is also an ipad that can be used. There is also a big screen for the project progress status and can be use for presentations and other tv usages. A section of bean bags is also available for more relax settings.

In the dinning area there is also a beer keg that can used during parties or for a Friday afternoon for those desired to drink. The company also provide many fruits and snacks for developers to enjoy during their breaks. There is also a foosball table that many developers enjoy using during lunch breaks. An arcade machine is also provided for those who enjoy the nostalgic games.

Everyone have a standing desk with plenty of room for notebook the computers and hardware or product under development. A drawer is also supplied to store useful parts. Some employees also have been supplied with laptops in addition to the all in one computers. There is also a lot of hardware, mechanical and electrical supplied available to every one to use for development and research.

The company have a weekly Monday stand up meeting where everyone will come together and have a catch up of various status follow by the management team announcing various important notices. Every fortnight during the Monday morning meeting there will be breakfasts. Every Friday most employee will also enjoy a drink and some snacks and discuss problems with a more relax mind which tends to solve most of the problems. Sometimes during these times announcement or presentations will be made. Every once in a while when there are important dates the company will also celebrate these with team parties. The parties are pleasant as family members are also invited as well.

3 Description of works

Upon arriving on my first full day of work at Teknique, Nick took me through the company history, believes, practise, goals and perhaps most importantly why Teknique exist. We then go around the office to meet everyone and the different teams. Afterwards we continue the induction to looking through the engineering and business process of the company. We then set up our own machines with the correct softwares.

Throughout the 14 weeks period of the internship many different projects were undertaken. The experience consisted of research, prototyping, design, development and deployment. Research involving finding out what is the state of the art of the technology and what is feasible to be implemented. Prototyping require both physical set up and software interface to be created to test the proof of concepts. Design including picking parts that are cost efficient and durable and making system with robust software and hardware architectures. Development is usually done as an individual where integration is needed for some of the projects from the team which will later be presented to and review by our supervisors. Sometimes a project will also be presented to the entire company. Deployment includes physically setting up system to be use or create automate systems for software to be run.

The JIRA system by Atlassian is used as the project management system. This requires planning at the start of each project to design how the project will be tackled. Tasks are broken down into manageable size for a person to be take on individual and time of each task are estimated. The team then pick the tasks they will be approaching in each sprint. Here sprint is a development period. Time length of the sprint is a week.

3.1 Product Reviews

Before diving into the engineering aspect of the internship the first few days were focused on reviewing many different products. This exercise requires the interns to work together to set up the cameras from scratch and writing a detail and visual report on each products. This including reviews on the packaging, unboxing, reading manuals and instructions, setting up the camera physically and in many cases signing up to apps or website to use the camera. This exercise was interesting and useful in many ways. First it introduce us to the sort of product we will be using and developing. Secondly, it show us the usability of the different cameras. Lastly, by using the camera myself, I can know what to improve the existing product on. From reviewing we find that existing product solve almost no problems at all. Some premium product are actually a lot less usable and effective than other less advertise product.

3.2 Data Logger

The first project the team was given with is to create a data logger. The purpose of this project is to develop a data logger system in both the hardware and the software parts which will log various data such as temperature, humidity and current of the environment of interest. I start off the project by estimating the sort of tasks that need to be completed for this project and estimated the time length for each of the tasks. Next phase is the research phase. As this is the first project, most of the research is in the background

information of the hardware we are given with. We are to use different sensors connected to a labjack T7-Pro model as shown in figure 1 to interface with the raspberry pi to log the data. A main design decision that the team had to make was which web service library to use. After some research we have decide upon using the flask library as it is well supported for raspberry pi environment. The project requires me to assemble the system together such as connecting the various wires to the right locations on the labjack. The labjack is connected to the pi through ethernet and usb cable. Some soldiering was used in this project to connect wires to the sensors. Python is the language I used to get data from the sensors and publish to the web service. The data is processed and save in csv format to allow easy visualisation of the information sensed as csv can be open in excel for plotting. Finally the last step is to set up the data logger system to the physical testing location. This also require setting up the software web service system on the pi to run so user can use the web service to interact with the system. The user can choose what sensors to start and stop, polling frequency (i.e how frequent to sense) and downloading the data from the sensors in csv format.

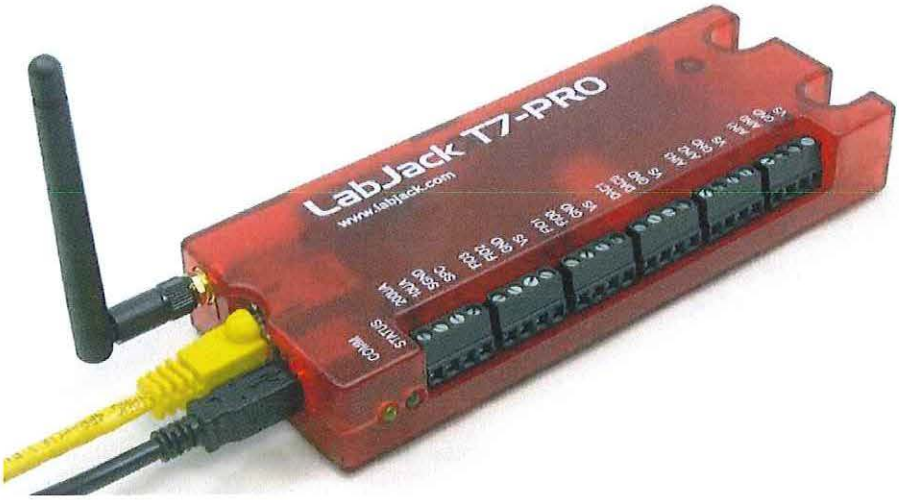


Figure 1: Labjack T7

3.3 Pi Image creation

Teknique is in the process of automating as many processes as possible. One of the process to be automated is the creation of the environment for the deployment. Traditionally, when a new system is required, a person will go through the following steps:

1. Manually download a Raspian image for the Pi
2. Flash the image onto a SD card
3. Resize the SD card to desired partition size
4. Install all the required software for the desire environment
5. Setup the system to have correct settings
6. Download or copy the necessary files

The task is to automate and provide streamline of this process. Jenkins is an open source automation server which provide deployment functionality as shown in figure 2.

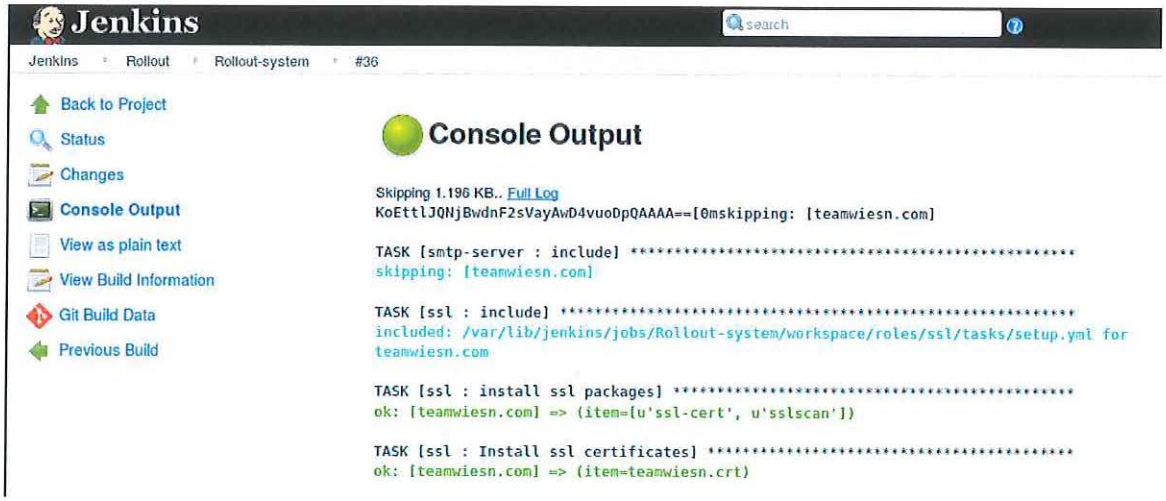


Figure 2: Sample Jenkins job console output

Jenkins provides a very powerful continuous integration feature that will automatically check for changes in the repository (code base) and trigger the build, in this case the creation of the image. The project is split into two main parts. First part is the automation script to set up the desire environment. The second part is to configure the Jenkins job to integrate with the repository and run the automation scripts on the correct device. In the first part it requires the script be written in bash. The script will download the latest image, resize the image to the correct partitions, assign desired setting, installed require software and download or copy files into the image. The second part is using the web interface of Jenkins to configurate the scripts to run and the frequency of checking for the changes in the repo. It is also require to set up where this automation process will run. The delegated system is called the slave node which is chosen to be a raspberry pi. In both phase a lot of manual testing is done by flashing the image onto the SD card to check if the automation script is functioning work. The integration of the two stages are proven to be more difficult then expected as there was limited number of pi so the code was develop mostly in a linux virtual machine(VM) rather than the pi. Another problem was that the automation script is operating on an image file on the raspberry pi hence it was different to operating directly on an image due to access rights. This require me to changes many of the code and add in many access right for the script to run. Running and testing the build job will also take a substantial amount of time due to the time required for setting up and the limited hardware power of the pi. As an end result, the streamline system is correctly functioning. Now, other developers are able to get a completely new pi set up in a few minutes compare to previously when hours is required.

3.4 Command Line Integration

After the completion of the image creation project I moved on to the command line integration. This project is similar in spirit to the previous project where automation is the main purpose. An external software that Teknique use to flash firmware onto the camera evaluation boards uses gui. This is not usable in a headless system (i.e without screen and keyboards). Also for automation using the command line is preferred over

gui. I was task with creating a command line interface of the software to allow raspberry pi flash the firmwares. This requires a few interesting new systems such as the hardware camera boards, Qt and C++. Qt is a cross platform software development framework for embedded system and desktop. It provides useful functionalities for interfacing with embedded system. This task requires me to research through the Qt framework to find the functions the project needs and look through the external software source to figure out what changes need to be made. One major experience is the code quality required. As this is a system that the company use often it is a high value piece of work that require high coding and documentation quality. In this project I learnt the code reviewing process and more about good coding quality.

3.5 Filter Wheel

Developing in parallel with the command line integration project is the filter wheel project. The filter wheel project is the first project that the interns are responsible for all parts of the process. This includes the following: requirement elicitation, research about component, deciding which component to purchase, 3d modelling, assemble discrete parts and develop of software interface to control and use the sensors and integration with existing software systems. The project starts off with reading and understand the brief. As the brief is a summary of desirable feature and limitations, many questions were asked to elicit more information about the final product. This elicitation also continues throughout the project to clarify ambiguity and uncertainties. After this initial phase of information gather the team start to design the prototype. Many aspects are considered. This includes critical path planning, costs, robustness, accessibility, ease of use and engineering details. As the team need to order components to allow the system to be useful, hence, this was completed first as it will cause a blocker for waiting them to arrive.

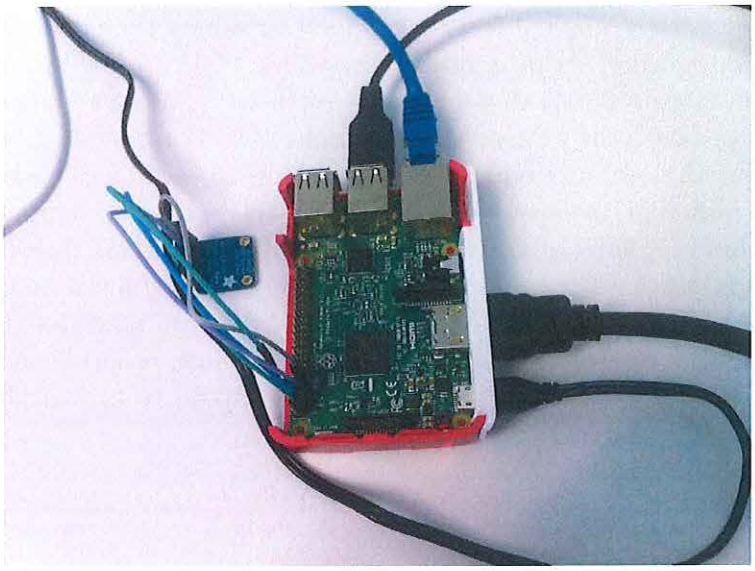


Figure 3: Raspberry Pi Setup

The problem is analysed to determine the necessary specifications required. These include what torque the motor can support and the voltage needed of the sensors and controllers need. Other consideration are costs and accessibility. This means we will look

at local store and what they offer rather than overseas as it will take a longer delivery overhead. It is interesting to see that the other part of my degree, my conjoint in physics was used in aiding the designing the prototype.

The filter wheel is also carefully designed to cover constraint such as distance to the targets as the system will need to fit in a confined space. The filter wheel will also meed to be a specific size to cover the desire field of view. Thickness is also an engineering trade off between robustness, cost and weight. The desired filter wheel is to be light, low cost, easy to manufacture buy still be robust. The filter wheel also need to support the motor so that it can be turned by the motor. After the design is done the filter wheel prototype modelled on Sketchup, a 3D modelling software. The software allow the design to be export to files that are printable by a 3D printer. Printing the model with 3D printer is long and difficult job as due to the speed and size of printer, the model cannot be printed at once. Hence, different parts are printed and assembled together.

While waiting for the 3D printing process to finish, the code for the controllers and sensors are started on. Libraries are found for communication to the hardware so that it allows the creation of the high level drivers for the peripherals. Testing the peripherals is interesting as the team get to see the results from the hardware such as motor spinning and stopping when it detect it is at home. The detection is the done by a hall effect sensor which essentially uses magnetic field to detect location of the wheel. The code is refactored into a python interface library allowing it to be used easily.

Next stage is to assemble all the different parts together. This include connecting device under test (DUT), the raspberry pi, the filter wheel and all other peripherals like sensors together. A wooden stand is made to allow the DUT sitting on top close to the filter. Different fixture are also 3D printed to hold the some sensors and the motor in place on the wooden stand. As there are over 5 sensors and controllers and each have many wires, wiring everything together was a difficult process. Some specialise wire are create as they need to be in longer length or that they need to split due to limited port on the raspberry pi. Sometimes when a wire break it will take a lengthy process to test which wire is broken. The DUT also have many cable such as usb and power cable connecting to the raspberry Pi. After all set up the system is able to run.

Final step is to integrate the python library with the existing testing system. This include importing the library into the testing system and calling the correct functions for the correct tests. After the integration is completed, the whole system is tested. As a result, for previously test that was not automated will change from pass to fail with the new automated filter wheel. This allows the test system to become more automated and removing manual steps such as physically putting a filter in front of DUT when the test is running. In additionally, sensors are also added to provide new functionality in capturing the status of the DUT.

The filter wheel project is one of the project that was presented on. This occred on the afternoon of one Friday. During this session we should the screen in the dinning area to show our presentation follow by a live demo of the filter wheel system.

3.6 Siren of Shame

Siren of shame is a siren that is design to go off when someone breaks the build as shown in figure 4. This will allow developer respecting the build as they will be traumatise by the Siren of shame. Hence, fewer broken builds and more higher quality build. The purpose of this project is for me to set up the Siren of shame. The requirement for the project was not well-defined as no one was certain what the Siren of shame was capable of. Some desirable features were flash the siren, making annoying sound. announcing what build break and who broke it. Through research I found that there is a library written in java which is provided by the Siren of shame sellers that allow the users to interface with the siren to light up and make sound. Next part is designing how the system is going to get the build informations. Although the Siren of shame sellers provide some support of the siren with automated server such as Jenkins most of these are gui-based . Hence I create the interface myself. To provide the desirable features I create Jenkins jobs which will be trigger to turn on the Siren of shame if a build breaks. For the jobs under monitor configuration is required to pass on the build information on to the Siren of Shame job. Lastly, using the espeak text-to-speech library I will announce which build break and who broke it. Finally, after testing the full system the Siren of shame it is deployed. Within a first hour there is already victims of the siren which induce many laughters. The Siren of shame is well received by the developers.

Figure 4: Siren of Shame - The big Siren

3.7 Porting Ruby Project to Java

One of the project is written ruby and to provide for obfuscation it is required to compile the ruby source into Java. Ruby is an interesting language where one of most popular and standard implementation of the language is will run the code on a C interpreter. In this implementation essentially the direct source of Ruby is used with no obfuscation. Another implementation of the language is Jruby, which is running compile ruby code on the Java virtual machine (JVM). To provide security in deployment I was task to compile the project into Java. I started off this task by discovery possible ways to run the ruby on Java. Different methods were attempted. However, Jruby was chosen due to its popularity and functionality. After compiling the ruby source to Java class, testing

the deployment is done with docker. Docker is a powerful tool that allow easy build, ship and running of software. It provide a contained environment similar to but more light-weight and is running the software directly without all the operating system emulation. In depth configuration of the docker files is required to run the Jruby application as Java implementation libraries is required instead of the native ruby libraries. As a result the deploy system can now run using the Java implementation of ruby to provide obfuscation of the source.

3.8 Filter Fan

This project is similar to the filter wheel project with the main difference of the requirement of having two filters instead of four and instead of using raspberry pi to interface it uses a labjack. Similar work is require for this project as before but with experience and many of the previous artefact the development time is a lot faster. Using the labjack is less convenient than the Pi however majority of the previous code was portable to this project with some syntactical changes. A main difference is that a different motor is chosen. This new motor is more robust and heavy duty.

3.9 Image Recognition

The image recognition project is an interesting project which required using machine learning to perform image categorisation. Tensorflow is used as the neural network library to perform machine learning. Tensorflow is a library produced by Google. It provides the basic underlying mechanism of how neural networks works. This means that it will perform the tedious task of back-propagation and other general task like how it utilise the cpu or gpu to train. The project have three main aspects, that is gathering data, making and training the models and deploying it on to the device.

Neural network is loosely but not exactly related to the brain. It models the ability of the brain in recognising patterns. It is essentially an optimisation problem this means that with the data we train on it will adjust its final predictions to give the over most accurate guess for different situations. For example, we might want a neural network telling us what is the price of a computer. A model can be something like a list of parameters such as processor, graphics card, ram, hard disk and others. The data we feed it is the computer information and their prices. The neural network will adjust its parameters for each data that it is fed. As a result it will be able to give a reasonable estimate of the price of computers.

In generating the dataset require for training the neural network, the team source many pictures and video and manually categorise them. GIMP, a image editing software that is similar to paint but more powerful is also use to help editing the images to remove unused parts of the picture. A main part of dataset is selecting which images are to go into the training, validation and testing set. The training set is the set that we feed the neural network to train on and this set will affect the model the most. The validation is a set that get tested on once in a while to check if the model is training well and not over fitting. Over fitting is when the model get to adapt to a specific data. For example if you train the computer price model with only computer with good graphics card and you feed it a a computer with bad graphics card for estimate the result will overestimate

the price of the computer. Validation set is generally a subset of the training set hence is similar in variation. The result from validation generally are not used to affect the model tuning but rather act as a feedback for the developer to know if the model is doing well. The testing set is a final set that the developer want to test on. General this is like the 'real world' set where if it perform well on these random data it should perform well in general. These data should be independent of the training or validation set. Hence, in generating our own sets we use independent footage and images for the different sets.

Creating model is difficult as there isn't any theory in what a good model should be. Most models are base on empirical results through trial and errors. Although this is the case but some general techniques can be used. These includes using multiple layers and using convolution layer. The idea behind convolution is that it will divide the data for example an image into different parts and use each part to be the tuning factor. Suppose you want to identify picture with or without computers in them, it will make sense to look for the computer in human sense. However, for naive neural network they don't understand this so they will just compare everything in the picture. Thus with different backgrounds it could fool the model. Using convolution layer avoids this. As convolution will look into each region the model will realise the interesting part is the computer rather than the background hence not be fooled. Concisely convolution is a feature pooling technique. Even with the techniques, other parameters such as the size of the convolution window and how large is each layer is tested on to generate a model that will produce the best results.

To deploy the artificial intelligence on the device requires many consideration. As the device has a much more limiting hardware and software capability, factors like correctness and speed performance, size and compilation are all required to be considered. This is proven to be a difficult task as the neural network will require a large amount of data hence pushing the limit of the size factor. Compression is done the model to reduce its size allowing it to be deploy onto the device. Speed performance is optimised via refactoring of the code and also compilation optimisation. Cross compiling is performed to compile all the code and libraries required into one single executable files from the host machine on to the device architecture. As a result the categorise AI systems is able to be run on the device with reasonable accuracy.

the deployment is done with docker. Docker is a powerful tool that allow easy build, ship and running of software. It provide a contained environment similar to but more light-weight and is running the software directly without all the operating system emulation. In depth configuration of the docker files is required to run the Jruby application as Java implementation libraries is required instead of the native ruby libraries. As a result the deploy system can now run using the Java implementation of ruby to provide obfuscation of the source.

3.8 Filter Fan

This project is similar to the filter wheel project with the main difference of the requirement of having two filters instead of four and instead of using raspberry pi to interface it uses a labjack. Similar work is require for this project as before but with experience and many of the previous artefact the development time is a lot faster. Using the labjack is less convenient than the Pi however majority of the previous code was portable to this project with some syntactical changes. A main difference is that a different motor is chosen. This new motor is more robust and heavy duty.

3.9 Image Recognition

The image recognition project is an interesting project which required using machine learning to perform image categorisation. Tensorflow is used as the neural network library to perform machine learning. Tensorflow is a library produced by Google. It provides the basic underlying mechanism of how neural networks works. This means that it will perform the tedious task of back-propagation and other general task like how it utilise the cpu or gpu to train. The project have three main aspects, that is gathering data, making and training the models and deploying it on to the device.

Neural network is loosely but not exactly related to the brain. It models the ability of the brain in recognising patterns. It is essentially an optimisation problem this means that with the data we train on it will adjust its final predictions to give the over most accurate guess for different situations. For example, we might want a neural network telling us what is the price of a computer. A model can be something like a list of parameters such as processor, graphics card, ram, hard disk and others. The data we feed it is the computer information and their prices. The neural network will adjust its parameters for each data that it is fed. As a result it will be able to give a reasonable estimate of the price of computers.

In generating the dataset require for training the neural network, the team source many pictures and video and manually categorise them. GIMP, a image editing software that is similar to paint but more powerful is also use to help editing the images to remove unused parts of the picture. A main part of dataset is selecting which images are to go into the training, validation and testing set. The training set is the set that we feed the neural network to train on and this set will affect the model the most. The validation is a set that get tested on once in a while to check if the model is training well and not over fitting. Over fitting is when the model get to adapt to a specific data. For example if you train the computer price model with only computer with good graphics card and you feed it a a computer with bad graphics card for estimate the result will overestimate

the price of the computer. Validation set is generally a subset of the training set hence is similar in variation. The result from validation generally are not used to affect the model tuning but rather act as a feedback for the developer to know if the model is doing well. The testing set is a final set that the developer want to test on. General this is like the 'real world' set where if it perform well on these random data it should perform well in general. These data should be independent of the training or validation set. Hence, in generating our own sets we use independent footage and images for the different sets.

Creating model is difficult as there isn't any theory in what a good model should be. Most models are base on empirical results through trial and errors. Although this is the case but some general techniques can be used. These includes using multiple layers and using convolution layer. The idea behind convolution is that it will divide the data for example an image into different parts and use each part to be the tuning factor. Suppose you want to identify picture with or without computers in them, it will make sense to look for the computer in human sense. However, for naive neural network they don't understand this so they will just compare everything in the picture. Thus with different backgrounds it could fool the model. Using convolution layer avoids this. As convolution will look into each region the model will realise the interesting part is the computer rather than the background hence not be fooled. Concisely convolution is a feature pooling technique. Even with the techniques, other parameters such as the size of the convolution window and how large is each layer is tested on to generate a model that will produce the best results.

To deploy the artificial intelligence on the device requires many consideration. As the device has a much more limiting hardware and software capability, factors like correctness and speed performance, size and compilation are all required to be considered. This is proven to be a difficult task as the neural network will require a large amount of data hence pushing the limit of the size factor. Compression is done the model to reduce its size allowing it to be deploy onto the device. Speed performance is optimised via refactoring of the code and also compilation optimisation. Cross compiling is performed to compile all the code and libraries required into one single executable files from the host machine on to the device architecture. As a result the categorise AI systems is able to be run on the device with reasonable accuracy.

4 Reflective appraisal

4.1 Impressions and performance

Teknique is a fast growing production engineering company that takes pride in the work and developers. For example one newly hirer engineer for around a month was sent aboard to talk to oversea client as there is great trust in the ability of the developers. This means that everyone is well respected and often directly in contact with the clients. The company also like to keep it very open to everyone as every one ideas are appreciated and sought after.

Teknique consistently looking to improve their process. Two main process improvement is automation and development process. I was privileged enough to help the company in some automation process. This include test, deployment and development automation. As Teknique deal with many hardware so the automation process is even more complex and difficult. This include sending signals through software or hardware to devices to emulate interactions. In terms of development process the company are looking to workout how to better estimate their tasks length and complexity. The company are also working on better documentation to allow standardise streamline process. Through these improvements the company is performing better.

Although the developers are split into teams but many will work on different products at the same time. The developers will also be working on many parts throughout the project similar to the internship. This is different to some other companies that will people will specialise in specific areas. This has it beneficial as everyone will be able to see the full spectrum of the works. This allows everyone to take greater ownership and capability to understand each other's work.

However, due to the above reason it was difficult for the interns to fit into the whole development process as the experience required is immense. Since the interns do not have the background and is only at the company for the summer holiday thus unable to allocate the interns on client facing projects. Hence this result in the interns working on many internal project these ranges from improving process, prototyping and researching. That being said, interns were able to gain a lot of new skills in the production process and know the structure and process of the company well.

4.2 Skills Gained

I spent a significant portion of my time (around 40%) at Teknique working on testing, configuration and prototyping. During this part I was able to learn and hone skills in mechanical and electronic skills. These skills are difficult to obtain through university studies due to the degree structure. I also gained some skills in 3D printing when prototyping for the projects.

The remaining 60% of my time was spent on working on many different software engineering. These including writing drivers for controlling controller, interfacing with sensors to get data, test and deployment automation and web services. This tasks requires a wide variety of skill set which many are new to the interns. Another interesting task I worked on is using machine learning. AI is always are topic I am interested but

unable to fix into my university schedule due to the limit classes I take as a conjoint student. Through the project I was able to use TensorFlow framework to train neural networks to provide machine learning system.

Many of the projects require a wide range skills that is difficult to experience through school work due to limited resource and time. In particular, I was able to gain a lot experience in embedded software engineering. This is very rewarding in my opinion as it is using a lot of software to build many 'cool' things.

Throughout the project I seen how the team interact internally and externally as there were oversea client visiting. The main skills I learnt from these is how to convince others on a good belief and how to make it a win-win relationship for everyone. This encourage me to communicate and become less shy with many of the other developers. I was able to assist others on the work I produced for the company and was one of the most rewarding part of the internship.

4.3 Lessons Learnt

An important lesson learnt in the work experience is the production life cycle. During university studies although many project were undertaken and many skills were gained out of them but none where able to mimic a real production system. This ranges from research to deployment and from mechanical system to software system.

Another important lesson learnt is the way software engineering process works. Although, no one know exactly what is the best way to produce software but from experience the standard water fall model does not work well. Even now the more accepted and in use method agile, has its own flaws. From my experience I seen through out my time at Teknique, I felt that a flexible agile method performs the best. This is a method that essentially follows some agile principles but not strictly. For example one agile method is scrum which have a principle that their is a 15 min stand up meeting daily. For those who follow this strictly will time the meeting and stop regardless the progress of the meeting. However, intuitively and practically this is certainly not the best. Teknique will approach this with a more flexible mind such as only holding meetings when necessary and only as long as needed. This also synergies greatly with its culture aspect that what works the best is depending on many factors hence there is no textbook solutions to these.

5 Conclusions

In conclusion, I found myself very lucky to have worked at Teknique over the summer break of 2016-2017. The company could have let me be doing the same thing for the entire internship but instead put in a tremendous effort in organising projects. Despite the extremely tight schedule the company was under, many of the team leads will spent a great deal amount of time to design a project that will teach me many skills and provide professional experience and support for the team when required. Due to their willingness in expanding the skills of the interns as much as possible, I was able to learn and hone a wide range of skills. These skills include production system work flow, embedded system engineering, deployment methodologies, continuous integration and deployment, machine learning and many others. These skills will help me greatly in my professional life as a software engineer or as an academic.

I found that working with my fellow employees at Teknique to be a very positive experience. Everyone at work is very approachable and happy to help each other to reach the common goals of the company. The team consist many talented individuals with great ideas and are keen to share them. Many of the employees also held degree of master or PhD which was a real treat to discuss their ideas on the industry and academia. I especially grateful that the team will provide intellectual help when required. This means that they will not just simply given or seek the answer for me. But rather give ideas and perception of the problem. As many of the problems the company and myself work on have never me done before so we are the first one to seek an answer.

From my experience at Teknique it has become clear to me of how effective it is to focus on user and solving real problems. Teknique always look for what is that we are lacking in the world and how can be improve. They never just work on a project for the sake on grinding. This greatly reflect in the result of many of Teknique's project where they will plan with the client to see what is actually useful rather just getting a useless product into the market. From the result it is clear that the clients very appreciate Teknique's conviction. This allow them to retain long ongoing partnership and attract many new clients. This will also be in important observation for me as I will use this to better produce solutions in the future.

A small but not least lesson I learnt is the importance of how relationships are dealt with. These includes developer to developer, developer to client and developer to leads. Teknique work hard to provide a extraordinary culture and environment to work with. The teams are encourage to communicate together whether it is work related or a sport people play during the weekends. Everyone appreciate and use this culture to help produce greater quality works. Although this seems like a small thing but I feel that this is one of the reason why Teknique is able to attract and retain many great talent to contribute towards its successes.

I am grateful for the extensive internship which lasted 14 weeks. This length allows the interns to learn robustly many new ideas in particular in production and embedded system without being rush to finish off projects.