

Design of a White Lab

Component Vending Machine

Design and Build Report of a Component Vending Machine for the Undergraduates of White Lab



Prepared by:
Baden David Morgan
MRGBAD001

Prepared for:
Mr. J. Pead
Department of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronic Engineering

November 14, 2016

Key words: Vending Machine, Embedded Systems, Web Design, Circuit Design, Design Report, Build Report, C, Python, PHP, HTML, MySQL

Plagiarism Declaration

1. I, Baden David Morgan, know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I, Baden David Morgan, have used the IEEE convention for citation and referencing. Each contribution to, and quotation in this report from the work, or works, of other people has been attributed and has been cited and referenced.
3. This report is my, Baden David Morgan's, own work.
4. I, Baden David Morgan, have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Full Name:
Baden David Morgan

Date:
November 14, 2016



Signature

Terms of Reference

Title:

Design of a White Lab Component Vending Machine.

Description:

The UCT component store cannot stay open 24/7 however students would appreciate if they could get access to components on request. Most student requests can be solved by providing a small subset of components. A modular machine may be a solution to late night component queries.

Deliverables:

- Mechanical Design of a component Dispenser for different sizes.
- Electronics capable of counting, logging and dispenses components.
- Create it in a modular fashion to be extendable as student requirements increase.

Skills/Requirements:

Some Mechanical Design, Electronics, Embedded Systems.

Area:

Electronics / Embedded Systems.

Acknowledgments

Throughout this project I was given the opportunity to work on topics I had never delved into before. I was also able to exercise and improve my skills as a practical engineer. This project would not have been possible without help from many of the people I know.

Firstly I like to thank Jason Russel for helping me with some of my knowledge lacking on web development. He helped me make the website functional allowing me to prove that this project could work as an internet connected device.

I would like to thank Brandon Piner for the help he gave me throughout this project. Whether it be him just listening to me for hours and letting me bounce ideas off him, or helping me debug my circuit boards and staying till we found the problem. He has been such a big help as a friend throughout the years and I greatly appreciate the help he has given me for this project and more so, his friendship.

James Roberston my boss deserves a mention as he was the first person I ever worked for. The experience I gained working for him and the opportunity he gave me helped me realize that I was capable of being an engineer, giving me confidence in the work I did even when I didn't. I will always be grateful for the opportunity he gave me.

Berdine Smit who I would like to thank for helping with some of the proof reading of my document.

Phillip Titus, Riyadh Jacobs and Masonwabe Nomaya for the extensive help they provided during the assembly of the enclosure. Without their help The enclosure would not have been in its current state and I greatly appreciate their help and guidance. I would also like to specially thank them for helping me when access to campus was limited and they were still willing to help me. If not for this I would not have been able to complete the enclosure in time.

I would also like to thank Brendon Daniels for the many components he has supplied and the help he has given me over the course of this project.

Second to last I would like to thank Justin Pead my supervisor. The guidance he has provided has been more than pivotal to the success of this project, the success I feel I may not have been achieved had he not been my supervisor. Also over my years at UCT he has helped me with multiple projects and o9i8 very easy to approach when I have had a question. I would also like to thank Justin for treating me more like a peer than a student giving me confidence in what I was doing. This projects success is shared with and accredited to him.

Finally I would like to thank my family. My brother over the years who has been like a best friend to me. "You can choose your friends but you can't choose your family. And I do choose you as a friend" a quote he made at his 21st that I will always remember. His friendship and support has been the most valuable friendship to me over the 23 years I have been alive and I will never take it for granted. And most importantly I would like to thank my parents, without whom my achievements would not have been possible. They have supported me financially for the entirety of my academic career and I can't even begin to thank them enough for this. They have supported me 100% of the way. Including helping with proof reading this report. I could not have asked for more loving and caring parents. To all the support and love they have given me over the many years I thank them the most.

Abstract

The White Lab was a hub for the UCT undergraduate to study and work on their project and assignments. To allow them to do so certain services had been supplied to assist them. One of these services was the White Lab Component store which operates during the day. This store allowed students to acquire components during the day. However, at night they had no options to acquire components. The white Lab Component Vending Machine was to solve this problem.

The research began by reviewing work similar to that of this project. The similar work focusing on medication dispensers reviewing three in total to gain an understanding of the mechanical methods used and draw inspiration for the designs of this report. Reviews into topics pertaining to aspects of the project, like the communication standards needed and the RFID module to be used for the Component Vending Machine.

Multiple design aspects were approached for the Component Vending Machine to enable a ground up design that could facilitate the White Lab Component Vending Machine. Delivery mechanisms were designed to dispense DIP package components and DIP IC holder. A PCB, called the MCU module, was designed in order to control the delivery mechanism. A PCB was designed in order to interface with a Raspberry Pi and enable it to control the MCU modules and connect to a PSU. An enclosure was designed that would be able to fit all the delivery mechanisms and electronics that made up the Component Vending Machine. Additional designs were done for aesthetics to enable the Component Vending Machine to be user friendly.

A LAMP server was installed on the Raspberry Pi and a custom website was installed that would enable students to order components online before collecting them at the Component Vending Machine. They collected them by using their student card. The enclosure was assembled by welding the frame and bolting the panels onto the frame. The drop plate was bent and attached to the enclosure. Six delivery mechanisms were built, one for each specific design in this report. Six MCU modules were assembled and connected to the a delivery mechanism each. The PCB to interface with the Raspberry Pi was assembled. Finally all the assembled pieces were put into the enclosure and connected to enable them to communicate with one another.

Tests were done in order to make sure the Component Vending Machine was reliable enough to meet the requirements. Upon testing it was found that delivery mechanisms performed better than expected and surpassed the reliability requirements. It was found that the Component Vending Machine was also able to communicate with a website for online orders and able to reliably dispense orders that would be made on the online store.

Although the Component Vending Machine was a success improvements were recommended. The enclosure frame was not rigid enough and needed to be strengthened. The RFID antenna had a poor range or the aluminium panel was absorbing the signal and a redesign of the RFID scanning system was needed to make it more reliable. Due to the scope of the project the variety of the Component Vending Machine was limited to DIP packages. Further research and testing should be done to find solutions for other packages. The project was recommended to continue and advance with manual and instruction on how to assemble more modules to add to the existing stock of the Component Vending Machine.

Contents

1	Introduction	1
1.1	Subject and motivation for the Research	1
1.2	Objectives of this Research	1
1.2.1	Problems to be investigated	1
1.2.2	Purpose of this Research	2
1.3	Scope and Limitations of the Research	2
1.4	Plan of Development	3
1.5	Note for Reader	3
2	Literature Review	4
2.1	Similar Work	4
2.1.1	Portland State University Vending Machine	4
2.1.2	Medication Dispenser, University of Tasmania	5
2.1.3	Medication Dispenser, Narcotic Rehabilitation	6
2.1.4	3D Adjustable cavity Medication Dispenser	7
2.2	Communication between devices	8
2.2.1	I2C	9
2.2.2	RS-485/TIA-485	9
2.3	RFID Reader	10
2.3.1	System Specification	10
2.3.2	UCT's RFID Solution	10
2.4	Website, Hosting and Server	11
2.4.1	The Web server: Apache vs. Nginx	11
2.4.2	The Scripting Language: PHP	12
2.4.3	The Database: MySQL	12
2.5	ATX Specification	13
3	System Specifications	14

3.1	Mechanical Specifications	14
3.1.1	DIP Specifications	14
3.1.2	Delivery Mechanism	15
3.1.3	Enclosure	15
3.2	PCB Specifications	15
3.2.1	Raspberry Pi	16
3.2.2	MCU Modules	16
3.3	Software Specifications	16
3.3.1	Master Program for the Raspberry Pi	16
3.3.2	MCU Module	16
3.3.3	Website	17
4	Design and Prototyping Methodology and Procedure	18
4.1	Design	18
4.1.1	Mechanical Design Methodology	18
4.1.2	Circuit Design Methodology	19
4.1.3	Software Design Methodology	19
4.2	Prototyping Methodology and Procedure	19
5	Component and Material Selection	20
5.1	Communications bus	20
5.2	Servo Motor	20
5.3	Stepper motor and Driver	20
5.4	Power Supply	21
5.5	Microcontroller	21
5.6	Delivery Mechanism Housing	21
5.7	Enclosure Frame	21
5.8	Enclosure Casing	21
5.9	Enclosure Shelving	22
6	System Design and Prototyping	23

6.1	Guide For Component Tubes	23
6.1.1	IC Guide Version 1	23
6.1.2	IC Guide Version 2	24
6.1.3	IC Guide final	24
6.2	Delivery Mechanism	25
6.2.1	Vertical Roller Version 1	25
6.2.2	Prototyping Vertical Roller Version 1	26
6.2.3	Vertical Roller Version 2	27
6.2.4	Prototyping Vertical Roller Version 2	27
6.2.5	Horizontal Roller Version 1	28
6.2.6	Prototyping Horizontal Roller Version 1	29
6.2.7	Horizontal Roller Final Version	30
6.2.8	Untested design: Feed-Up	32
6.3	Enclosure	32
6.4	Enclosure Redesign	34
6.5	RFID Coil Cover	34
6.6	PCB Design	35
6.6.1	Raspberry Pi HAT	36
6.6.2	MCU Module	36
6.6.3	LCD Breakout Board	37
6.7	Software Design	38
6.7.1	Software overview	38
6.7.2	API	39
6.7.3	Web design	40
6.7.4	Raspberry Pi Master Program	41
6.7.5	Microntroller Software	42
7	System Assembly	46
7.1	Software Assembly: Server	46
7.2	Enclosure Assembly	46

7.3	PCB Assembly	48
7.3.1	Pi HAT	48
7.3.2	MCU Module	49
7.3.3	LCD Breakout Board	51
7.4	Delivery Mechanism Assembly	51
7.5	Overall Assembly	53
7.6	Additional Notes and Final Assembly	55
8	Results and Discussion	57
8.1	Prototyping Tests	57
8.2	Comparison test	58
8.3	Final Build Stress Tests	58
8.4	Communications testing	59
8.5	Final build Results and Discussion	60
9	Conclusion	61
10	Recommendations	62
A	Research Mind Map	67
B	Detailed Drawings	68
B.1	IC Guide DIP Components	68
B.2	IC Guide DIP Holder	69
B.3	Vertical Roller & Housing Version 1 & 2	70
B.4	Horizontal Roller & Housing Version 1	71
B.5	Horizontal Roller & Housing Final Version	72
B.6	Horizontal Roller Final Version	73
B.7	RFID Coil Cover	74
C	PCB Diagrams	75
C.1	PI HAT Diagram	75
C.2	MCU Module Diagram	76

C.3 LCD Breakout Diagram	77
D Material Request Forms	78
E Website	85
E.1 Components Store Page	85
E.2 Contacts Page	87
F Templates & Drawings for Assembly	88
F.1 DIP 8 Template	88
F.2 DIP 14 or 16 Template	89
F.3 DIP 20 Template	90
F.4 Drill Template	91
F.5 Exploded Enclosure View	92
G Master Program Log	93
H Manuals and Guides	94
H.1 Programming Guide	94
H.2 Restocking and Maintenance Guide	95
I Terminal Commands	97
J Ethics Forms	99

List of Figures

2.1	The Portlands State University's component Vending Machine and how components were presented [5].	5
2.2	Proposed design of medication dispenser from the group at UT [6].	5
2.3	Dispensing mechanism for Medication dispenser for narcotic rehabilitation with cover removed [7].	6
2.4	Picture of an (a) isometric and (b) cross sectional view of the tablet bucket and its different compartments. [8].	7
2.5	Picture of the 3DAC with the rods (a) unextended and (b) extended. [8].	8
2.6	An illustration of the dispensary action of the 3DAC. [8].	8
2.7	External Low Pass Noise Filter Circuit[9]	9
2.8	When idle one can see the line is held high, this is seen before data is sent and after [11].	10
2.9	The RDM 6300 Module with an external antenna connected directly to the module [16].	11
2.10	Graph showing active websites and their backend web server being used [20].	12
2.11	ATX 20 pin connection layout [25].	13
3.1	Mind map used to help assist the definition of the topic, system specifications and design of the Vending Machine.	14
3.2	Two tubes of components, the top full of DIP IC Holders and the bottom one being full of DIP 8 Components.	15
6.1	IC Guide (a) detailed diagram and (b) isometric view.	23
6.2	Comparison between Version 1 and Version 2 of the IC Guide.	24
6.3	Comparison between Version 1 and Version 2 of the IC Guide.	24
6.4	Rectangular IC guide for DIP IC holder tubes.	25
6.5	Vertical Roller and Housing Delivery mechanism.	25
6.6	Vertical Roller (a) cross section (b) isometric view (c) view to help explain loading.	26
6.7	Prototype of Delivery mechanism using the Vertical roller.	27
6.8	Common jam action that occurs with vertical roller.	27
6.9	Changes made to the Vertical Roller for version 2.	28
6.10	Horizontal Roller and Housing Delivery mechanism.	28
6.11	Labelled isometric view of the Horizontal roller.	29

6.12	Picture of the prototype Delivery mechanism with the horizontal roller.	30
6.13	Final design of the horizontal roller and housing.	30
6.14	Comparison between version 1 and final version of horizontal roller	31
6.15	All rollers to dispense DIP 8, 14 or 16 and 20 for comparison.	31
6.16	Feed up design with cutaway and top view to illustrate how it operates.	32
6.17	Frame of the enclosure illustrating the upper and lower compartments.	33
6.18	A detailed drawing of the sloped plate.	33
6.19	Isometric and top view of the enclosure with the case on interior.	34
6.20	Isometric view of the redesign.	35
6.21	Isometric back and front views of the RFID coil cover.	35
6.22	PCB layout of the Raspberry Pi HAT, top and bottom layer.	36
6.23	PCB layout of the MCU Module, top and bottom layer.	37
6.24	LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlaid.	37
6.25	LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlaid.	38
6.26	LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlaid.	43
6.27	State Machine Diagram for MCU software.	44
6.28	A simplified logic diagram of the task manager and how it will function.	45
7.1	Pictures of the enclosure frame with the walls bolted on.	47
7.2	Pictures of the enclosure frame with all the internal elements in place.	47
7.3	Pictures of the PiHAT PCB by itself, on the left, on the Raspberry Pi, on the right.	49
7.4	Pictures of the MCU Module PCB, with the LCD plugin on the left, without the pcb plugin on the right.	50
7.5	Pictures from the etching process to make the LCD breakout baord.	51
7.6	Picture of the 3D printed IC guide and roller after assembly.	52
7.7	Picture of the housing structure for the delivery mechanism.	52
7.8	Picture of the sensor tester circuit schematic on the left and the physical circuit on the right.	53
7.9	Picture from the rear of the shelf, top image, and the front, bottom image.	54
7.10	Picture of the back and front of the RFID antenna cover.	54
7.11	Picture of the finished Component Vending Machine from the back, front and with the door open.	55

7.12	Labelled picture of the interior of the vending machine and its components.	56
7.13	Labelled picture of the Raspberry Pi and PSU situated behind the sloped plate and below the wooden shelf.	56
8.1	Graph showing the results of the prototyping tests done during the design phase of the project.	57
8.2	Graph showing the results of the prototyping tests done to compare the reliability rating with and without a vibration motor.	58
8.3	Graph showing the results of the stress tests done on all the assembled delivery mechanism for the final build.	59
8.4	Common mode noise observed on the transmission lines.	59

List of Tables

3.1	DIP package dimension for various components measured in millimeter.	15
6.1	API used to communicate from master to slave.	40
6.2	API used to communicate from slave to master.	41
6.3	Database description of variables and what they are for.	42
7.1	Bill of materials for the Server.	46
7.2	Bill of materials for the enclosure.	48
7.3	Bill of materials for the Pi HAT.	49
7.4	Bill of materials for the MCU module.	50
7.5	Bill of materials for the LCD breakout board.	51
7.6	Estimated Bill of Materials for the Delivery mechanism.	53

Nomenclature

ABS	Acrylonitrile Butadiene Styrene - A common plastic used in 3D printing
ADC	Analog to Digital Converter - A hardware component that enables a digital device to be able to sample analog signals
API	Application Programming Interface - A method used to create a system specific communications specification
ASCII	American Standard Code for Information Interchange - A standard set out to relate numbers to symbols and letters for use in computer science and engineering
DNS	Domain Name System - A server type that enables a network to assign URL names to a specific IP
ECE	Electrical & Computer Engineering - A type of engineering
HAT	Hardware Attached onTop - A common name given to a piece of hardware that interfaces with a Raspberry Pi
HTML	Hypertext Markup Language - The language used to generate static content on a website
HTTP	Hypertext Transfer Protocol - The protocol used to communicate over the internet and make and receive requests
I2C	Inter-integrated Circuit - A hardware level communications protocol using a clock
IC	Integrated Circuit - A circuit that was integrated into a small package, like the DIP package
IP	Internet Protocol - The protocol used to identify servers on the internet
MCU	Microntroller Unit - A digital logic device that is programmable
PCB	Printed Circuit Board - A board that has a conductive metal printed on it to make electronic products
PHP	Hypertext Preprocessor - A language used to help make dynamic content on websites
PLA	Polylactic Acid - A common plastic used in 3D printing
RFID	Radio Frequency Identification - A protocol using near field communications technology
RX	Receiver Number x - Relates to the receive pin of a communications protocol, typically UART
TX	Transmitter number x - Relates to the transmit pin of a communications protocol, typically UART
UART	Universal Asynchronous Receiver/Transmitter - A hardware level communication protocol without a clock
URL	Uniform Resource Locator - A name relating to an address on the internet given by a DNS

1 Introduction

1.1 Subject and motivation for the Research

The White Lab is a hub for the electrical engineering students of UCT to interact with one another and work on assignments and projects. The White Lab has many facilities to help students with these assignments and projects including supplies such as wiring, veroboard and components. The White Lab can be very busy during the day and at night. On a busy day, students can be seen lining up to collect components from the White Lab component store or be seen waiting for it to re-open. At night however, they do not have the luxury of the component store, besides a selection of resistors on the wall outside the component store. Not being able to acquire components from the White Lab component store during specific times can cause delays in students' assignments and projects, especially, overnight. The proposed solution to this problem is the White Lab Component Vending Machine. This Component Vending Machine would be operational 24 hours a day 7 days a week, with an online store for ordering components and the physical Component Vending Machine where students can collect components.

The components to be dispensed are those most requested by students. A list of commonly ordered components was compiled, prior to the investigation of this report, by the head of the White Lab Component Store. The most common components observed from this list were seen to be in packages such as DIP and TO packages. This information was compiled by the Principal Technical Officer of White Lab, Justin Pead, and the manager of the White Lab Component Store, Brendon Daniels. These packages brought about the hypothesis that a mechanism for dispensing individual components could be designed to cater for specific packages. The need for lines at the component store would be reduced by introducing the Component Vending Machine and allow students to acquire components at night when the Component store would be closed. These component packages were universal, meaning a designs to cater for these packages could be a possible solution. It also meant that if a design for a package could be achieved, multiple different components could be dispensed.

This report documented the research, design, build process and assembly of the White lab Component Vending Machine.

1.2 Objectives of this Research

The main objectives of this report were to investigate the mechanical and software design of a Component Vending Machine. The Component Vending Machine needed to:

1. reliably dispense individual components on a regular basis.
2. be internet controllable for remote component ordering and control.
3. be self contained and user friendly.
4. stock components in a manner that they could be dispensed and restocked easily.

1.2.1 Problems to be investigated

The above mentioned basic requirements of the Component Vending Machine could be formalized and investigation problems formed, as listed below:

- **Internet Viability:** Investigate the possibility of connecting such a device to the internet and what software and hardware would be required. A test case would be examined and constructed to prove the possibility of an internet connected Component Vending Machine.
- **Mechanical Delivery Mechanism:** One of the main focuses of this research was to investigate designs of previous projects. Once information on previous projects was acquired, a solution would be designed to meet the requirements for single component delivery mechanisms. These delivery mechanisms had to be modular to promote expansion.
- **Enclosure design:** An enclosure would be needed in order to house the delivery mechanism and stock of components. The enclosure had to be big enough to house the planned delivery mechanism to be built for this report and future delivery mechanism
- **PCB and Circuit design:** PCBs were needed to control each delivery mechanism. This had to be designed to meet the specification of the designed delivery mechanism. Challenges that were to be investigated are communication due to a noisy environment and microcontroller selection and designing a circuit around it. The circuit had to accommodate a modular design for future expansion.
- **Software requirements:** On top of the software needed to facilitate an online connection, software would be needed in order to control the delivery mechanisms and the Component Vending Machine itself. This required some embedded software design and software development.
- **Assembly:** The assembly of the Component Vending Machine would be documented in order to find problems and assess the viability of the design proposed in this report. Each element of the design would be addressed individually.

1.2.2 Purpose of this Research

The purpose of this research and report was to assess the overall viability of a 24 hour Component Vending Machine and all the components that made it. The design and research was to help current and future UCT students by giving them an extra method of acquiring components in the White Lab and extending the times they may do so. It was intended that this design be an in place solution for this problem or a prototype that could be used to build a more commercial product. The Aim of this research was not to replace the White Lab Components Store, but rather an extension of it.

1.3 Scope and Limitations of the Research

The scope of this project was defined only by the components supplied by the White Lab but was narrowed down, in order to focus on the overall objectives. This meant compromises needed to be made. The scope of the project defined that an entire Component Vending Machine be designed from the ground up. The scope of the internet connection was limited to a test case of specifically designed website to prove the viability of having the Component Vending Machine operate in conjunction with a web based store. The current component store stocks a plethora of components that student can request. It would not be viable nor effective to design for all cases, as not all components were requested as often as others. The scope of the design was limited to DIP package components, those being the most commonly requested components. This information was obtained from the current White Lab Component Store. It was intended that recommendations be given on other component packages but no implementations will be specifically discussed.

1.4 Plan of Development

Initially, review and research was done on peers' work that was similar to the topic of this report. It was reviewed in order to gain insight into how they approached the problem and to find possible problems and ways to improve on their attempts. This was not to copy from their work, but to take inspiration from their work. Research into concepts and theories that would help gain an understanding on certain aspects of the research were reviewed next. This included reviews of the hardware and software to be used. These reviews are presented in the first section.

A breakdown of the system specifications was layed out in order to gain a better understanding of the research in the next section. Thereafter, a methodology was layed out in order to help understand how the research was approached.

The component selection was presented after the methodology. The components for all the designs were selected and the reason for selecting the aforementioned components were reviewed and presented in the next section.

The next section covered the design of the research project. It covers the design of all the components that made up the research. This was done by presenting each version of the different designs that were made during the course of the research and explaining why they were discarded or used.

The following section will cover the assembly of the project the research is on. The assembly draws reference from the design to help explain how it was assembled. The assembly concludes with an overview and presentation of the final product.

The next section covers the testing that was done at various stages of the research. Each test is first presented then a discussion and remarks are made about the results of the tests.

The conclusions comes next, finalizing the design and giving a verdict on the research project based on the results from the previous section.

Finally, recommendations are presented for improvements that could be made to the research, design and assembly. The recommendations cover many issues and possible implementations all gained from experience throughout the course of the research.

Additionally, appendices are attached to give further insight into the research.

1.5 Note for Reader

If this document is being read on a computer or a digital device that supports the pdf format, take note that citations, cross-references and links are all clickable. This is to make navigating this report easier and more enjoyable to read on electronic devices.

2 Literature Review

Investigations into work similar about the topic of this report i.e., dispensers of small objects, were done in order to gain more insight into the report and better understand how a design of this calibre should operate. The reviews were done so that similar mistakes could be mitigated and to possibly improve on existing ideas. A literature review of various works will be presented in this section. It will start with work and designs similar to that of this project. After similar work has been reviewed and presented, documentation and literature on fundamental aspects of the project will be reviewed. This is in order to gain specific insight into how to customize the design in order to adhere to the requirements of this project effectively.

The investigations started off by looking at multiple sources. The list of possible sources was narrowed down to 10 in total as those 10 were the most relevant to the research of this project. After further reading it was found that not all the sources were thorough enough or outdated. M3DITRACK3 [1] was reviewed but later discarded as the design documented in the report was insufficient. The Multipurpose Robotic Dispenser [2] was also discarded as its design report was insufficient. After a quick read of the smartphone-enhanced pill-dispenser [3] report it was found that the main focus was on software and so this was also discarded. Additionally patents were looked at but were either out of date or better designs had already been reviewed. This lead to the four sources that were reviewed below.

2.1 Similar Work

Similar work to the topic of this report will be presented in this section. It was difficult to find work specifically related, so for the most part, medical dispensers were considered a close comparison. This comparison was chosen as medication comes in a small package and can also be specific to each package, where one unit of medication was dispensed at a time. This drew parallels to the White Lab Component Vending Machine requirements and so work in this field was reviewed, but only the design aspects of these works.

2.1.1 Portland State University Vending Machine

At Portland State University under the ECE department, a vending machine was retrofitted to provide a wide range of components, 24/7. They employed a method of using zip lock bags and placing a package of components in each bag. These bags were then put in a standard snack vending machine like the ones commonly seen at stores and on campuses, like UCT. In order to restock the machine, volunteers packaged these bags with various components. A picture of what was inside the vending machine can be seen at Figure 2.1 [4, 5].

A major advantage of Portland State University's solution was that it was not component specific. Since a vending machine could have multiple racks with multiple rails on each rack, the capacity of components could be very large. Although this was a good solution to the problem and up scaling an old vending machine was a good use of resources, this solution has many drawbacks which are presented below.

The loading solution was to have multiple volunteers put packages together and manually load each rail of the vending machine. This required multiple individuals input to reload a rail which could delay restocking the machine. Since packages were pre-packed, students do not have a choice of smaller quantities. Another issue was that it being a vending machine , even though it was a paid model, the



Figure 2.1: The Portlands State University's component vending machine and how components were presented [5].

access was not regulated . That meant that one person could empty a rail in one day if they chose to do so. Overall this was a good solution but a more targeted solution would be required for the UCT's Component Vending Machine.

2.1.2 Medication Dispenser, University of Tasmania

A group from University of Tasmania [6] proposed a solution for medication dispensers that would be installed at a patients house. The device was to be connected to the internet and the dispensing action would be initiated by the physician overseeing the patient's whose home the device was in. Figure 2.2 Illustrates how the device would operate mechanically.

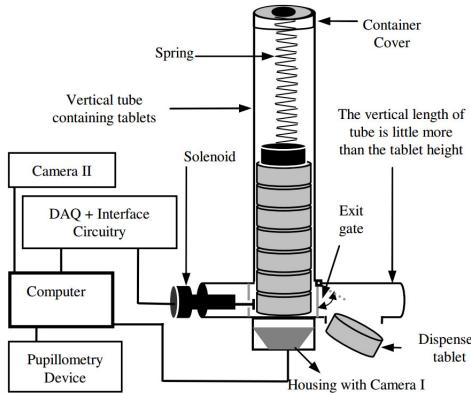


Figure 2.2: Proposed design of medication dispenser from the group at UT [6].

The dispenser was fed by a gravity, spring assisted, magazine of medication, which was labelled “vertical tube containing tablets” in Figure 2.2. Although having a gravity fed system allows for a larger number of tablets in the magazine, the spring hindered this capability. The spring would limit the number of tablets by the compression force of the spring i.e.; the force imparted on the units by the spring that would crush the tablets. It would also be limited by the size of the spring causing a further design choice not shown in the report; length of the spring vs the spring constant. Getting the spring constant and length right for this loading mechanism would greatly affect the reliability of the magazine feed and determine how many tablets would be able to fit into a single load of the magazine.

A solenoid was placed at the bottom of the vertical tube to dispense the tablets and was actuated

to dispense one tablet. The solenoid was actuated via an “interface circuitry” which in turn was controlled by a DAQ receiving commands via an internet connection. This was similar to the design requirements to the White Lab Component Vending Machine. A physician controlled when a tablet would be dispensed. One issue with the device was that it did not have any sensors depicted that would detect a jam or empty magazine. Although a camera was shown this was so that the physician could get visuals of the patient for whom the tablet was being dispensed to.

Although the device did not have any sensors for jamming or empty magazine detection, having it connected to the internet allowed for logging of the device to check it was operating reliably. The physician could keep track and moderate dosages as they saw fit. This seemed ideal in a medical application. However, this would not work for a component dispenser. A useful idea from this medication dispenser was using a gravity fed system allowing reduced moving parts and potentially greater stock, reducing the need for constant restocking. Having an internet connected device fell under the same requirements of the Component Vending Machine. The solution the University of Tasmania came up with was to compartmentalize the internet connection and the circuitry. This compartmentalization idea between the internet connection and embedded electronics could be used in the Component Vending Machine.

2.1.3 Medication Dispenser, Narcotic Rehabilitation

The second medication dispenser reviewed was one intended for narcotic rehabilitation [7]. It had many features similar to those needed for the White Lab Component Vending Machine including single tablet dispensing and control via the internet.

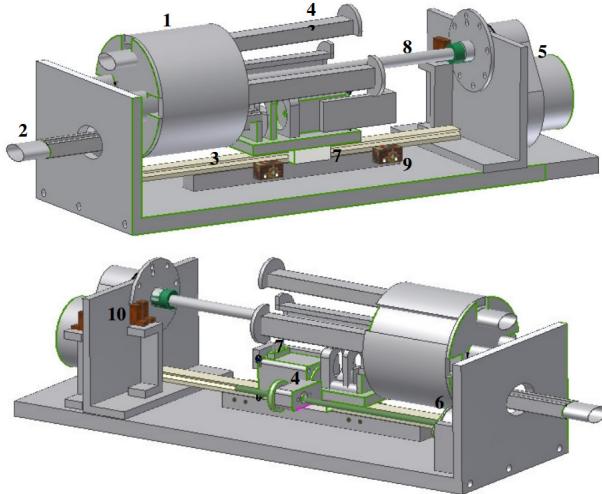


Figure 2.3: Dispensing mechanism for Medication dispenser for narcotic rehabilitation with cover removed [7].

The medication dispenser started off with a few specifications ranging from a tamper-proof housing to remote control via the internet, those being related to the design of the Component Vending Machine. Depicted in Figure 2.3, the mechanism was driven by a stepper-motor , (5), connected to a cylinder, (1), by a shaft at (8), with the dosages attached to the cylinder, (4). As the dosage rotated to the bottom position another stepper motor would engage a mechanical drive, (7), to push the dosage rod out as shown at (2) and retract the rod. Limit switches, at (9), were used to determine the position of the bottom rod and optical sensor was used to determine the amount of rotation of the cylinder at (10). It measured the rotations by using a slotted disk and shining a light through the slots and counting the times the light was blocked and re-appeared. The mechanism would then be situated in a strong housing to prevent tampering, and if tampering did occur, it would be obvious due to

damage needed to open the housing. This dispensary mechanism was controlled by a microcontroller communicating with a computer, which in turn, was connected to the internet.

The cylinder provided a reliable delivery mechanism but limited the amount of stock of the tablets that could be loaded in the dispenser. This could be a design choice by the people who designed it as patients being rehabilitated should not have access to medication in large amounts even if it was protected. This reliable method of using a cylinder or a wheel could be advantageous as the design rotated with all its parts making it difficult for a jam to occur from clashing of parts. The only danger would be if the dispensary rod was extended and the cylinder rotated. This was avoided by using sensors to detect the extension of the rod. This rationale of preventing the device from causing damage to itself by effectively using sensors and programming should be implemented in the Component Vending Machine.

Another interesting approach used for the programming was the compartmentalization of tasks. The microcontroller acted on its own with all the programming to do a complete dispensary action if needed but, received commands to do so by a computer connected to the internet. This method made each individual device act reliably, like the microcontroller and the computer which enabled the system as a whole, to be more reliable. This kind of programming style should be used for the Component Vending Machine to make it work effectively.

2.1.4 3D Adjustable cavity Medication Dispenser

The 3D adjustable cavity Medication Dispenser was aimed to solve a few problems observed in previous attempts by other parties. These problems were; the density of tablets stored in the mechanism, the variety of tablets and the adaptability of the mechanism. It aimed to do this with its 3D adjustable cavity solution, “3DAC” for short [8]. The 3DAC system was also intended to be a home dispensary system meaning it had to be compact for a home space.

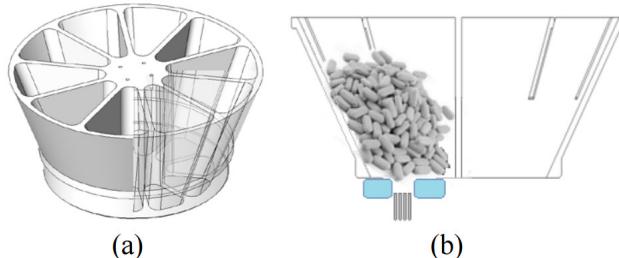


Figure 2.4: Picture of an (a) isometric and (b) cross sectional view of the tablet bucket and its different compartments. [8].

The problem of the density of the tablets and the variety was solved with a bucket called the “Hopper” which could be seen in Figure 2.4. The Hopper consisted of 10 compartments enabling it to simultaneously hold 10 different types of medication at any one time. The individual compartments were also sized to fit a large number of tablets solving the problem of the density of the tablets. The hopper would rotate with an opening on the bottom for a tablet to fall through into the 3DAC.

The 3DAC was theorized, in order to solve the problem of having an adaptable delivery mechanism able to dispense multiple shapes of tablets. It was designed based on models of common shapes of tablets which the team behind the 3DAC simplified into a rectangular shape of varying size for each, which would encapsulate the whole tablet. This was done to explain the choice of a rectangular cavity for the 3DAC.

The 3DAC illustrated in Figure 2.5 shows it in two configurations. The 3DAC was designed to have 3 adjustable walls that could move to make the right shape for any specific shape of a tablet.

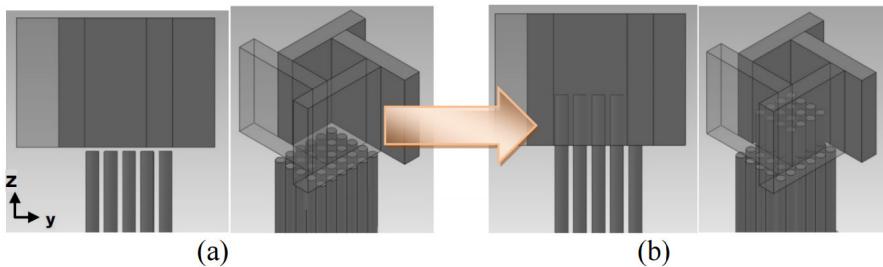


Figure 2.5: Picture of the 3DAC with the rods (a) unextended and (b) extended. [8].

Compressible rods were used to form the base of the 3DAC which would enable the depth of the cavity to be adjusted to the specific height of the tablet to be dispensed. This made it possible for the 3DAC to dispense a wide range of tablet shapes without having to manually change the delivery mechanism and the specific alignment of the cavity could be coded into the device. This coding would make the adjustment automatic and improve the performance of the delivery mechanism, making the 3DAC able to dispense tablets from all 10 compartments of the hopper regardless of shape.

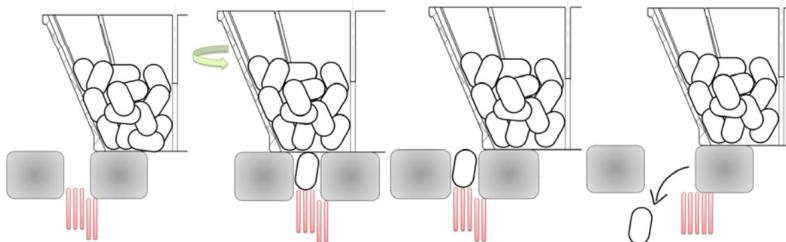


Figure 2.6: An illustration of the dispensary action of the 3DAC. [8].

The dispensary action of the 3DAC has been illustrated in Figure 2.6. As can be seen, the 3DAC would position itself under the hopper with the cavity adjusted to the requirements of the tablets in that section of the hopper. The tablet would then fall into the cavity and the 3DAC would move away from the hopper. One of the walls of the 3DAC would then give way and the compressible rods would retract to assist ejecting the tablet. During the testing this method was proven to be effective at expelling a tablet 95% [8] of the time, giving evidence that this was a potentially successful design.

This solution to the dispensary mechanism was very adaptable but complex. The complexity might not apply to the Component Vending Machine but the ideology of a simple design that could adapt to different requirements that could be applied. This was not to say an automatic cavity should be used as that would take too much time to develop by itself and, an electronic component with pins which could easily get caught on the edges of the cavity or other components was more complex than a tablet. The hopper was also not an ideal solution for the Component Vending Machine because of the complexity of an electronic component compared to a tablet. However, having a gravity fed system such as the hopper was definitely the solution that was needed for the Component Vending Machine.

2.2 Communication between devices

Since it was decided to have each delivery mechanism act independently from one another and potentially a master device, a means of communicating between each device or from master to slave was needed.

2.2.1 I2C

I2C was a communications protocol that was easy to use with most micro-controllers having built in hardware to deal with I2C. The main hurdle with using I2C would be failed communication due to noise. Some examples of noises would be switching noise from power supplies or the environment itself [9] and signal generators like the ones present in UCT's White Lab.

One of the ways to improve noise immunity and improve reliability would be to use an external RC filter as suggested in the White Paper from Lattice Semiconductors. One of the hurdles with such a filter on I2C lines was finding a balance between loading and filtering. The higher the time constant ($\tau = RC$) the slower the rising edges of the line and the greater the driving load on the IO of the micro-controller. Below was an image of a recommended filter from the White Paper with a good balance between loading and filtering. The values set at $R_{pullup} = 1800\Omega$, $R_{s1} = 130\Omega$, $R_{s2} = 51\Omega$, $C_f = 180Pf$ [9].

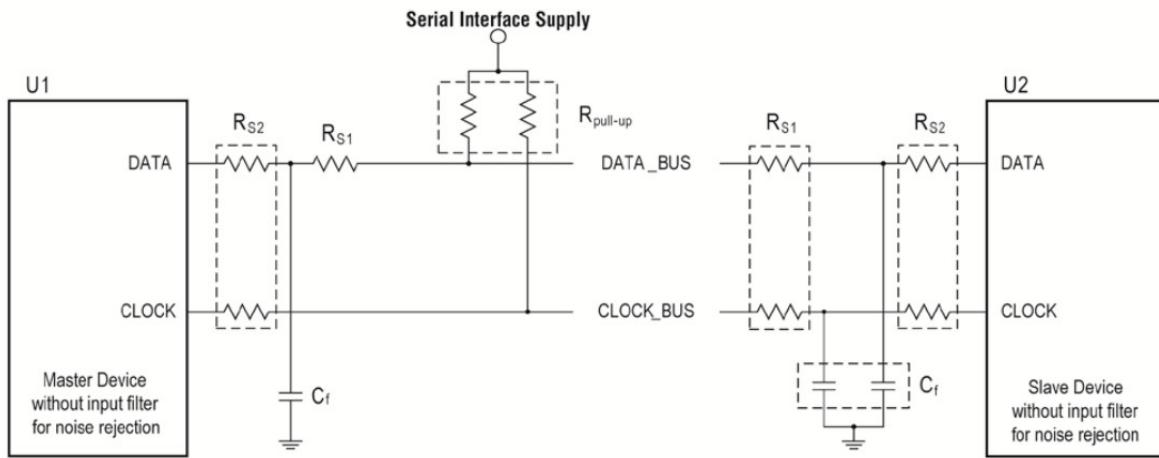


Figure 2.7: External Low Pass Noise Filter Circuit[9]

2.2.2 RS-485/TIA-485

RS485 was considered as it was an industry standard in factories with electrical equipment needing to communicate with one another. This was because it could operate over long distances (up to 120 meters at 100kbps [10]), and in noisy environments. This was ideal for the Component Vending Machine as it would most likely be placed in the White Lab at UCT where there were multiple devices capable of interfering with the communications bus and potentially corrupting the data as discussed in the above section.

The network topology was similar to that of I2C where each device was “daisy chained” to one another making what was called a bus for communications. It was common to use an IC such as a bus transceiver to facilitate the RS485 standard where the hardware does not support it and use UART, with CTS and RTS pins if available [10]. UART cannot be used alone even with shielded cables because both devices on the bus hold their TX lines high as shown in Figure 2.8. This made it impossible for 3 or more devices to be connected on the same lines for UART as corruption would occur.

Although with CTS and RTS pins, one could use the UART in RS-232 mode, however this required two extra pins to run on the bus. Using the RS-485 standard would alleviate this problem as it was designed to work on 2 lines with differential signals when using it in a half-duplex configuration. This gave RS-485 a great advantage against noise immunity as it was not susceptible to all kinds of noise.

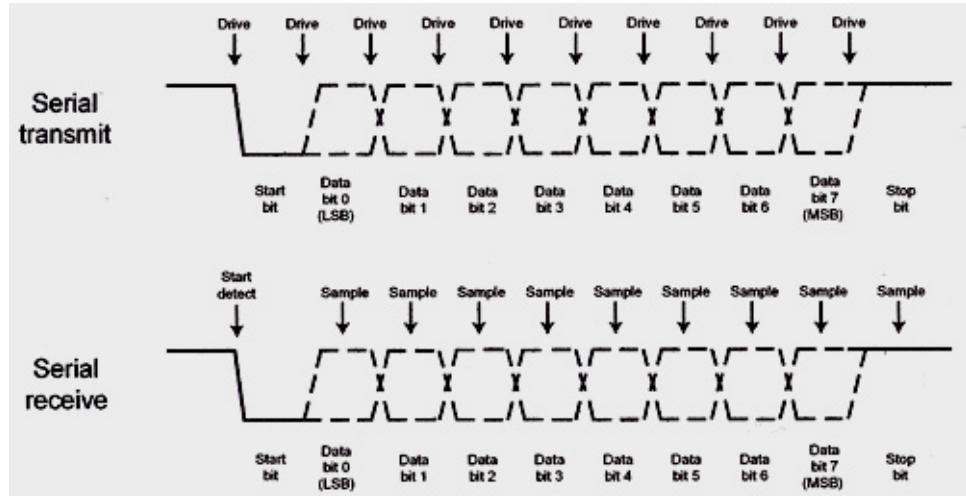


Figure 2.8: When idle one can see the line would be held high, this is seen before data is sent and after [11].

Noise in a system could be split into common-mode and single-ended noise of which RS-485 was immune to common-mode noise unlike standard UART or I2C. This made single-ended noise the only issue. Single-ended comes from improper transmission line termination from mismatched resistance on the output and input transmission line. This can be solved by using a terminating resistor on each node of the bus which matches the resistance of the line impedance. In addition to proper termination, twisted shielded pairs were recommended making the bus less susceptible to interference [12].

2.3 RFID Reader

In order for students to be identified when requesting components from the Component Vending Machine their student cards would be utilized to match them with their order. This meant hardware to identify these student cards was needed. This hardware was reviewed in this section.

2.3.1 System Specification

There were three different classes for the operating frequency of RFID systems (low frequency, high frequency and ultra-high frequency), and 3 classes of device systems related to how they were powered (active, passive and battery assisted passive) [13]. The system that was decided upon that would be used for the Component Vending Machine would be a low frequency system with an RFID Reader and Passive tag. In RFID tags were referred to as the item to be tracked or the identifier in this case the student card. The tag had an on board antenna and a “tag-chip” which contained an ID that could either be factory set, programmable or write-once. The antenna was used to power the tag, by receiving power from the reader when in range, and transmitting data to the reader. The reader also had an antenna which was used to transmit power to the tag and receive the data being transmitted by the tag [14].

2.3.2 UCT's RFID Solution

Since an understanding of the fundamentals was attained, the physical reader was then reviewed. The reader to be used was a solution put together in house at UCT using the existing RDM 6300 module

and MCP2200 FTDI Chip [15].

The RDM 6300 was a 125KHz low frequency card reader for 125KHz compatible tags like the ones used by UCT for student cards. It supported an external antenna with a range of about 50mm [16]. This was a fairly popular module in the maker community as it was relatively cheap, at about \$12.50, compared to commercial equivalents. It was also popular because of the many resources associated with the maker community and its widespread adoption meaning any problems would be easy to troubleshoot through this community if needed. A picture of the above mentioned RDM 6300 with an external antenna attached can be seen in Figure 2.9.

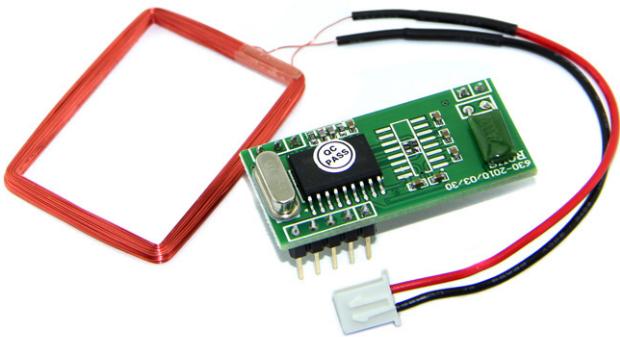


Figure 2.9: The RDM 6300 module with an external antenna connected directly to the module [16].

The MCP2200 was a USB2.0 to UART Protocol Converter when paired with the RDM module would allow the RFID Reader to communicate with any computer with a USB port [17]. This would allow the RFID reader to send data over the USB cable to the computer whenever a tag was read making it easier to interface with the RFID device. This conversion was needed as the UART interface on the Raspberry Pi was to be used for communications on the RS485 bus.

2.4 Website, Hosting and Server

In order to prove that the Component Vending Machine could operate with a website, an interactive and functional website would be needed. The model used for the website would be a LAMP archetypal structure, or Stack, as this was a very popular implementation of a web server. LAMP stands for Linux, the operating system, Apache, the webserver, Mysql, the database server and PHP, the scripting language used, an acronym of the software bundle used, all of which were open source [18].

2.4.1 The Web server: Apache vs. Nginx

In order to host a functional website, a web server had to be used. Although a LAMP stack was originally favoured, other web servers were considered. The major alternative was Nginx as it had some benefits over Apache. Its main advantage and selling point was that it could handle more connections concurrently than Apache so for a heavy duty web site Nginx was necessary [19]. Apache was chosen over Nginx however, as Apache was a more popular web server with widespread popularity and because the website required would not have a high client base. Netcraft did a survey every month posting the results at the end of each month. They had been getting responses from over 1 billion sites which gave credibility to the survey. According to the survey, as at the end of August, Apache

had the majority share in active web sites due to its continued support with a market share of about 46% compared to Nginx or Microsoft's 22% and 10% respectively, these were taken from the latest data in Figure 2.10. Although it had shown a steady decline in market share since 2011 it was still the leader in web servers [20]. This meant support and knowledge base would be abundant making the use of Apache easier to troubleshoot over Nginx the primary contender. The graph in Figure 2.10 shows the market share of each web server program over the period of August 2016 to June 2000.

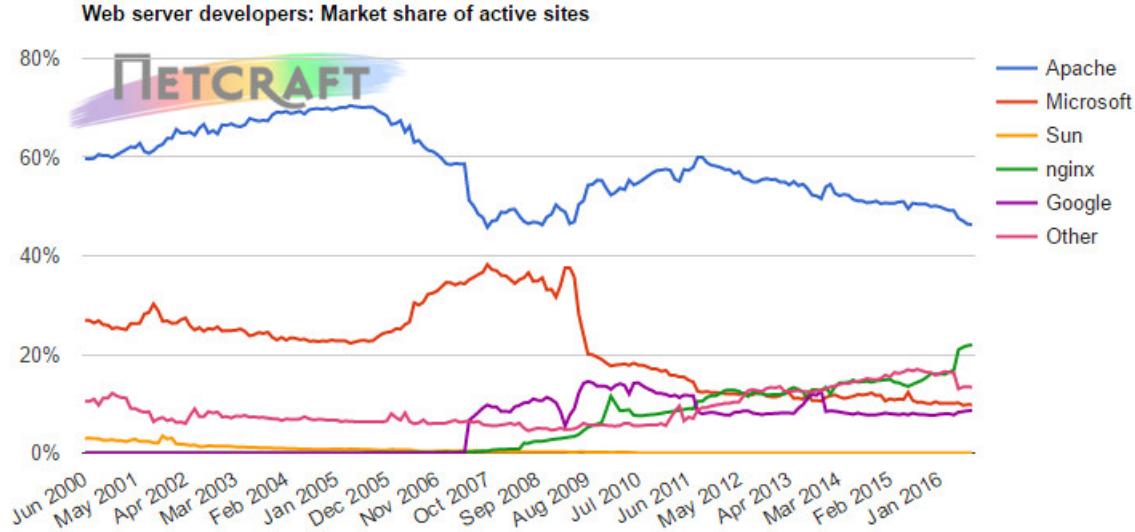


Figure 2.10: Graph showing active websites and their backend web server being used [20].

Apache translated a URL that would be attached to an IP address and fetched files related to that IP address, returning them to the browser or program that the IP request came from. This could also be a program, where the server would execute the program requested and return the output. This was all done through a protocol called HTTP (Hypertext Transfer Protocol) which allowed the browser to make a request in a manner the server would understand it, similar to a protocol used to communicate between micro-controllers. More specifically for Apache, these were files stored on the computer the LAMP stack was installed on and in a directory Apache was directed to [21, 22].

2.4.2 The Scripting Language: PHP

If the site was directed to a PHP file or a file it saw contained PHP syntax it would pass it onto the PHP interpreter. The interpreter would then execute the code on the server side and a result would be returned with a static page like HTML. Although a static page could be served using just HTML, PHP ad the advantage of enabling dynamic content based on server side variables; such as content from a database. Another advantage of PHP was it could be embedded into HTML code. This made it possible to make the UI look great with static HTML and serve embedded dynamic content with PHP. Another big advantage of PHP which helped it gain a large market share of users in its infancy was its ability to interface with multiple database servers. The one that is to be focused on in the report would be MySQL [23].

2.4.3 The Database: MySQL

MySQL was a database platform built to run independently. It managed data by storing in databases with separate tables each with its own rows and columns. It was able to relate certain data from one table to another by using user defined rules making operation and navigation fast when used properly. MySQL also helped protect the databases with proper protection giving access to users defined during

configuration, making it a secure way to store sensitive data. MySQL used the ANSI/ISO standard SQL (Structured Query Language) which enabled easy access when permitted. The SQL standard and access to MySQL worked on a query basis, ie; when an SQL statement was constructed and sent to the MySQL. Once the query had been interpreted, the relevant data pertaining to the request was returned [24].

2.5 ATX Specification

In order to power the Component Vending Machine, a computer power supply was to be used. In order to understand how to use this power supply connection, the standard used was investigated. The standard used for computer power supplies was well documented and open source, which was called the ATX Specification and was made by Intel [25]. Additionally, tutorials were followed on how to setup the ATX connection for use outside of motherboard applications to be thorough. Revision 2.1 of the ATX specification was used instead of the latest 2.2 as the latest version has been updated to only cover ATX2 which was an updated standard adding addition pins for newer motherboards.

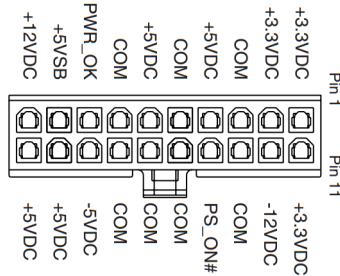


Figure 2.11: ATX 20 pin connection layout [25].

The ATX standard complied with the specifications set out by Intel to operate their processors. This was useful as an ATX compliant power supply had a 12V; 5V and 3V3 supply line capable of supplying large amount of current. The connection layout is illustrated in Figure 2.11. In order to use the power supply specific connections needed to be made and certain power rails had to be used. A kettle plug had to be connected for the AC power to power up the PSU. The AC switch had to be turned on and the “PS_ON#” must be grounded to supply power over the 12V, 5V and 3V3 rails. If left open, the voltage rails would be shut off. A “PWR_OK” output was provided from the power supply that indicated if the voltage rails were supplying the correct voltages with a high signal [25]. This was not necessary for operation but convenient for indication. In order for the power to maintain a constant voltage it may need to be tricked into thinking it was connected to a motherboard. This could be done by connecting a 5W load to the 5V rail [26].

3 System Specifications

In order to begin designing the Component Vending Machine, the specifications needed to be defined in detail in order to know what the designs needed to focus on. There will be three major sections in the design and discussion of this report. Those will be; the mechanical design including the delivery mechanism and enclosure, the PCB design of both controller for the delivery mechanism, which will be called the MCU module from now on, and the Raspberry pi module plugin, and the software for the Master program i.e. the Raspberry Pi, the MCU modules and the website. A mind map was created to assist defining the specifications and can be seen in Figure 3.1. It shows the relevant specifications for each three sections and how they interact.

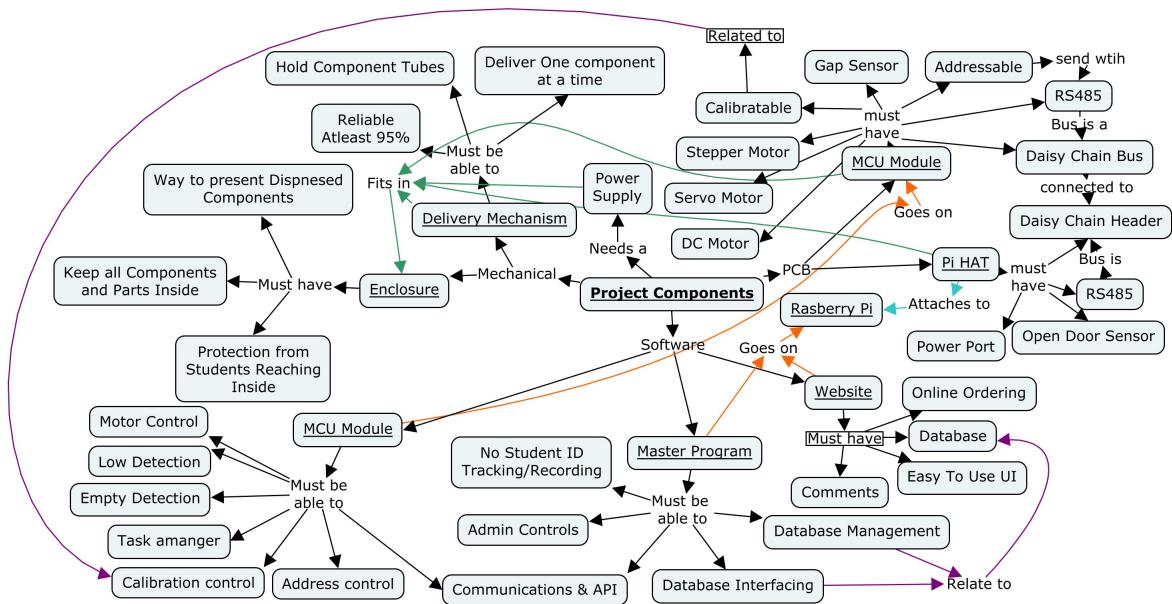


Figure 3.1: Mind map used to help assist the definition of the topic, system specifications and design of the Component Vending Machine.

A mind map is also included in Appendix A Research Mind Map which was created on commencing this research to better define the topic and help decide on a direction to take. This mind map is a precursor to the mind map above but not as relevant to this section.

3.1 Mechanical Specifications

The mechanical specifications and requirements will be discussed in more detail below.

3.1.1 DIP Specifications

The dimensions of various DIP components were measured in order to determine the design requirements for the dispensers. The method of investigation was searching a well-known vendor's site[27] for datasheets on the various DIP components. Datasheets were checked for dimensions until 5 different measurements were obtained, the list is summarized in Table 3.1.

	DIP 8	DIP 14	DIP 16	DIP 20
	9.6	20	19.55	25.73
	10.66	19.5	20.32	26.42
	10.16	20.19	19.5	27.17
	10.82	19.5	21.97	25.4
	10.2	20.32	19.81	24.5
Range	9.07 - 10.66	19.5 - 20.32	19.5 - 21.97	24.5 - 27.17
Average	10.06	19.19	20.23	25.84

Table 3.1: DIP package dimension for various components measured in millimeter.

From the data collected it showed that there were only 3 different packages to cater for, as the DIP 14 and 16 packages were observed to be very similar.

3.1.2 Delivery Mechanism

The main objective of the delivery mechanism needs to be stated as being able to deliver one component at a time in a reliable manner. The delivery mechanism needed to be very reliable with a success rate of 95% or above. It was set high because a failure in the system would mean human intervention to correct the error breaking the autonomous nature the Component Vending Machine was to have. This autonomy was another specification as the Component Vending Machine was to operate at potentially late hours of the night when no authority to amend a failure would be available. To improve autonomy, Component tubes, pictured below, would need to be held in the machine while being fed into the delivery mechanism to give the Component Vending Machine a large capacity.

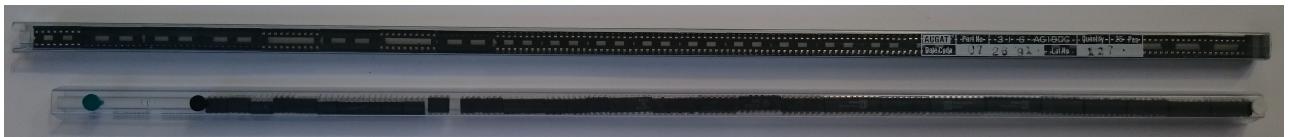


Figure 3.2: Two tubes of components, the top full of DIP IC Holders and the bottom one being full of DIP 8 Components.

3.1.3 Enclosure

The enclosure's priority specification was, to be able to hold all the modules of the Component Vending Machine. The enclosure also needed to be able to have space for expansion for future requirements that may arise after the completion of this research to keep the project as a whole alive. It should also be able to prevent people from sticking their hands into the machine and interfering with the operation of the Component Vending Machine. Lastly, the enclosure should be able to present the dispensed components in a manner easy for the person ordering, to retrieve.

3.2 PCB Specifications

The PCB requirements and specifications will be discussed in more detail below.

3.2.1 Raspberry Pi

The main purpose of the PCB to that would interface with the Raspberry Pi, was to be able to power the Pi and create a starting point for communication bus. This module should also be able to sense if the door of the enclosure was open. This would be to prevent the machine from dispensing components or from moving parts, while someone could potentially have their finger in the mechanism to try and fix a fault.

3.2.2 MCU Modules

The MCU module had many small specifications, contributing to serve the main purpose of being modular and interchangeable for different types of delivery systems that must be required. It would need to operate a DC motor, stepper motor, and servo motor, either on at a time or all at once. Gap sensors should be able to detect an empty load for delivery or a jam, or a low cartridge or a tube. It was later decided that the MCU module should be able to operate an LCD. This was so that messages could be relayed to someone using the Component Vending Machine, i.e., to communicate a problem with the machine. A micro-controller capable of handling the specifications of the MCU module should be coupled with a crystal designed to the microcontrollers specifications was needed. A communication and power bus that could be daisy-chained to additional modules was required for expansion. This was so that more than one module could be attached to the same communication bus. Calibration controls to adjust and fine tune the operation of the device were needed to enable adjustments. This would allow for small imperfections in manufacturing and building to be circumnavigated by changing set limits in the code.

3.3 Software Specifications

The specifications of the Software requirements and specifications will be discussed below.

3.3.1 Master Program for the Raspberry Pi

The main function of the Master program needed be to recognize student cards and the student number attached to them and communicate with the MCU modules. This communication would allow the Master program to control the actions of each MCU module when needed. The Master program must also manage the database and had to make sure the orders were not invalid. Another feature the Master program needed to have was administrative control in order to be able to induce additional administrator features on each MCU module for maintenance purposes. The Master program had to be able not only to interface with a database, to do all the above, but also should not record any user ID's or information taken from a student card.

3.3.2 MCU Module

The specification for the program was set out to make sure the program ran reliably. To ensure this, one of the first requirements was that the MCU would use very few delays and instead use a task manager. The task manager needed to enable the MCU module to operate without delays when dispensing components so that the system would not be held up by any one task. In addition to the task manager, motor control for the 3 kinds of motors had be present. Detection for empty load and

low cartridge/tube using optical gap sensors was required so the module could sense if a component were present in the mechanism or not. An address storage system must be in place so that the device would know when it had been commanded over the communication bus. Finally, the MCU module had to be required to have a task manager to calibrate the device and save the newly set values. A summary of the I/Os needed has been listed below:

- 22 I/O
- 3 Analog Inputs
- 1 PWM Channel
- 1 UART Interface

3.3.3 Website

The website was a proof of concept in order to show the device was capable of interacting with the information saved by a website. The web page needed to be easy to use for users wishing to make orders from the Component Vending Machine. There should be a comments section of some form, so users could give feedback or report problems with the Component Vending Machine. The website should also be able to interface with the database in order to store and read data from it.

4 Design and Prototyping Methodology and Procedure

Prior to beginning the design A clear methodology was needed to proceed in order to get the best results. This included a set of rules to follow when designing and testing prototypes. This section aims to discuss these and elaborate on how the design was approached to meet the requirements set out in the previous section.

4.1 Design

The methodology behind the mechanical design will be reviewed first, then circuit design, software design and finally the prototypes:

4.1.1 Mechanical Design Methodology

In order to make an effective design, certain constraints were first laid out to limit the scope and complexity of the design.

A simple design approach was used to limit the complexity where simplicity and the method of forward thinking; “how would it fail”, were always the first and ongoing design considerations. Once a simple idea was theorized, details were added in order to make it more functional. Simplicity was not the main goal as complexity would be needed in some cases, i.e., where functionality took priority. The number of moving parts would be kept to a minimum to reduce complexity in order to prevent failure of functionality and structure.

Design of the delivery mechanism started out on paper as sketches of basic ideas, until these culminated into a practical. Once ready, the idea was designed in SolidWorks with the above mentioned goals. After the Model was fully defined in SolidWorks, the model was printed on a 3D printer to prototype and the effectiveness of the design was then tested. If the design had flaws, a redesign was done to change and eliminate those flaws and the model was printed again to further test and find any other potential flaws. This process was repeated until a reliable working prototype for the delivery mechanism was produced.

As for the enclosure, a similar process was followed, as was for the delivery mechanism. However, no prototyping was included as the cost for prototyping would have been too high. Another reason for no prototyping for the enclosure, was that the functionality was not as complex as the delivery mechanism. This meant that it was designed with measurements more so than functionality in mind. However, this is not to say functionality did not play a part in decision making. The frame of the enclosure was designed first, followed by the internal housing for the delivery mechanism. The Raspberry pi and power supply was design thereafter. The next step was that the shell was designed along with the slide for the components to fall down and then the front door. Once the design was completed, all the parts were detailed in sketches to finalize the design.

4.1.2 Circuit Design Methodology

The basic idea behind the design of the PCB was to make it versatile and able to adapt to the requirements of the project by adding in features to allow for multiple configurations of mechanical delivery requirements. This required a somewhat modular design.

The circuit started with a sketch on paper detailing what would be needed in the final design and what type of configurations it should be able to handle. Three configurations were considered as the mechanical system needed a motor of some kind. The design needed to include the ability to handle a stepper motor, servo motor and simple DC motor either one at a time or all concurrently. Included in the design was a set of sensors needed to track the status of the delivery and contents of rails. A Raspberry Pi HAT was theorized that would be capable of connecting the power source to the Raspberry Pi and starting the bus for the RS-485 communications and power rails. This HAT would be a fairly simple design to satisfy communication and power supply requirements.

4.1.3 Software Design Methodology

The software for the machine is one of the most important aspect to consider, as it would influence each part of the design and how they would interact.

The software design started with algorithmic state machine diagrams in order to simplify the understanding of the programs and how they would operate. Once an adequate algorithmic state machine diagram was achieved, programming was started. The program was split up into 3 main modules: Website, Master and Delivery Modules.

The website was designed using knowledge learned during the research building up to making the Component Vending Machine. It was designed to be easy for the user to understand and operate, as the UI for the interaction for the students who would eventually use the Component Vending Machine. Although the website was designed to be independent from the Master and Delivery Module code, it was briefly tested with them to confirm its functionality.

The Master and Delivery Modules were similarly and at times, concurrently designed in order to test their compatibility. Both were designed with a modular approach with each small block of code being developed and tested independently before integrating with the main code base. This allowed each small block of code to act on its own without interfering with other blocks of code, making the overall design more reliable. This also helped make debugging easier, speeding up the programming process.

Finally, all three modules were integrated together and tested thoroughly and updated until a working code base was achieved.

4.2 Prototyping Methodology and Procedure

Detailed planning and methodology was needed in order to test the viability of the prototypes for the final build.

In order to test the viability of the mechanical design of the delivery mechanism, a structure for testing and guidelines were drawn up to make sure each test was comparable to the following tests. This was done by making sure the tests were repeatable by eliminating external variables and utilizing a test method that could be used for all test cases. Also, a recording structure was created with data that would be recorded from test to test. Notes were also taken with each test to add contexts and additional information of the success or failure of the tests.

5 Component and Material Selection

This section will cover the components and materials selected for the design of the Component Vending Machine and why they were selected. The selection for the electronics entailed parts for the PCB which would enable the delivery mechanism to operate, the Master board that would control each addressable PCB and the power supply. The material selection ranged from materials used in the delivery mechanism to the metals and plastics needed for the enclosure.

5.1 Communications bus

In order to communicate using an RS485 bus, a differential bus transceiver, was needed to convert the UART signal. The SN75176B was chosen as it had many advantages over and above numerous other differential bus transceivers. It acts as a half-duplex RS485 device when connected to a RS485 bus meaning it can only communicate one way at a time which met the requirements of the communication's requirements. It also meant only two wires would be needed for communication plus a common ground. To manage the half-duplex communications, each transceiver had data enable and receive enable triggers to allow it to communicate and receive data only when needed, allowing the device in control to manage its communications better, i.e. when tasks would be performed and no communication would be expected or needed it could turn off its bus connection. This would allow for more reliable task execution.

5.2 Servo Motor

In order to drive the roller version of the delivery mechanism, an appropriate servo motor was needed. The Hextronik 9g servo was selected, or any 9g servo equivalent (9g stands for the weight of the servo), as it was inexpensive at €2,42 [28], at the date of compiling this report. It was capable of moving the rollers and was light and compact making the overall design compact. Later on though, it was decided to use a stronger and more reliable Corona DS-339 as it did not have overshoot that was observed in testing on the 9g servo. Although it was more expensive at €7,61 [28], it was more readily available at UCT.

5.3 Stepper motor and Driver

Although a stepper motor was not used in any of the final builds, the design catered for a stepper motor for future development. The stepper motor the PCB was designed for, was the NEMA 17. This stepper motor was chosen as it was a fairly commonly used stepper motor and popular in the 3D printing community, RepRap project [29], meaning trouble shooting potential problems would be easier than using another motor.

The stepper motor driver chosen to be compatible with the Component Vending Machine was the A4988 Ramps board. The stepper motor driver was from the same community that popularized the NEMA 17. This meant the compatibility had been tested making the two an ideal pair.

5.4 Power Supply

The power supply chosen was a PSU from a computer. The computer PSU was chosen as it could supply 3.3v, 5v and 12v simultaneously and provided enough power to operate all the electronics concurrently. It was also chosen as it would eliminate the design process of making a power supply and provide a more reliable and effective solution.

5.5 Microcontroller

The choice was narrowed down to two microcontrollers, the STM32F051C6 and the ATmega1280. They both met the requirements set out in subsubsection 3.3.2. In the end, the STM MCU was chosen over an Atmel MCU based system mainly because of cost and it was more accessible at UCT. An Atmega1280 costs above R200 where as the STM MCU costs around R30 [30], at the date this report was compiled.

The STM MCU featured a 48MHz max system clock with 32Kbytes of flash and 8Kbytes of SRAM. It had 11 timers in total which would assist with the PWM channel and task manager. It had 27 available I/O channels. Of those 27 channels 10 could be mapped to the ADC, 6 could be mapped to the timers and used as PWM channels. Additionally, the STM MCU had 2 UART interfaces.

5.6 Delivery Mechanism Housing

It was decided that the housing structure for the delivery mechanism would be clear 2mm perspex. This was because acrylic glass would be able to handle the small forces applied to the mechanism. It was also chosen, as acrylic glass was easy to cut given that a laser cutter was used, making for accurate assembly of the design. Lastly, clear acrylic glass was specifically used so the housing would not impede viewing of the delivery mechanism for the assembly, maintenance and repair.

5.7 Enclosure Frame

Steel L beams and bars were chosen as the frame for the enclosure. Steel was chosen because it would be strong and durable making the enclosure last long. Also, steel was one of the easier metals to weld making it ideal, as the facilities were common for welding steel.

5.8 Enclosure Casing

Aluminium was chosen over acrylic glass for the casing for the enclosure. This was because aluminium is stronger than acrylic glass making it harder for people to tamper with the enclosure and breach it to access the interior. Acrylic glass and aluminium were roughly the same price per meter² making the advantages of aluminium the obvious choice.

5.9 Enclosure Shelving

Wood was chosen for the shelving for the delivery mechanisms, as it was the most cost effective solution when compared to steel, aluminium or acrylic glass. Wooden planks 12mm thick would be used to meet the requirements of the shelving.

6 System Design and Prototyping

The design of the Component Vending Machine is split up into three sections as prior sections have been. The mechanical design will be presented first, then the PCB design and finally, the Software development and design. Prototyping will be presented where relevant but not in its own section, in order to maintain its context with the design it is related to. All 3D models, unless specified otherwise, were produced using Solidworks.

6.1 Guide For Component Tubes

In order to help make restocking the Component Vending Machine easier, the delivery mechanism would be fed from the same component tubes the components were stored in. The idea was also to make the delivery mechanism gravity fed, an idea inspired by the literature review discussed in subsubsection 2.1.2 and subsubsection 2.1.4. This would mean restocking would only entail disposing of the empty tube and fitting in a full tube of components, instead of individually loading each component. A guide for these component tubes was needed to keep them in place and make restocking simple for the user. This guide was called the IC guide. Although the Guide did not change much through each version, improvements were made to compensate for each problem encountered. Although the Guide forms part of the delivery mechanism, it deserved an update review of its own, therefore, the relevant changes are presented in this subsection.

6.1.1 IC Guide Version 1

The Guide was to hold a component tube for DIP IC's which have a trapezoidal shape.

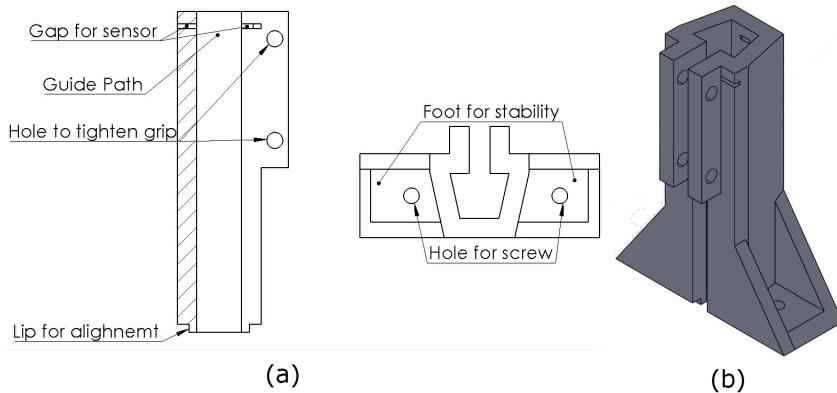


Figure 6.1: IC Guide (a) detailed diagram and (b) isometric view.

In Figure 6.1 the IC guide has been depicted in (a) detailed diagram showing features and (b) an isometric view for a real life rendering. The component tube would fit in the centre of the guide in the "Guide Path". Gaps were added for an IR LED and IR phototransistor so the rail could be monitored at the "Gap for sensor". This sensor would be able to detect if the level of the IC's in the tube were below the height of the gap providing information about the stock of the tube. Holes for screws were added to allow for tightening of the structure to grip onto the tube at "Hole to tighten grip". Two holes were made by different heights to distribute the load of the screws, reducing pressure applied at each point and thereby preventing failure of the IC Guide from cracking or snapping. This was tested with a mock build, which showed signs of fatigue with only one compression hole. A lip was added at

the bottom to properly align it with the delivery mechanisms structure shown as “Lip for alignment. Holes on the feet were also added to fasten the guide to the delivery mechanisms’ structure, shown as “foot for stability” and “Hole for screw”.

This version was deemed unusable, as tubes of varying thickness were encountered, making it impossible for the delivery mechanism to operate reliably. This was caused by the guide squeezing the tubes and causing a jam in the tube.

6.1.2 IC Guide Version 2

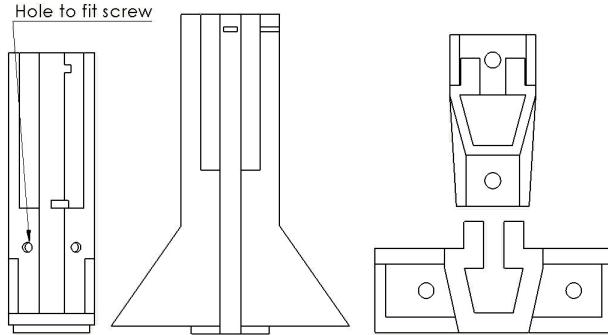


Figure 6.2: Comparison between Version 1 and Version 2 of the IC Guide.

Version two of the IC guide improved on the design of the first and addressed the problem of thicker tubes by expanding the guide hole slightly. Other changes included making the walls of the guide thinner to reduce material costs for 3D printing and adjusting the feet from the sides to the front and back. The changes can be seen in Figure 6.2. This helped to reduce the footprint of the overall structure of the delivery mechanism. Additionally, holes were placed so screws could fit to press up against the tube and further fix it in place.

6.1.3 IC Guide final

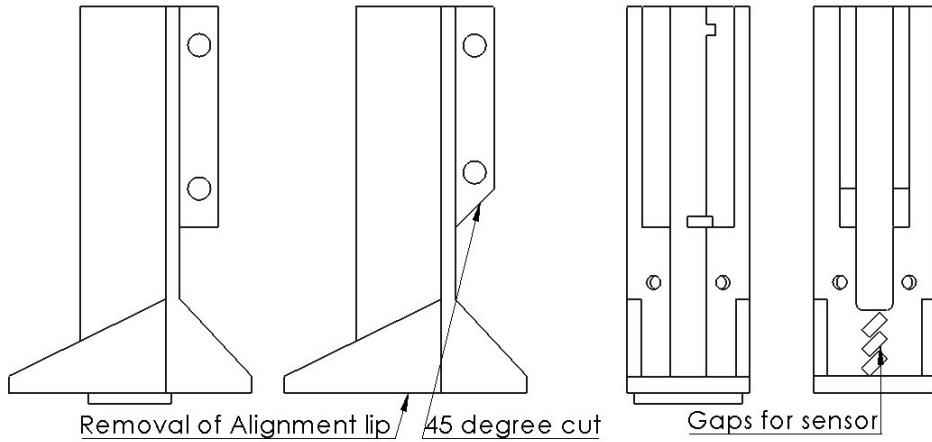


Figure 6.3: Comparison between Version 1 and Version 2 of the IC Guide.

The final version of the IC Guide changed three details of the design. First, the gap for the sensor was moved to the bottom and 2 additional gaps were added to allow for alignment of the gap and IC to be dispensed. The alignment lip was removed to allow for a more adjustable design per IC and compensate for possible misalignment caused by manufacturing and assembly. Lastly, all 90°corners

which were overhanging were changed to 45° slants to reduce supports being produced in 3D printing, reducing material costs. These changes can be seen in Figure 6.3. Additionally, a guide for IC holders was designed, as the tube for IC holders were rectangular. The guide hole was simply changed to fit this shape. This has been illustrated in Figure 6.4.

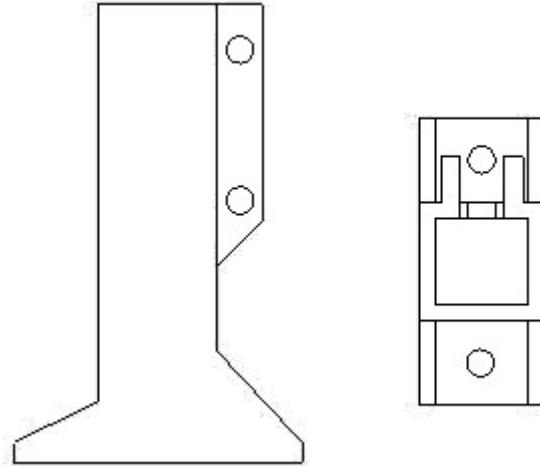


Figure 6.4: Rectangular IC guide for DIP IC holder tubes.

A detailed drawing of the final design of both the IC Guides, DIP Components and DIP Holders, can be seen in Appendix B.1 IC Guide DIP Components and Appendix B.2 IC Guide DIP Holder.

6.2 Delivery Mechanism

The delivery mechanism design and prototyping will be presented in this section, with version ranging from the preliminary design to the final design.

6.2.1 Vertical Roller Version 1

The vertical roller was named after the axis the part pivoted on. This pivoting motion was performed by a servo motor. This roller was designed to be able to be 3D printed, using PLA or ABS.

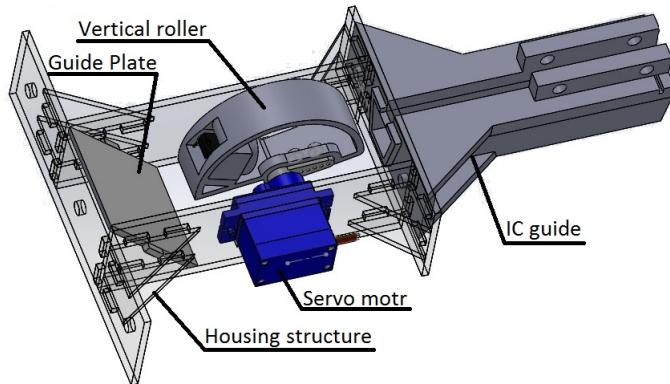


Figure 6.5: Vertical roller and housing delivery mechanism.

The “Vertical Roller” as labelled in the Figure 6.5 was situated in the “Housing Structure”, made of 2mm acrylic glass, that would enable it to pivot on the vertical axis with the assistance of a servo

motor. The housing structure had a width of 103 mm to keep it stable, allowing for 9 delivery mechanisms to be placed on a 1 m span. The design of the roller consisted of a bucket for the DIP component to drop into. This bucket would be situated on a wheel that would pivot the bucket from a position where a component would drop into the bucket. The bucket was designed to accommodate a component with splayed legs, hence took on a trapezoidal shape. The dimensions of the bucket were determined by measuring dimensions on a DIP component. The wheel would then pivot to a position where the component would fall out. This position has been depicted in the Figure 6.5. A plate labelled “Guide plate” was designed to force components to fall away from the housing if they fell straight down. The components would be guided into the bucket by the “IC Guide”. The design of the IC guide has been detailed in subsection 6.1. A more detailed drawing of the delivery mechanism in its housing can be seen in Appendix B.3 Vertical Roller & Housing Version 1 & 2.

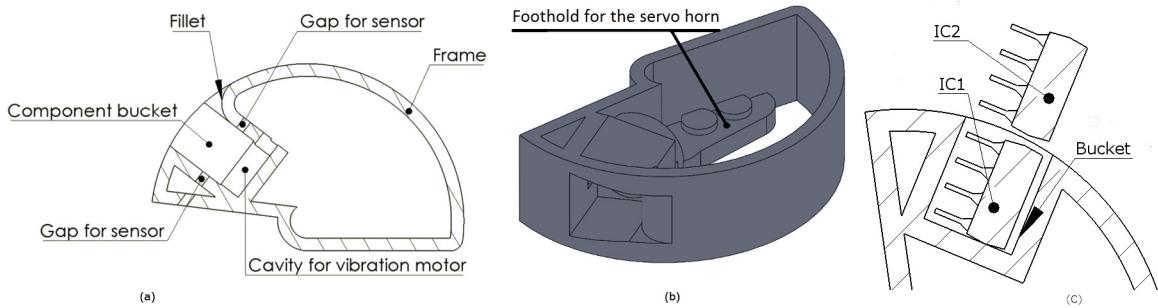


Figure 6.6: Vertical roller (a) cross section (b) isometric view (c) view to help explain loading.

The Figure 6.6 shows a cross sectional (a) and isometric view (b) of the vertical roller. The cross sectional view shows the features of the part. The cavity labelled “Component bucket” is where the DIP package would fall into when being loaded, for this design it would be DIP 8 components. A “Fillet” was added to prevent the next IC to be loaded from catching on the wall of the bucket. The slot labelled “Gap for sensor” would house an IR LED and an IR phototransistor, one on either side. The cavity at the bottom of the bucket was for a vibration motor, which would assist in loading and dispensing components. The idea behind the vibration motor was to assist the component if it became jammed. A “Frame” was to maintain structural rigidity. The isometric view shows detail of how the design would look in real life. A “Foothold for servo horn” has been labelled, which would be to assist the servo motor attach to the roller to pivot it along the vertical axis. The part was hollowed out to reduce material costs for 3D printing and the wheel form was used so the wall of the disk could be used to block IC2 as the roller would pivot.

In Figure 6.6, labelled (c), IC1 will be the name given to the component in the bucket. IC2, as shown in the figure above, will be the name given to component that would be loaded in the next loading cycle of the mechanism. This naming convention of IC1 and IC2 will be used in this report from now on. This diagram explains how IC’s would be guided into the bucket, one on top of another from a tube situated in the guide, which would be positioned above the delivery mechanism.

6.2.2 Prototyping Vertical Roller Version 1

A prototype of the delivery mechanism was built to test its functionality and reliability. A portion of the testing can be seen in the video [31, 32], or on the accompanying CD named “Video 1” and “Video 2”. The parts for the housing were laser cut from scrap acrylic glass reducing the prototyping costs. The vertical roller and IC guide were 3D printed using PLA.

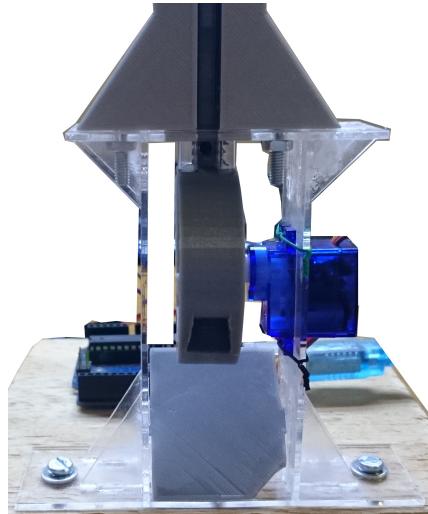


Figure 6.7: Prototype of delivery mechanism using the vertical roller.

In testing, the prototype had a reliability factor of just fewer than 76%. This made it unreliable and did not meet the requirement of 95%. The main cause of failure was IC1 pushing up on IC2 when the roller was rotating to release IC1, as depicted in Figure 6.8. Two other common failure modes that occurred, were IC2 being jammed up against the fillet and IC1 when being loaded, would catch on the lip of the bucket. Another undesirable characteristic observed, was that IC's would be ejected unpredictably, i.e., snapped out at speed, instead of a controlled fall.

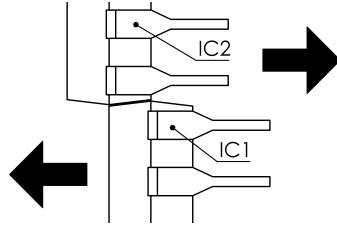


Figure 6.8: Common jam action that occurs with vertical roller.

6.2.3 Vertical Roller Version 2

Version 2 of the vertical roller was to improve on the mechanical failure caused by the fillet of the first version. A sloped fillet was added to prevent the IC1 from catching on the lip of the bucket when being loaded. It was deemed that the failure caused by jamming of the IC1 on IC2 could be mitigated by changing the way the mechanism operated by way of programming. Additionally, a centre hole was added to the servo foothold to make it possible to align the centre of rotation of the servo with the roller.

6.2.4 Prototyping Vertical Roller Version 2

The housing for version 2 was the same as version 1, as all that needed to change was the roller. The roller was swapped and the prototype was tested for reliability and functionality. A portion of the testing can be seen in the video [33], or on the accompanying CD named “Video 3”. Although the problem of IC1 jamming up against IC2 had been reduced because of changing the functionality of the mechanism, reliability of the device was still below 95%. At just above 86% the design was deemed unreliable. The common mode of failure of an IC being caught on the lip of the bucket was

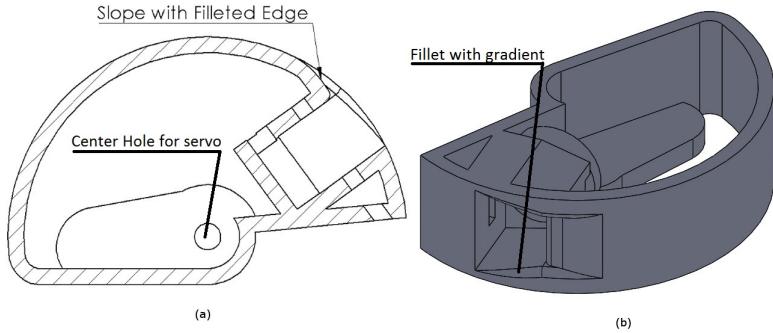


Figure 6.9: Changes made to the vertical roller for version 2.

still present. However, undesirable flinging of components was eliminated and IC's no longer jammed up against the slope which was a fillet in version 1.

6.2.5 Horizontal Roller Version 1

The horizontal roller just like the vertical roller, was named after the axis it pivoted on. Also driven by a servo motor, it was designed to be more reliable than the vertical roller and be 3D printed from either PLA or ABS.

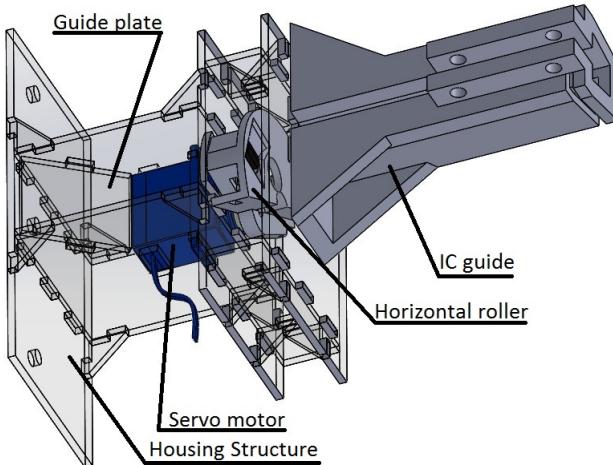


Figure 6.10: Horizontal roller and housing delivery mechanism.

The idea for the horizontal roller was inspired by a combination of the literature review in subsubsection 2.1.3 and subsubsection 2.1.4. The “Horizontal roller” labelled in Figure 6.10 would be situated in a “Housing structure” made of 2mm acrylic glass. The housing structure had the same span as the vertical roller. Additionally, the design was intended to be more adaptable with regards to the range of DIP components it could accommodate. In theory, if a longer DIP package was required, the height of the roller could be modified in order to satisfy the requirements. Another benefit of this design was that the wall of the servo and the housing created a closed channel to expel the components in a controlled direction, reducing the possibility of a component falling back and failing to expel from the mechanism. A more detailed diagram of the housing and roller can be seen in Appendix B.4 Horizontal Roller & Housing Version 1.

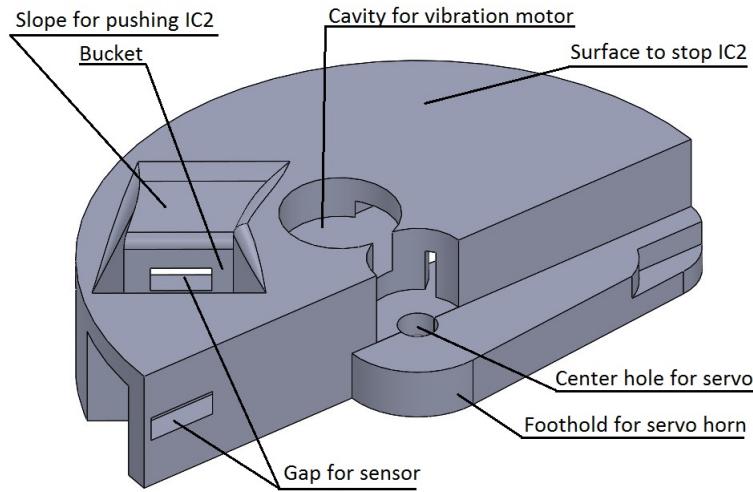


Figure 6.11: Labelled isometric view of the horizontal roller.

The horizontal roller has been illustrated in Figure 6.11. The feature labelled “Gap for sensor” would be for the IR LED and IR phototransistor that would detect if a component were in the bucket. A “cavity for a vibration motor” was made to assist loading of IC1 into the bucket. A slope was added to assist by gradually pushing IC2 up and the “Surface to stop IC2” would hold it up while IC1 would be dispensed. The feature labelled “Foothold for servo horn” and the “Center hole for servo” were to attach the horizontal roller to the servo horn and alight it with the centre of rotation for a balanced rotation.

This design worked by first loading a component, IC1, in the bucket. Then the roller would rotate about 90°. The component would then fall down, out of the bucket onto the guide plate and then be expelled. The roller would then rotate back to where it started to let another component drop into the bucket.

The horizontal roller was designed to eliminate some of the problems encountered with the vertical roller. Firstly, the problem of IC1 being caught in the bucket would be eliminated by employing a horizontal roller that used a different method to expel the component. Secondly, the problem pertaining to IC2 jamming up against IC1 as illustrated in Figure 6.8, was addressed by rotating IC1 instead of moving tangentially on the circumference of the roller away from IC2. This was because the force on IC1 by IC2 would be localized to one corner when rotating instead of being distributed across a surface when moving tangentially on the circumference of the roller. Lastly, unpredictable ejections would be eliminated as the component would be dropped down a guide controlling its fall path instead of being flung out as was the case for the vertical roller.

6.2.6 Prototyping Horizontal Roller Version 1

Another prototype of the delivery mechanism was built. The housing structure was made again as it was different from the vertical roller. Again, the housing was made of 2mm scrap acrylic glass, to keep prototyping costs low and the roller was 3D printed using PLA. A portion of the testing can be seen in the video at [34], or on the accompanying CD names “Video 4”.

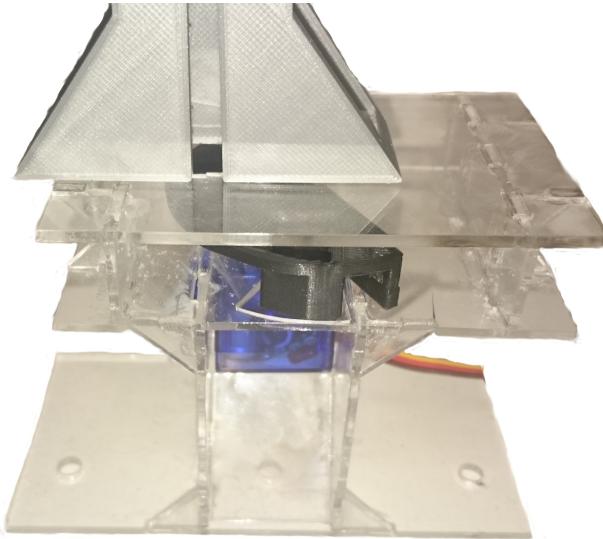


Figure 6.12: Picture of the prototype delivery mechanism with the horizontal roller.

The prototype was tested and was seen to have a reliability rating of just above 96% meaning it met the required 95% reliability rating. The main causes of failure were; not loading into the bucket and getting caught on the lip. Although these failure modes were reduced, they were still present. The vibration motor was intended to help with this issue. A second cause of failure was the component being lightly jammed in the bucket when trying to drop out. This was called a light jam as a small tap on the roller would expel the component. The vibration motor was intended to assist with this issue as well.

6.2.7 Horizontal Roller Final Version

The final version of the roller was designed to reduce printing costs as it needed supports to print. By using 45° slopes, which the printer could handle, the required supports would be reduced and bring material costs down. The housing footprint size was reduced in order to increase the density of delivery mechanism that could be packed into the enclosure, increasing the potential for more variety and density of components in the Component Vending Machine.

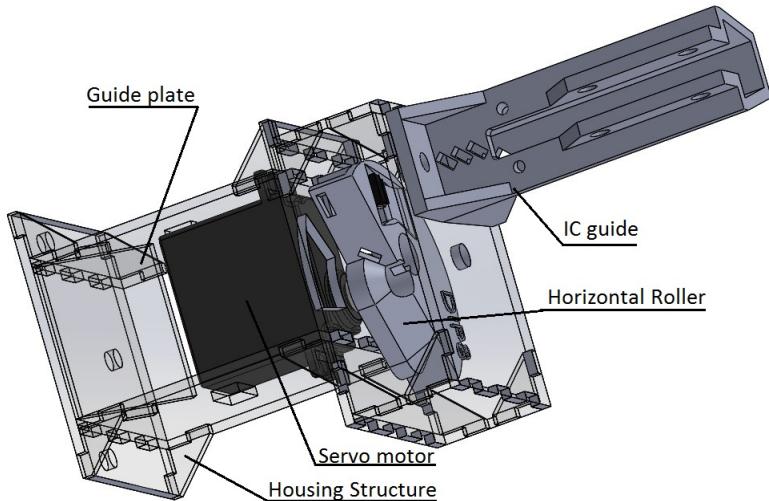


Figure 6.13: Final design of the horizontal roller and housing.

The servo motor was changed to a stronger servo motor, Corona DS-339, not because it needed the

power but because the previously used 9g servo intended to rotate the roller, used plastic gears and over time this could have led to threading. Whereas, the stronger servo motor had metal gears which would not thread as easily. Another reason for using the stronger servo was that it had a better build quality and responded to input more accurately. This accuracy refers to the 9g servo having overshoot when the roller was attached, causing a problem with the servos control algorithm and making it slightly unreliable. Lastly, the stronger servo was chosen over the 9g servo as it was more readily available than the 9g servo. The new “Housing Structure” now measured with a width of 65.4 mm meaning that on a 1m span, 15 delivery mechanisms could be placed. This was an increase of 6 from the previous design.

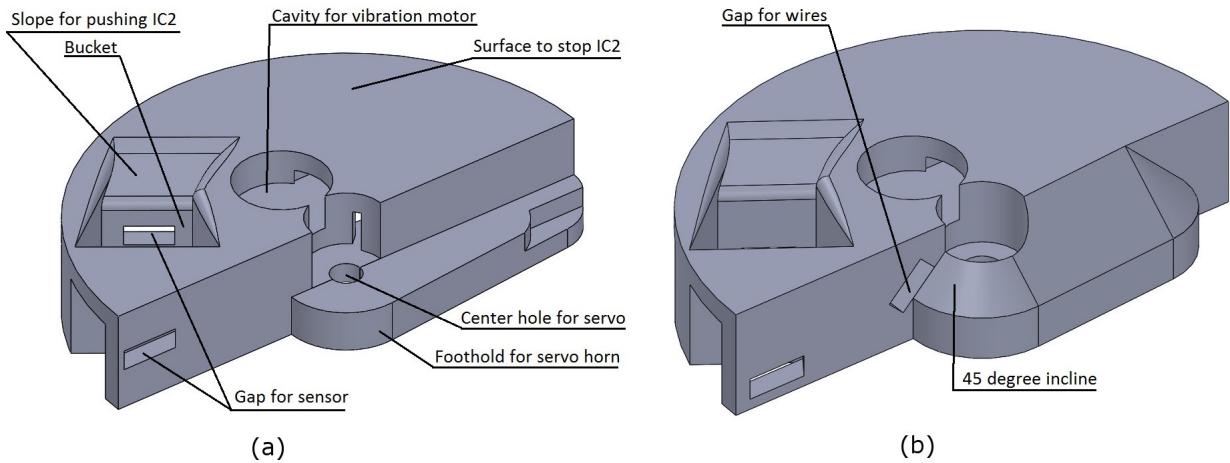


Figure 6.14: Comparison between version 1 and final version of horizontal roller

In addition to 45°inclines, where supports in 3D printing would be needed, a “Gap for wires” was added as can be seen in the side-by-side comparison of the roller versions. This gap was added so that the wire from the sensor on the front of the bucket could be tucked back through the hole and be neatly attached to the PCB behind the roller. The final design catered to multiple package sizes of DIP components, those being DIP 8,14 or 16 and 20.

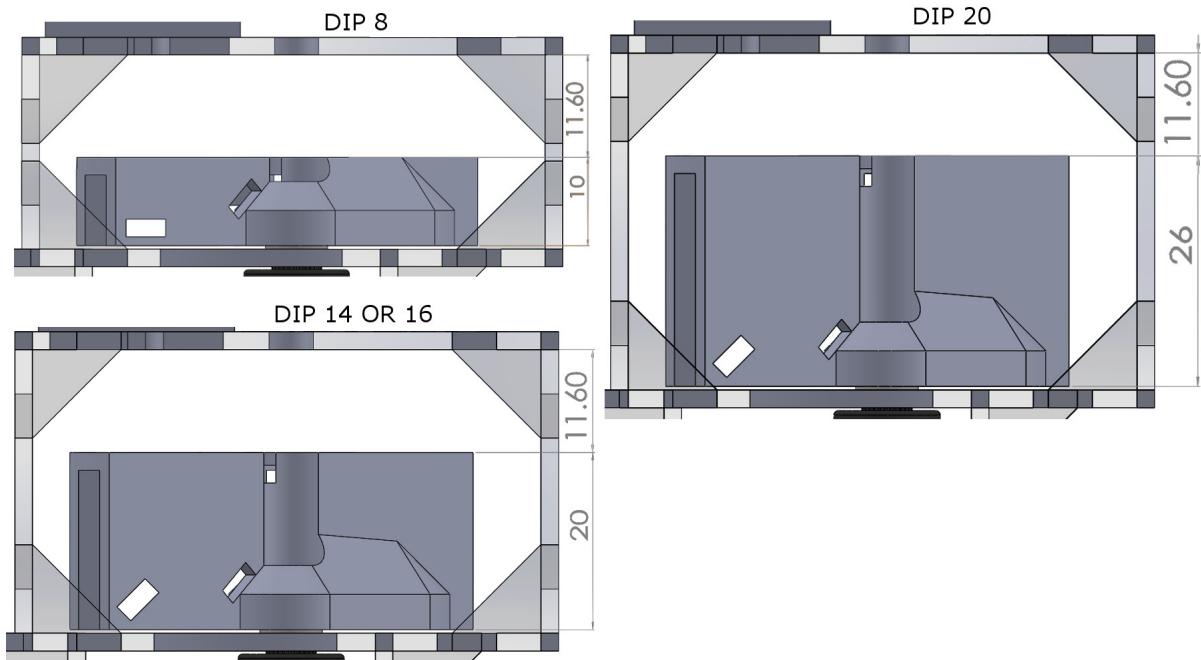


Figure 6.15: All rollers to dispense DIP 8, 14 or 16 and 20 for comparison.

The heights of the rollers wheels were determine by using the average of the component length

obtained in Table 3.1 and rounding to the nearest integer. Designs were produced to cater for these measurements as shown in Figure 6.15. Detailed drawings of the delivery mechanism that accommodates each version of component package length can be seen in Appendix B.5 Horizontal Roller & Housing Final Version. Versions of the roller for DIP IC holders were also designed, providing for different heights for each package length. All versions of the DIP roller can be seen in a detailed drawing in Appendix B.6 Horizontal Roller Final Version.

6.2.8 Untested design: Feed-Up

A third delivery mechanism was theorized and partially designed. It was not tested or built as the horizontal roller worked well enough that an improved design was not necessary. The feed-up method used a very different delivery method. Instead of using a gravity fed system, the components would be pushed up and out of the component tube. This would be achieved by using a stepper motor and a timing belt with a plate attached to the timing belt being pushed up the tube.

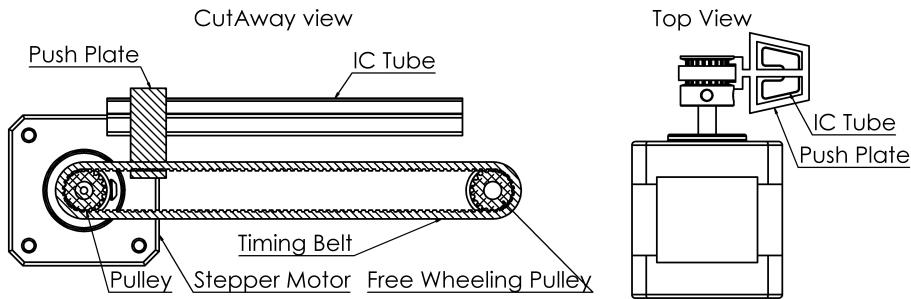


Figure 6.16: Feed up design with cutaway and top view to illustrate how it operates.

This design originated from using a feed screw to push the components up. That was deemed inefficient as it would require a lot of space to house the delivery mechanism with a feed screw and an IC tube. The solution, as illustrated in Figure 6.16 was to use a timing belt with a small push plate-like attachment that would gradually push the components in the component tube up, as the timing belt progressed. The component would reach the top of the tube and be pushed out and fall. This fall would be controlled by tilting the mechanism in a specific direction so the fall could be predictable. The timing belt would be driven by a stepper motor with pulleys, as the fine movement from a stepper motor would provide the accuracy needed. The steps could be counted to determine how far the plate was up the tube and how far it needed to travel to dispense a component.

The reason why this design was not favoured was because of its complexity and the fact that a modified component tube would be needed. This modified component tube would be fixed and the restocking process would entail manually placing components in this tube. The complexity stemmed from the mounting, although trivial by itself, it would be more complicated than the previously discussed solutions. This design may have had some advantages as it was designed to be non-component specific, meaning one design could satisfy the specifications of multiple package sizes. The success of the horizontal roller was the main reason why the feed-up design was never tested.

6.3 Enclosure

This section presents the design of the enclosure that would house the delivery mechanisms and the electronics that made up the Component Vending Machine. The enclosure design started with a frame to attach all the components and supports needed. The frame was divided into two sections to determine the measurements. Th bottom section would be for a guide slope for components to

fall down into a compartment for the user to collect their order. The upper section would be the compartment housing the interior parts of the Component Vending Machine.

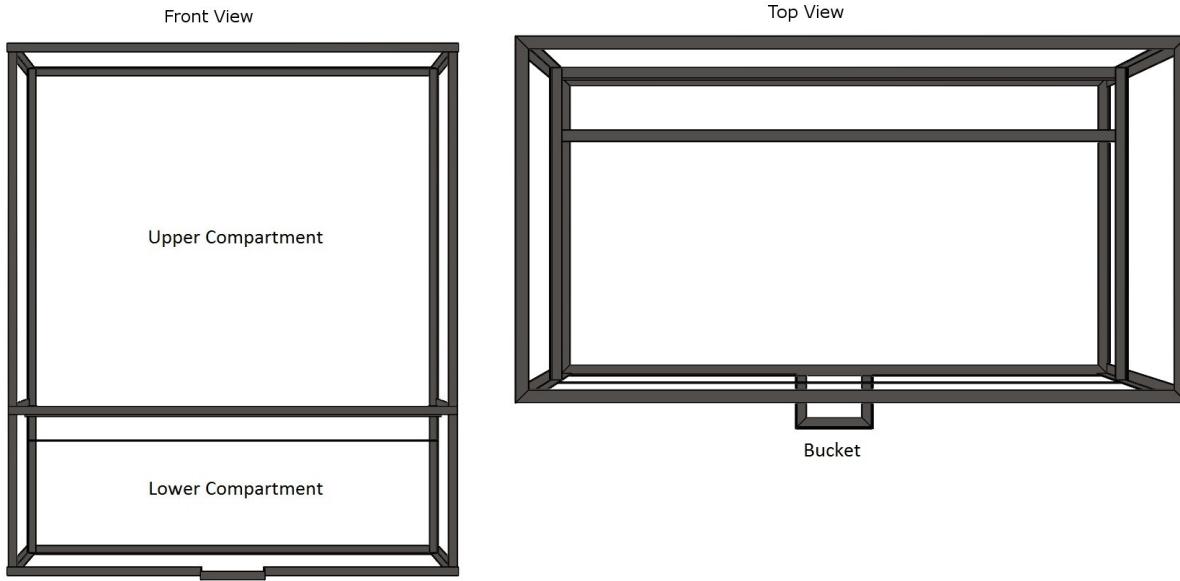


Figure 6.17: Frame of the enclosure illustrating the upper and lower compartments.

The “Upper Compartment” of the frame, as illustrated in Figure 6.17, was designed to fit the tallest delivery mechanism, that being the DIP 20 dispenser and the longest measured component tube, plus a little room for error as well as potentially taller component tubes. The DIP 20 dispenser measured in at about 100 mm, from the base to the top of the housing structure. The tallest component tube measured came in at about 600 mm, with an additional 150 mm for potentially longer tubes and 10 mm for error, leading to an upper compartment height of 860 mm. The width of the enclosure was determined by the space required to fit 3 rows of dispensers in the Component Vending Machine leading to a width that was rounded off to 600 mm. The “Lower Compartment” was designed to fit a sloped piece of aluminium that would guide the dispensed components to a “Bucket”, labelled in the top view, accessible to the user to collect the component. This sloped piece of aluminium has been illustrated in Figure 6.18.

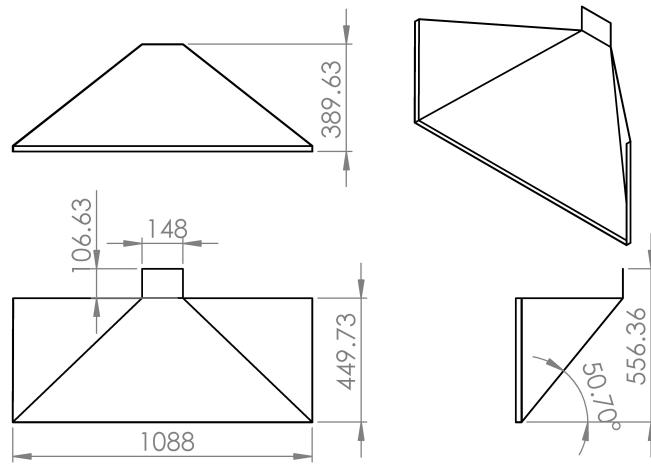


Figure 6.18: A detailed drawing of the sloped plate.

This sloped plate was designed to have an angle of incline above 45° to ensure that components would not be caught on the slope and fail to reach the bucket. For this reason, 50° was chosen. A render of the plate can be seen in Figure 6.18. The remaining dimensions were purely designed to fit within the

enclosure. The Enclosure casing was designed next. Aluminium sheets were used to cover the frame and prevent intentional tampering.

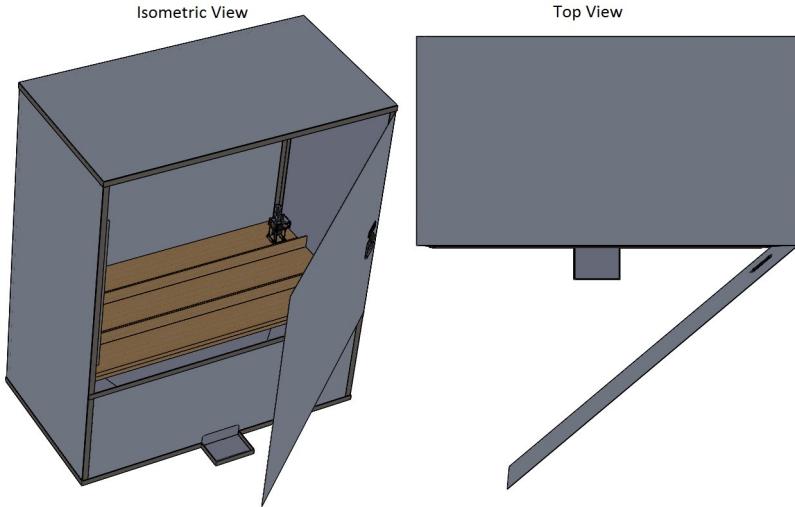


Figure 6.19: Isometric and top view of the enclosure with the case on interior.

The sheets that covered the enclosure can be seen in Figure 6.19. Inside, one can see 3 planks of wood as the shelving. The wood plank furthest to the back has a delivery mechanism situated on it to illustrate how the mechanism would be placed and a reference for size. The first plank, being the rear most plank, was intended to be installed for the Component Vending Machine. The other two were to represent the means for expansion when the first plank would run out of space for more delivery mechanisms. Drawings of the parts that made up the enclosure have been provided in Appendix F.5 Exploded Enclosure View and Appendix D Material Request Forms. These were used to assist assembling the enclosure and will be discussed in more detail at a later stage of this report.

6.4 Enclosure Redesign

A redesign was needed for the enclosure, as problems were encountered when trying to manufacture the sloped plate illustrated in Figure 6.18. The bending needed to be outsourced. The bending required was not feasible and so a small redesign of the enclosure was done.

As can be seen in Figure 6.20 the initial sloped plate was replaced with the “Simpler Sloped Plate”. It would be feasible to bend this simpler plate with the tool at hand, eliminating the need to outsource the bending. In addition to changing the sloped plate, the front door was extended to reach the bucket at the bottom. The bucket was also widened to cater for the changed sloped plate. This redesign was done after the material that was ordered. No additional material was needed. The material ordered was adequate to complete the redesign.

6.5 RFID Coil Cover

A cover for the RFID reader was needed to protect the coils from damage that could be caused by the user. The cover was a simple design encasing just the coils and not the circuitry it connected to, as this would be situated behind the cover, inside the enclosure.

The cover can be seen in Figure 6.21 from the rear and the front view. The cover was designed to be easily identifiable with the universal RFID symbol indented on the front of the cover, making it

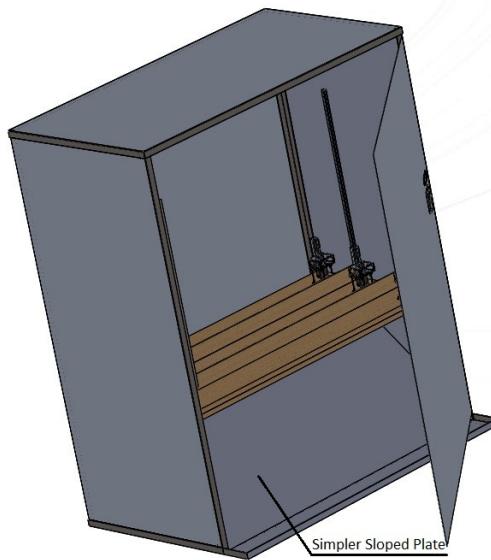


Figure 6.20: Isometric view of the redesign.

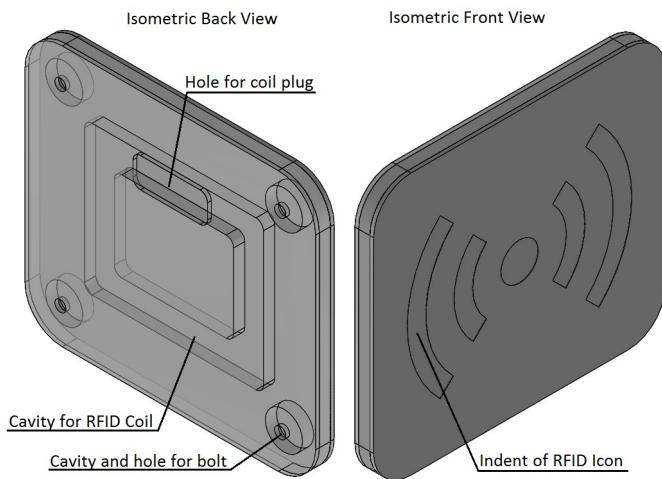


Figure 6.21: Isometric back and front views of the RFID coil cover.

easy for the user to know where to swipe their card, illustrated by the item labelled “Indent of RFID Icon”. The cavity was designed to fit the coil with a bit of room for unpredictable coil measurements. This has been labelled “Cavity for RFID Coil” in the figure. “Cavity and hole for bolt” labelled above illustrates one of the cavities made for a bolt and washer to be situated to make the cover mountable on the casing of the enclosure. It was designed as such so that no bolt was visible on the front side making for a clean design for the user to see. A detailed drawing can be seen in Appendix B.7 RFID Coil Cover.

6.6 PCB Design

This section presents the designs that made the electronics of the Component Vending Machine. The PCBs designed in this section were made in Eagle CAD.

6.6.1 Raspberry Pi HAT

The Raspberry Pi HAT was designed using measurements of the Raspberry Pi so that it would be able to fit on top of the Pi with standoff bolts allowing for a more compact design. A schematic of the circuit needed was initially designed in order to add the required components. This schematic can be seen in Appendix C.1 PI HAT Diagram.

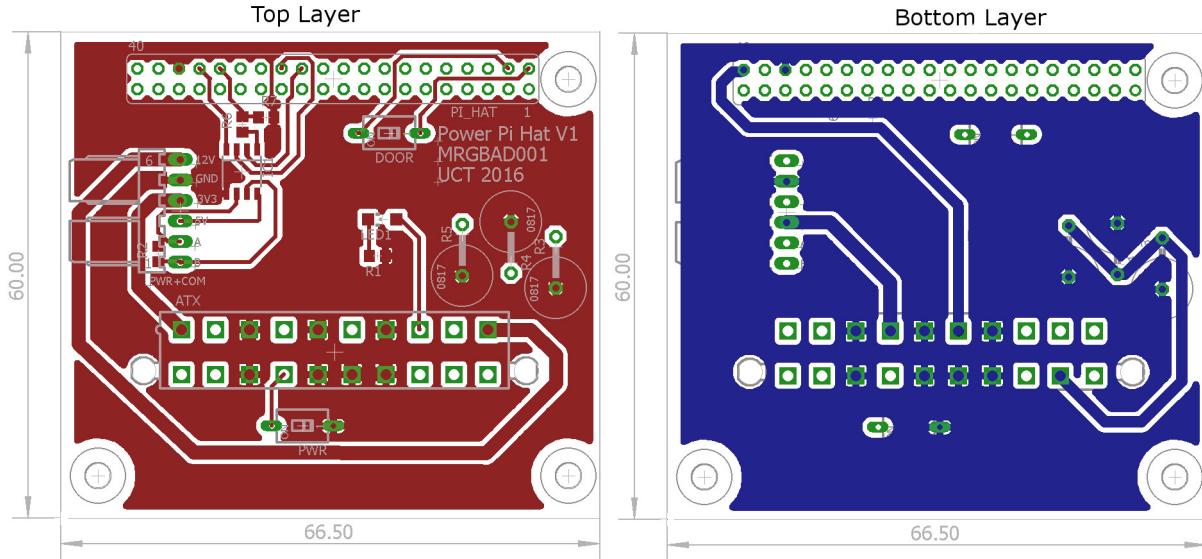


Figure 6.22: PCB layout of the Raspberry Pi HAT, top and bottom layer.

The Raspberry Pi HATs main purpose was to be able to connect to the power supply and create a starting point for the power bus, as well as provide a way of connecting to the RS485 bus. The PCB layout has been illustrated in Figure 6.22. In order to attach the ATX plug from the computer PSU to the Pi, a Molex socket was used that met the standards of the ATX plug. This connector has been labelled “ATX” in the Figure 6.22. The differential bus transceiver discussed in subsection 5.1 was used to facilitate the RS485 communication, labelled “IC1”. The receive signal was divided by a voltage divider, labelled “R6” and “R7”, as the Raspberry Pis inputs were 3v3 tolerant. In addition to the main features, a switch was added to turn off the power supplied to the system from the computer PSU, labelled “PWR”. A switch for the door was also added so that the status of the door could be detected and has been labelled “DOOR”. Finally, a bank of resistors was added to stabilize the computer PSU by drawing 5W, tricking the PSU into thinking it was connected to a motherboard. These have been labelled “R5” to “R3”.

6.6.2 MCU Module

The MCU module was originally designed to sit flat behind the delivery mechanism it was intended to control. It was later decided that it would be propped up vertically to help increase the density of the delivery mechanism that could fit in a meter length. The schematic was initially designed and can be seen in Appendix C.2 MCU Module Diagram.

The top and bottom layers of the PCB layout has been illustrated in Figure 6.23. The MCU external oscillator was placed first, followed by the DIP switch, “SW1”, and SN75176B placed to facilitate the basic operation and communications, “IC1”. Sensors and headers for the power communications and motors were then placed; “GAPSNR1”, “GAPSNR2”, “VBRMTR”, “SERVO”, “STEPPERMTR”, “PWR+COM1”, “PWR+COM2” and “PROG”. A potentiometer was placed at “R13” to assist with calibration of the modules. Terminals for selecting calibration mode and connecting the trimpot was

placed at “MODE” and “TRIM” respectively. In addition, 3 LED indicators were used in the bottom left corner showing green for power, red for MCU on and yellow for status. Holes were added to provide for mounting the board on standoff bolts.

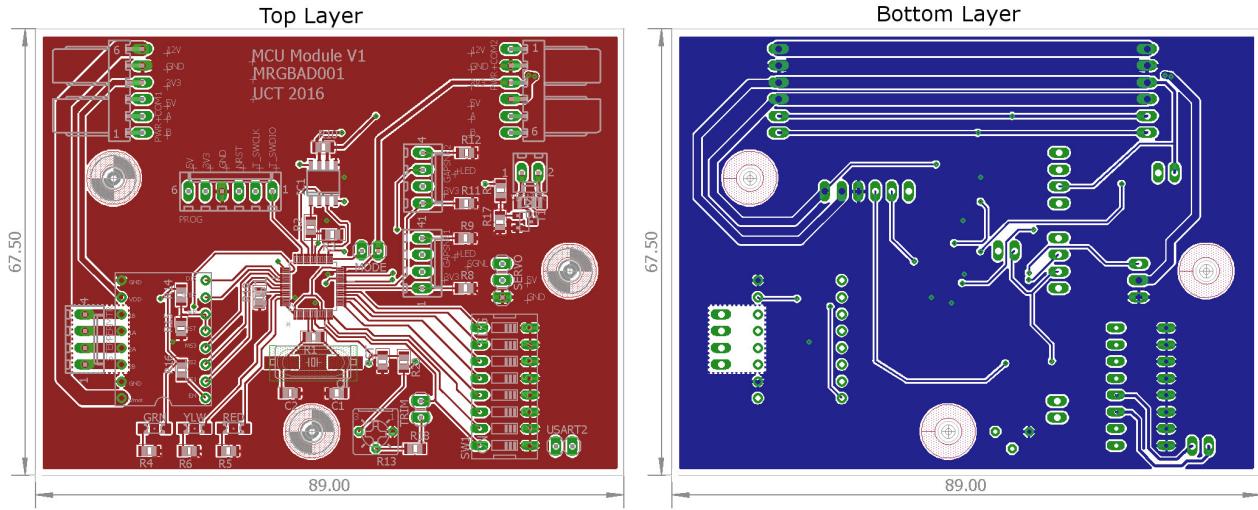


Figure 6.23: PCB layout of the MCU module, top and bottom layer.

6.6.3 LCD Breakout Board

An LCD breakout board was needed as it was later determined an LCD was to be used to give users feedback on the status of the dispensing session and other messages. The breakout board was designed because of the need to fit it to a board that it was not designed to fit on. The decision to use the LCD was made after the MCU module it was to be connected to, was sent for manufacturing. One of the MCU modules would be able to interface with the LCD breakout board because the initial design did not need a stepper motor, making them available for use for the LCD. A schematic of the LCD breakout board can be seen in Appendix C.3 LCD Breakout Diagram.

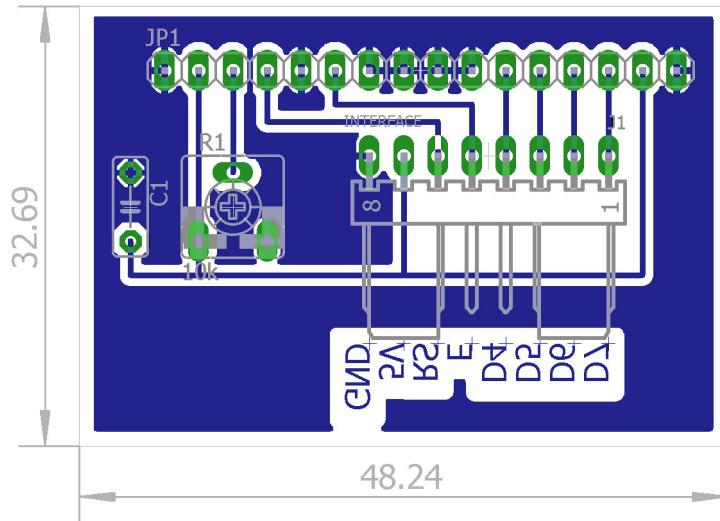


Figure 6.24: LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlaid.

The PCB layout of the LCD breakout board has been illustrated in Figure 6.24. Unlike the previous two PCBs, the board was designed to fit on one layer. It was for this reason that the silkscreen was shown on the bottom layer to illustrate where components would be placed. The board was designed to fit under a 16x2 LCD module using standoff bolts. The LCD would be attached via the 16 way

female header at the top of the board, labelled “JP1”. The back light was wired via tracks on the PCB. A trimpot, at “R1”, was attached to adjust the contrast ratio of the printed text on the LCD. A Molex header was used, at “J1” to interface with the LCD with only the relevant pins being given a header pin with each one labelled on the tracks. This was done so that no silkscreen was needed to reduce manufacturing costs and simplifying the assembly. This simplification was designed to make it possible to make a homemade etched PCB.

6.7 Software Design

This section will cover the design and basic testing of the software. This includes software on the microcontroller, the Raspberry Pi and the Website. First, an overview of the software design will be covered to provide an idea of how each program would interact with one another, followed by a discussion of the API before covering the actual software.

6.7.1 Software overview

The Component Vending Machine was designed to have a very clear and modular design, with each piece of hardware and software acting independently from one another. The reason for doing this was so that, if one module were to be swapped out for another, the whole system would not need to be redesigned. This would promote improvement and further development.

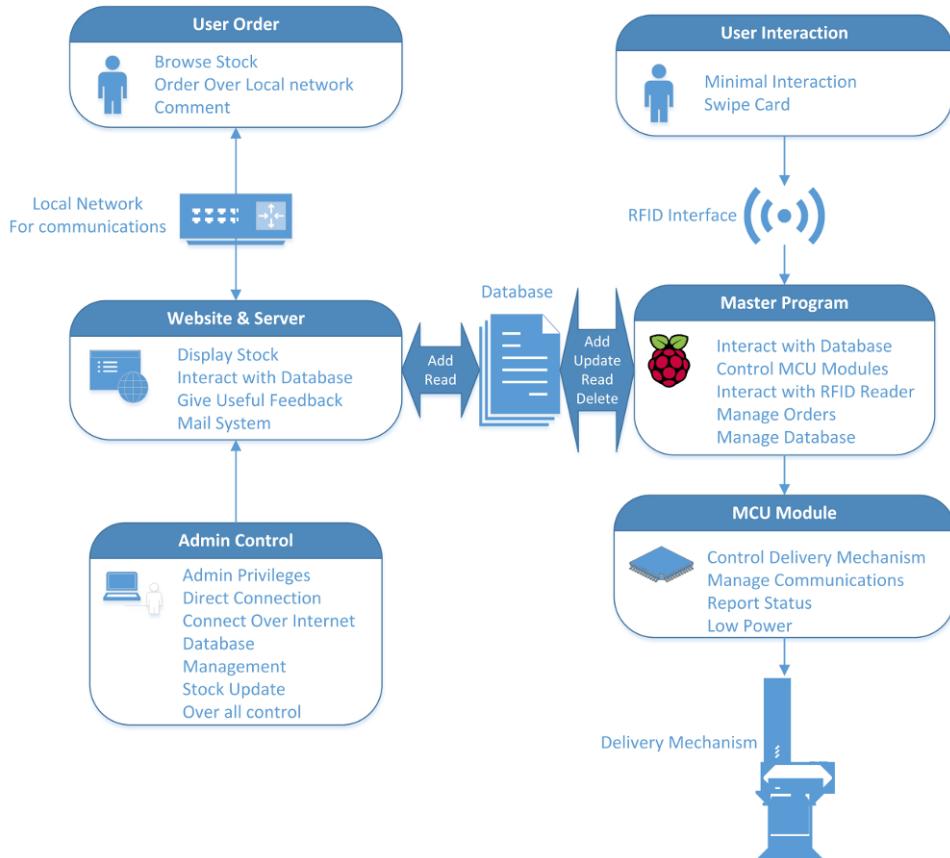


Figure 6.25: LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlaid.

The Figure 6.25 illustrates the different modules and how they would interact for the purposes of this version of the Component Vending Machine. Users would order online using the website designed for the purpose of verifying the viability of an internet connected Component Vending Machine. This

website would be hosted on a LAMP web server, hosted on the Raspberry Pi, allowing the website to interact with a database. The website would only have the ability to add or read entries in the database giving it limited control. In addition, a mail server would be used to generate administrators status updates, for example; Empty rails. Administrators would also have access to the website with elevated privileges allowing them complete control of the web server, website and database. This would allow administrators to edit who would be notified by the email system, update and manage the databases on the web server, website updates and edits as well as Master program updates.

The Master program would also be connected to the database. This way there would be no direct interaction between the website and the Master program, everything would run through the database. This would be a safer way of managing orders, as the database tool itself has protection for invalid entries. In addition to this, the Master program would have database management tools with elevated privileges, allowing the Master program to add, update, read and delete entries in the database. The Master program would also need to interact with the RFID reader and MCU modules. The routines on the Master Program would however only be invoked when there were a user interaction through the RFID reader. Once invoked, the Master program would communicate with the MCU module which would control the delivery mechanism. This module would be a low power solution, consuming very little power when idle. The MCU module would also report status updates back to the Master program.

6.7.2 API

In order for the Master program and MCU module program to communicate, an API was developed in order to maintain a reliable and trustworthy communications protocol. The API was defined for a package length of six bytes in total, from either master or slave, to keep it constant. The only time this deviated, was when the master would send LCD print data. The table below documents the API that was used to communicate between the Master and MCU programs:

Master to Slave API (in Hexadecimal)		
Byte Number	Name/Description	Values
1 st byte	Start Byte	0xA1
2 nd byte	Address Byte, of device to interface	0x00 - 0xFF
3 rd byte	Command Byte: 1. Call for Blank Response 2. Dispense Components 3. Free up Jammed Device 4. Print on the LCD	Specific Values: 1. 0xB1 2. 0xB3 3. 0xB5 4. 0xB7
4 th byte	Value Byte (no. pertaining to previous byte): 1. No effect 2. Number of components to dispense 3. No effect 4. 4 th byte to 3 rd last byte would contain message to display on the LCD	0x00 - 0xFF
5 th byte	Check Sum Value	Calculated using Bytes 1 to 4
6 th byte	End Byte	0xF1

Table 6.1: API used to communicate from master to slave.

The API was defined in such a way that it would be difficult for a corrupt message to be processed, ensuring only valid messages were used. The API would be set in place as an additional layer of security for the communication, in addition to the differential bus. The tables Table 6.1 and Table 6.2 can be used to replicate the protocol for any device intended to join the communication line.

6.7.3 Web design

The website was designed using Sublime Text Editor with the framework, as researched in subsection 2.4, being the LAMP server. The majority of the content was designed with static HTML, with some PHP content embedded in the HTML. The overall design was produced to be user friendly and functional on most displays and browsers.

An image of the component store page can be seen at Appendix E.1 Components Store Page and a picture of the contacts page can be seen at Appendix E.2 Contacts Page.

The website heading and cover layout were designed first. An image of basic components and the UCT logo were edited together to make a cover photo for the website and to represent the topic of the website. A menu bar was placed underneath this cover photo to allow for navigation on the website. In addition, a border was applied to the website to make it stand out. These three features were standard on each page of the website. The Website would consist of two pages; the component store page and the contacts page. The components store page began with basic instructions. Followed would be a design to fit three columns of components order forms, a number seen to work on most display sizes, and as many rows as needed, determined by the amount of components needed on the store page. At the bottom of the component store page a text entry box, that would require a student number, with a submit button was placed. When the submit button were to be pressed the website would be directed to a PHP script that would check the order to make sure it did abides by the rules and

Slave to Master API (in Hexadecimal)		
Byte Number	Name/Description	Values
1 st byte	Start Byte	0xD1
2 nd byte	Address Byte, of device sending message	0x00 - 0xFF
3 rd byte	Report Byte: 1. Jam in delivery mechanism 2. Empty rail 3. Low stock in rail 4. Successful dispensary	Specific Values: 1. 0xC1 2. 0xC2 3. 0xC4 4. 0xC8
4 th byte	Constant value	0x01
5 th byte	Check Sum Value	Calculated using Bytes 1 to 4
6 th byte	End Byte	0xE1

Table 6.2: API used to communicate from slave to master.

correct it if needed. The script would then enter the data in to the database and give the user a small notification, in the form of a pop-up box, that the order had been processed and a breakdown of what the order contained. Then the user would be redirected back to the store page. The Contacts page was intended to simply allow users to contact the administrator with problems encountered with the Component Vending Machine or comments about improvements and additional features that could be implemented. It would contain an introductory message with an entry form below that would require; a name, student number, email address and message. A send button would be at the bottom of the form with which to submit the form. Once submitted the user would be redirected to a script that would send the relevant information in the form of an email to the administrator. Then the user would be redirected back to the contacts page.

6.7.4 Raspberry Pi Master Program

Both the Master program and MCU program were both designed in a modular fashion, in that most tasks were isolated into their own function, making it easier to adapt a certain function for alternative overall program functionality. This was done so that as the project progressed in the future, the software could be easily updated to adapt to future requirements.

Before discussing the program, the databases should be explained. Two tables were used in the database, one called “Orders”, for the orders that would be placed on the Component Store page and another table called “Components”, for a record of the dispensary mechanisms in the Component Vending Machine, the component they dispense and the address they would be using. Table 6.3 shows all the variables that would be in each table and what they would be for.

The Master program logic diagram can be seen in Figure 6.26. The diagram shows a simplified view of how the Master program operates. The program began by checking the RFID reader for incoming data. The database was then queried with the RFID data to get the student number. Extra checks were done to check if an administrator were present. If so, a free-up command would be performed, resetting the empty and jam flags in the database and relevant modules. The orders relating to the student number would then be acquired from the database and dispensary requests would be sent to each relevant module one by one. Responses from each module would be received after each module had finished dispensing and the Master program would act accordingly.

Name	Type	Description
Orders Table		
StudentNo	String	The student number related to the order
PartName	String	The component name being ordered
Quantity	Integer	The number of components being ordered
Date	Date	The date the order was placed
Done	Integer	A flag to indicate whether the order has been fulfilled or not
ID	Integer	A unique ID number given to each order
Components Table		
Address	Integer	The address related to a delivery mechanism
PartName	String	The name of the component, the address points to, dispenses
Jam	Integer	A flag to indicate a jammed or not at the address
Empty	Integer	A flag to indicate empty or not at the address
Low	Integer	A flag to indicate low or not at the address

Table 6.3: Database description of variables and what they are for.

The responses expected were; Jam, Empty, Low and Success. A Jam and Empty report would warrant the Master program to email an administrator advising the person to attend to the Component Vending Machine and the reason why. A Low and Success report would just update the relevant tables, but no email would be sent. All cases would give a report in the log. This will be discussed later in the report.

This would be repeated until all orders, relating to the student/staff number that corresponded to the requested dispensary action, were done. The program would then return to an idle state and wait for a student card to read and receive RFID data that would then begin this process again.

In addition to the above loop, additional user feedback was provided in the program. Firstly, a log was kept of all the activities of the program and whether there were errors or successful operations. Secondly, LCD messages were transmitted to the first MCU module, which should have an LCD connected, to give the user swiping his/her student card feedback. Lastly, limits in the Master program software were put in place to limit the number of components ordered by any one person in one day.

6.7.5 Microntroller Software

The MCU program logic diagram can be seen in Figure 6.27. Although the logic flow for the MCU module was fairly complicated and could be broken down into two sections; the initialization and integrity checks, then the main logic loop. The diagram only contains the main functionality of the modules program. Additional functions will be discussed briefly later. The state machine diagram for the MCU module and its software can be seen in Figure 6.27. In order to gain a better understanding of the diagram, key features were highlighted. Those not highlighted should be explained by the diagram itself.

The MCU module program would start off with initialization. The initialization would entail setting up registers for the required inputs and outputs including PWM and ADC channels. Also, the Timers and UART channels would be setup. Lastly, the address of the module would be saved from the DIP switch meaning the address would only be recorded once. If the address were to be changed, the module would need a reset in order for it to take effect. Once initialized, the integrity checks would begin. The first check would be the “Mode” check, which would determine whether the module should run the dispensary or the calibration task. If the calibration mode should be selected, the module would run through different angles on the servo motor for loading and dropping a component. The module would hold at each angle for a set period of time and the angle on the servo could be adjusted

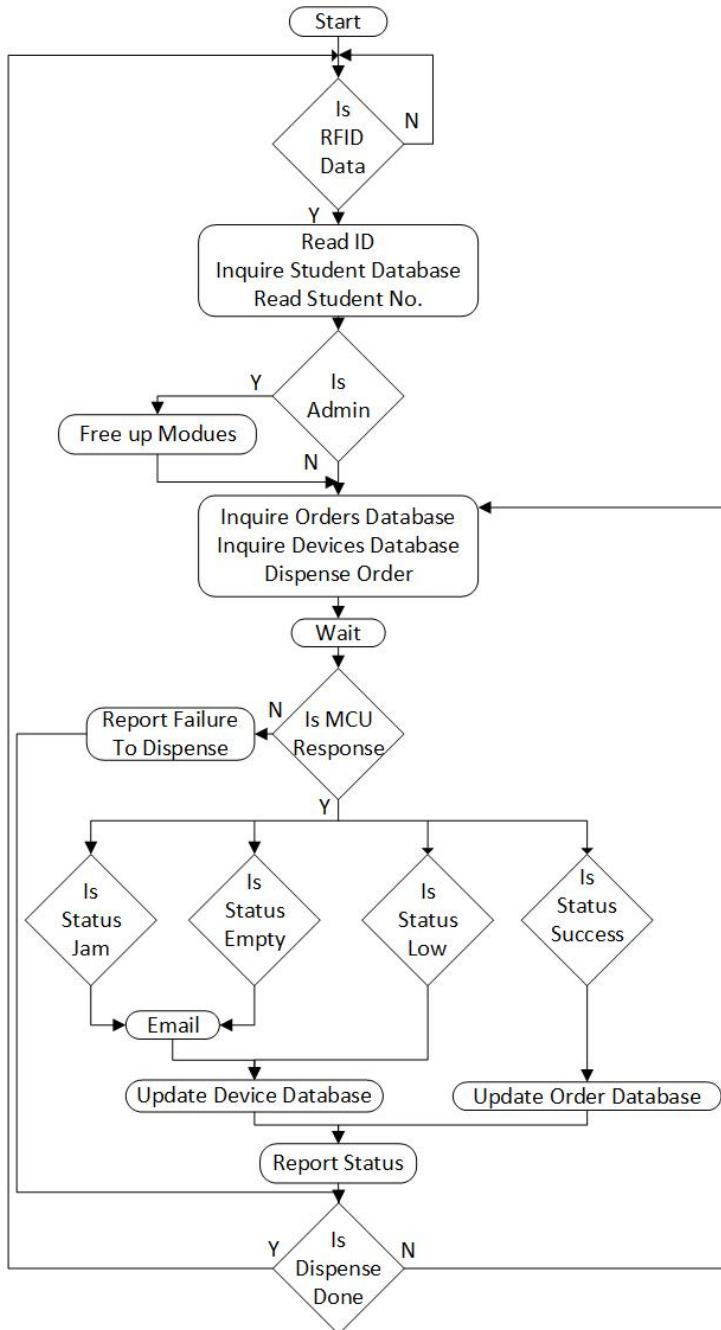


Figure 6.26: LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlaid.

to match the delivery mechanism using the potentiometer. If in delivery mode, the module would run the dispensary task when requested via the UART communications over the RS485 bus. The module would idle in a low power state until serial data would be received. This low power state would release the angle on the servo by using a 0% pwm and turning off the red and yellow LED. The check for jam or empty states would be done next, which would have been set, if a previous attempt at dispensing occurred in a jam or the tube on the delivery mechanism was empty. The only way to reset this flag would be by sending a free command from the master. Two other major checks needed explaining. The first being “Is IC In Bucket”, which would sample IR sensor on the bucket to determine if a component were in the bucket. The second would be the “Is IC In Tube”, which would sample the IR sensor on the IC Guide to determine if there were any components left in the tube.

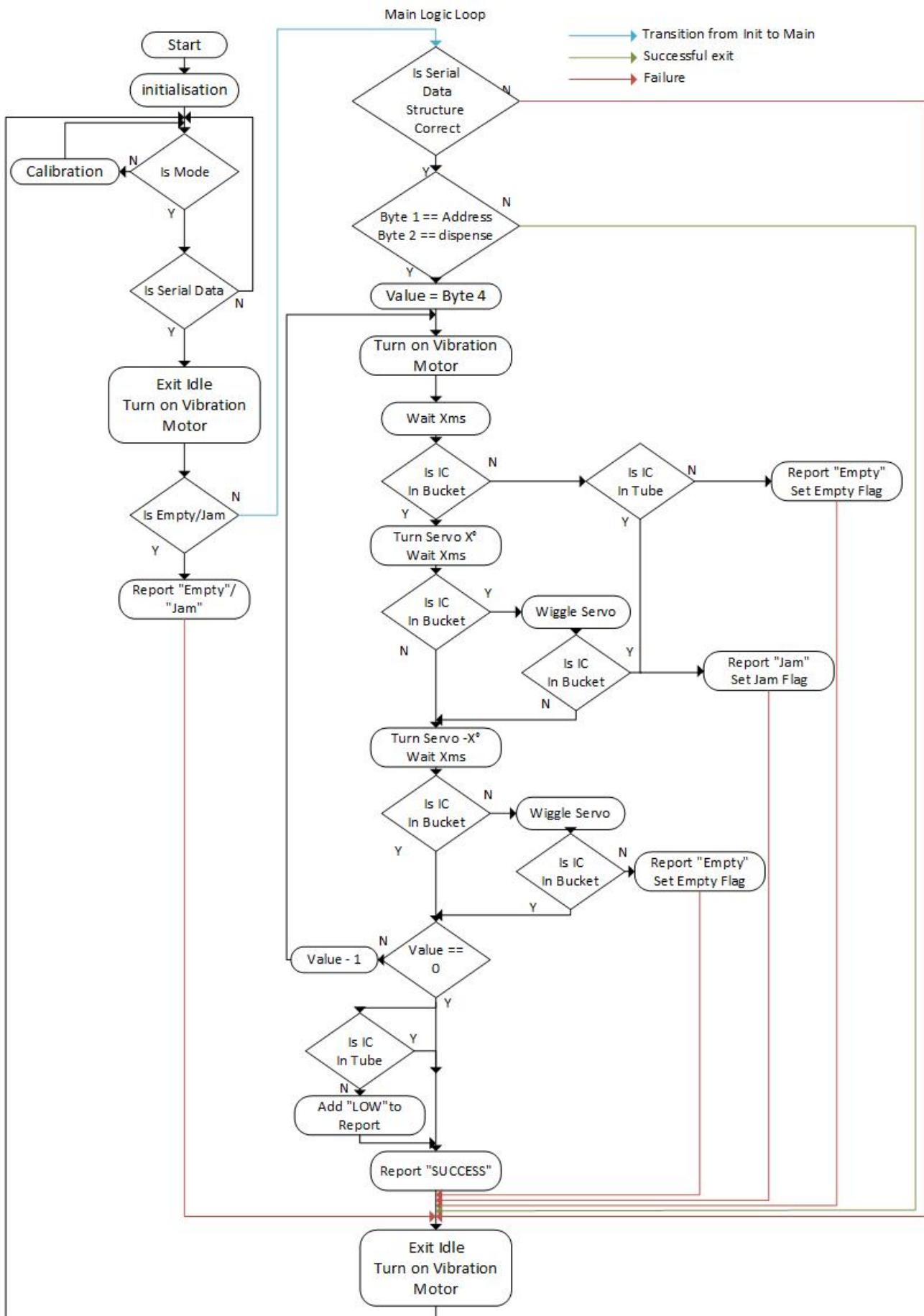


Figure 6.27: State Machine Diagram for MCU software.

The last feature to be highlighted is the reporting and where “Add” is used. This was highlights the fact that multiple reports can be sent simultaneously by using bit logic, where there were four report codes. The last four bits of the report code byte, represented one report code each.

Additional features, not covered in the diagram, were also made available for the module. These came into effect when a different command, than a “Dispense” command, matched those in the API. The first would be a “Call” command which would prompt the module to send a blank “Success” report back. Next the “Free” command would prompt the module to set the Jam flag back to default allowing the module to resume normal operation. The last being the “LCD” command, which only applies to the module with the address 1, as it was intended that this module had an LCD connected to it. If an “LCD” command were to be received, bytes 4 to the 3rd last byte would be the data that should be displayed on the LCD. The data received should be in ASCII format. The normal integrity checks would be applied for all the above to ensure no corruption could occurred.

The alpha version of this program was tested on a prototype setup using prototyping cables and electronic breadboard to simulate the PCB prior to assembly to verify its’ functionality and to ensure it would not damage the PCB somehow. This was very successful and effective, as it helped debug various problems prior to having access to the PCB.

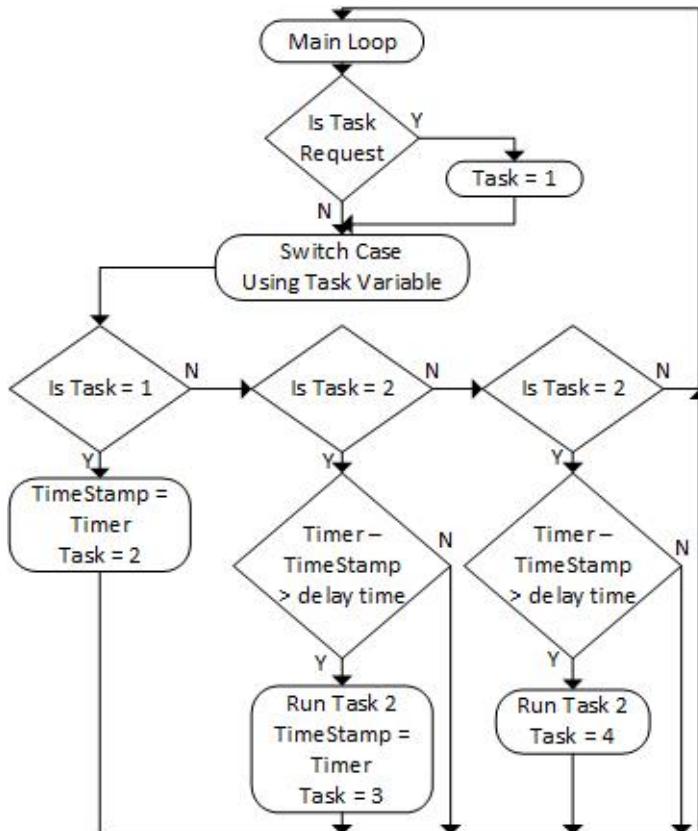


Figure 6.28: A simplified logic diagram of the task manager and how it would function.

One of the other features and specifications discussed in previous sections, was the minimal use of delays and a task manager to facilitate this. This would be designed using a timer that would be set to count up once every 1 millisecond. The design would be set to use a task counter and a conditional statement, based on the task counter, to facilitate invoking individual tasks. A task would entail an action that would occur without delays, for example when the sensors were to be sampled and the servo would be moved X°. After these actions would be taken, a short delay would be needed before the next action would be taken. In order to achieve this kind of task manager, a method was theorized and has been illustrated in Figure 6.28. This method could be implemented for any number of tasks and would be implemented in the MCU module programming.

7 System Assembly

This section will cover the assembly of the Component Vending Machine, from software, aspects which have not been covered yet, to assembling everything together. The Software assembly will be presented first, followed by the enclosure assembly. Once the enclosure assembly has been presented, the PCB and delivery mechanism assembly will be presented, one by one. Finally, a report of putting it all together will be presented. Where prices are quoted in the bill of materials, the prices were acquired from either RS components [30] or from the receipt from purchasing the material unless stated otherwise.

The prices quoted were up to date at the time this report was compiled and do not serve as a cost comparison for future assembly. Where pricing has not quoted the items were obtained from UCT from old or unused stock where the price was not readily available.

7.1 Software Assembly: Server

In order to host the website that had been designed in subsubsection 6.7.3, a LAMP server was installed in the Raspberry Pi.

An operating system needed to be installed on the Raspberry Pi before a LAMP server could be installed. The latest version of the Raspbian OS, a fork of the Debian OS, image was acquired from the official Raspberry Pi site [35]. The latest version of Win32DiskImager was then acquired, [36], and used to mount the Raspbian image onto a Micro SD card. Once the process was complete the Micro SD card was mounted in the Raspberry Pi and the Pi was connected to the network with an internet connection, with a CAT5 cable. The IP address was obtained by searching the network for the Raspberry Pi. Once the IP was obtained, a terminal to SSH into the Pi was used. In this case, the program used was Putty.

Once the Pi was powered and running, the installation of the LAMP server could begin. A detailed log of commands used to make this possible, has been included in Appendix I. The databases for the repositories was initially updated along with the software on the Pi itself. The LAMP server was then installed along with the required mail server for administrative notifications. The plugins for the Master program was then installed, along with registering the Master program to run on boot.

The bill of materials for the server can be seen in Table 7.1. All the materials for the server were provided by UCT.

Part Name	Cost	Quantity
Raspberry Pi	R 636.51 *	1
Micro SD Card (Class 10)	R 108.95 *	1

* Obtained from UCT

Table 7.1: Bill of materials for the Server.

7.2 Enclosure Assembly

The enclosure assembly followed the design using drawings from Appendix D and subsection F.5 to guide the process. Initially, materials were ordered, aluminium and steel L and flat bars. The orders were placed through UCT to known suppliers that had working relationships with UCT. Assembly of

the enclosure began with cutting the steel L and flat bars to length for the frame only. The frame was then welded using stick welding. Initially the base and top of the frame were welded, then the columns in the corners. The walls of the frame, made of 2mm aluminium sheets, were then cut using a guillotine. The walls were clamped to the frame and holes were drilled through the walls and the frame. The walls were drilled one by one until the 4 walls, 2 sides back and top, were all drilled and attached using bolts for removal if needed. A picture of the enclosure at this stage of the build can be seen in Figure 7.1.



Figure 7.1: Pictures of the enclosure frame with the walls bolted on.

Holes and openings were drilled in the front door plate and rear plate to accommodate the PSU, Ethernet cable, LCD and RFID cover. The next step was to bend the drop plate using a bending press. The height of the drop plate was marked inside the frame and the piece to hold the wooden shelves was welded onto the frame. A strip, which would hold up the shelves, was also welded onto these pieces to bolt the drop plate in place. The drop plate and frame were drilled and bolts were placed to secure it in place. The front door panel was placed on the frame and marked with the hinges in place. Holes for the hinges in the frame and aluminium were drilled and the hinges were secured with countersunk bolts and nuts. The next procedure was to drill the holes for the lock. The lock was then attached with countersunk bolts and nuts. A picture of the enclosure at this stage of the build can be seen in Figure 7.2. All the materials used are listed in the Table 7.2. The enclosure had final dimension of width, length and height on top as follows; 605x1098x1300, and width and length on the bottom as follows; 705x1098



Figure 7.2: Pictures of the enclosure frame with all the internal elements in place.

Additionally, a small plate was fashioned in order to mount the Raspberry Pi perpendicular to the rear wall instead of against it to allow for the Ethernet port to be exposed from the back. This was not specified in the design but was added to reduce additional costs to allow for the Raspberry Pi to connect to the internet via CAT5 cable. A piece of wood was cut, from a sheet of compressed chip board, for the shelf and was included in the overall assembly of the project. This piece of wood was specified in the designs with width, height and length as follows; 1077x128x12 (mm).

Part Name	Cost	Quantity
Steel Bar	R 19.67 ***	1
Steel L bar	R 162.51	1
Aluminium sheets	R 1549.2 ***	1
M6 bolts and nuts	R *	42
M5 bolts and nuts	R *	8
M4 bolt, nuts and washers	R	12
Wood	R *	1
Flush Aluminium hinge x 2	R 77.00 **	1
Hasplock 75mm	R 81.00 **	1

* Obtained from UCT ** Obtained from a hardware store *** Estimated from the amount of material used and that ordered through UCT.

Table 7.2: Bill of materials for the enclosure.

7.3 PCB Assembly

Multiple PCBs were assembled, the process of this assembly for the Pi HAT, MCU module and LCD breakout board have been documented below. The PiHAT and MCU module PCB were manufactured through Trax. As most of the components selected were selected for the reason that they would be readily available at UCT, the acquisition of all the components was quick. This made assembly begin as soon as PCBs arrived while also reducing the external costs of the project.

7.3.1 Pi HAT

Once all parts had been acquired, the assembly of the Pi HAT began. First, the ATX connector was soldered on and the voltages on important pins, like the power pin for the Raspberry Pi, were

measured with the power supply plugged in. After the correct voltage levels were observed, the remaining components were soldered as specified in the design in subsubsection 6.6.1. A Picture of the finished PCB can be seen in Figure 7.3. As can be seen in the picture, some components were not soldered as they were not necessary. The $120\ \Omega$ termination resistor was left out as the communications worked without it. The power resistors, R3 to R5, were left out as the PSU worked without them. These resistors were a precaution in case the PSU needed to be tricked to supply a regulated voltage. A bill of materials for the PCB has been listed in Table 7.3, including those excluded from the build for the purpose of this report.



Figure 7.3: Pictures of the PiHAT PCB by itself, on the left, on the Raspberry Pi, on the right.

Part Name	Cost (Unit Cost)	Quantity
PCB	R *	1
Molex MINI-FIT JR. Series 5556 20 way	R 18.30	1
SN75176B	R 8.97	1
6 way 2.54mm header friction lock	R 3.64 *	1
Red LED 1206	R 0.94	1
805 150Ω	R 0.94 *	1
20 way 2.54mm socket	R 22.71 *	1
M3 metal standoff	R 5.40 (R 1.80)	3
Reed switch	R *	1
SPST Rocker Switch	R 6.13	1
805 120Ω	R 0.94 *	1
Axial Resistor 15 Ω 2W	R 5.76 (R 1.92)	3

* Obtained from UCT

Table 7.3: Bill of materials for the Pi HAT.

7.3.2 MCU Module

The assembly began with soldering the STM32F0 microcontroller. The IC was then tested for heat or static damage by connecting it to power without any programming or external oscillator. Next, the external 8Mhz oscillator was attached with the stabilizing capacitors. Again, the microcontroller was tested and the code was uploaded to test proper operation. After successful tests were run, the remainder of the components were soldered to complete the PCB. Six MCU module boards were ordered in total. Initially, three were soldered for testing purposes and for further debugging of the code. The idea was to test the code with more than one module on the system, to check for irregularities and problems with the code, as only one prototype on the system had been tested prior to assembly.

There were two configurations made. One for the primary device at address 1 with an LCD and 2 without the LCD plugins. The plugins entailed a 4 way terminal on the “EN”, “MS1”, “MS2”, “MS3” through holes and a two way terminal for the “STEP” and “DIR” through hole. Those through holes mentioned related to the stepper module for future expansion and yet to be designed delivery mechanisms. A picture of both versions built for this report can be seen in Figure 7.4. A complete bill of materials can be seen in Table 7.4. The remaining three PCBs were soldered once the code worked reliably with more than one MCU module on the bus. They were assembled as stated above.

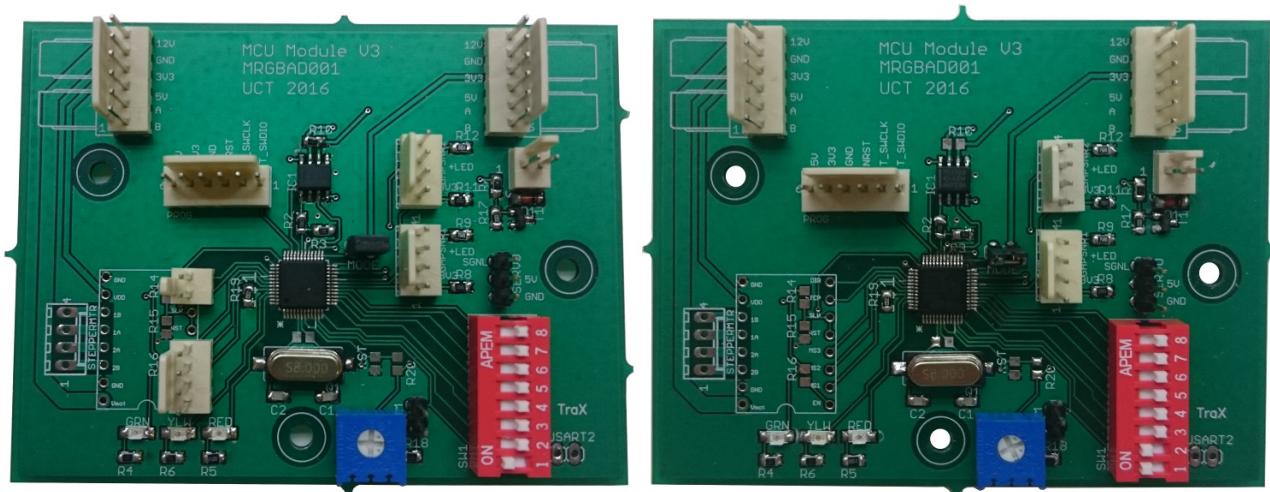


Figure 7.4: Pictures of the MCU module PCB, with the LCD plugin on the left, without the pcb plugin on the right.

Part Name	Cost (Unit Cost)	Quantity
PCB	R *	1
STM32F051C6	R 34.64 *	1
SN75176B	R 8.97	1
8MHZ HC-49 SMD	R 4.36 *	1
805 10pf	R 0.34 (R 0.17) *	2
805 1MΩ	R 0.94 *	1
General Purpose Diode SOD80 SMD	R 0.42	1
NPN 200ma rating SOT23	R 2.05 *	1
DIP Switch	R 7.70	1
805 120Ω	R3.76 (R 0.94) *	4
805 150Ω	R 1.88 (R 0.94) *	2
Red LED 1206	R 0.94	1
Green LED 1206	R 0.98	1
Yellow LED 1206	R 1.04	1
805 3K6Ω	R 0.94 *	1
805 2K Ω	R 2.82 (R 0.94) *	3
805 1KΩ	R 2.82 (R 0.94) *	3
805 10KΩ	R 2.82 (R 0.94) *	3
3 way 2.54mm header	R 0.51 *	1
4 way 2.54mm header friction lock	R 4.84 (R 2.42) *	2
6 way 2.54mm header friction lock	R 10.92 (R 3.64) *	3
2 way 2.54mm header	R 0.38 *	1
5KΩ Potentiometer	R 5.40 *	1
M3 metal standoff	R 5.40 (R 1.80)	3
805 75Ω	R 1.88 (R 0.94) *	2

* Obtained from UCT

Table 7.4: Bill of materials for the MCU module.

7.3.3 LCD Breakout Board

The LCD breakout board was made using a method called the Toner Transfer Method. This method required very little resources and could be done without professional equipment reducing the manufacturing cost of PCB making.

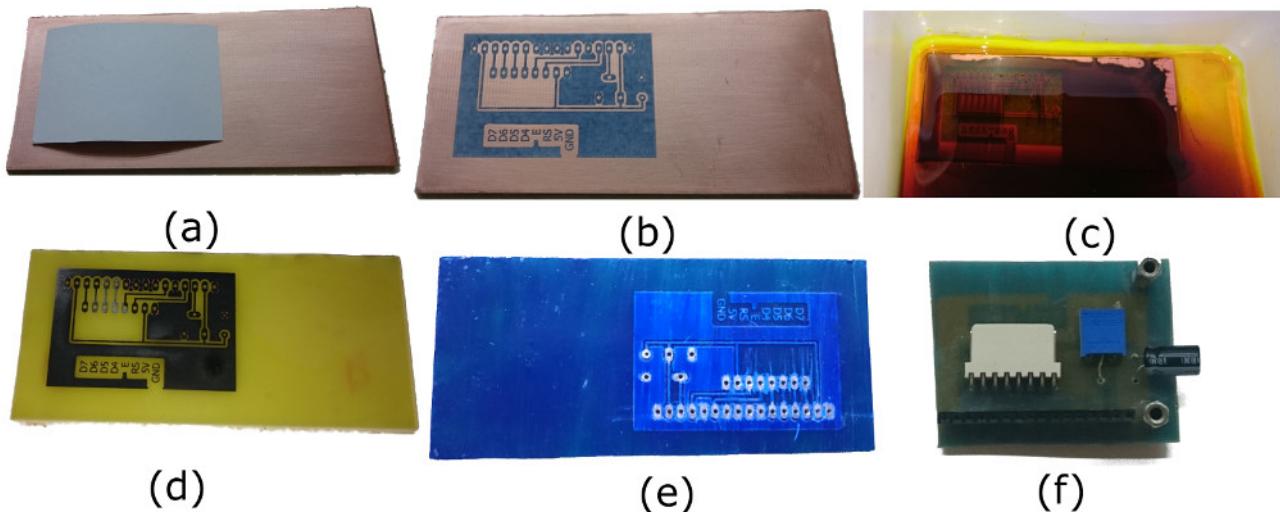


Figure 7.5: Pictures from the etching process to make the LCD breakout board.

The Figure 7.5 illustrates the assembly process at various stages of the process. First, the PCB layout was printed on photo quality paper using a laser printer. This print was cut out and placed on the piece of copper clad to be etched (a). The print was then transferred onto the copper using an iron, the result of this transfer can be seen in (b). Next, the board was placed in a solution of Ferric Chloride and a small amount of water. Left for a period of time, the copper was dissolved and the result can be seen in (d). The toner was then removed using 1000 grit sandpaper and holes were drilled where needed. The board was then painted with UV treated solder mask which can be seen in (e). Finally, the board was cut to size and the components were soldered on (f). A bill of materials can be seen in Table 7.5.

Part Name	Cost (Unit Cost)	Quantity
20cm ² of copper clad	R 6.96	1
16 way 2.54mm PCB Socket	R 15.15	1
16x2 Topway LCD (LMB162ABC)	R 98.86 *	1
1K Trimmer Potentiometer	R 6.75	1
220uf Capacitor	R 4.08	1
8 way Molex	R 9.35	1
M3 metal standoff	R 3.60 (R 1.80)	2

* Obtained from UCT

Table 7.5: Bill of materials for the LCD breakout board.

7.4 Delivery Mechanism Assembly

The acrylic glass was ordered to begin assembly of a delivery mechanism. The acrylic glass was cut as per one of the templates in Appendix F Templates & Drawings for Assembly. The roller and IC guide were 3D printed using PLA. Once all the parts had been acquired, assembly could begin.

Bolts and washers were placed in the designated holes. The next step was to solder wire leads onto the IR LEDs and phototransistors. A pair of IR LED and phototransistor, with wire leads, were glued into the designated slots making sure to line them up with the intended component to minimize false readings. The sensors were lined up in the IC guide by placing a tube in the guide and placing components in the tube to determine the optimal slot, of three, to use. The Vibration motor was then placed inside the cavity made for it and leads soldered to it. Finally, the servo horn was lined up with the foothold. A small amount of glue was used to secure it. The finished product for one of the IC guide and roller can be seen in Figure 7.6.

The housing structure was assembled next. The structure was assembled according to the design and drawings in subsection B.5 using Tensol adhesive to secure the structure. The servo motor was placed inside the housing structure during assembly. The 3D prints that were assembled and described above were then fixed to the housing structure with bolts, nuts and a screw for the servo horn. Sockets were attached to the end of the leads to enable them to be connected to the MCU modules. The final product of this assembly can be seen in Figure 7.7. In total, six delivery mechanisms were assembled, one for each design.

7.5 Overall Assembly

Once all individual modules and the enclosure had been assembled, the final product of the research could be assembled. This section will present the methods used to assemble all the components of the project together to complete the Component Vending Machine.

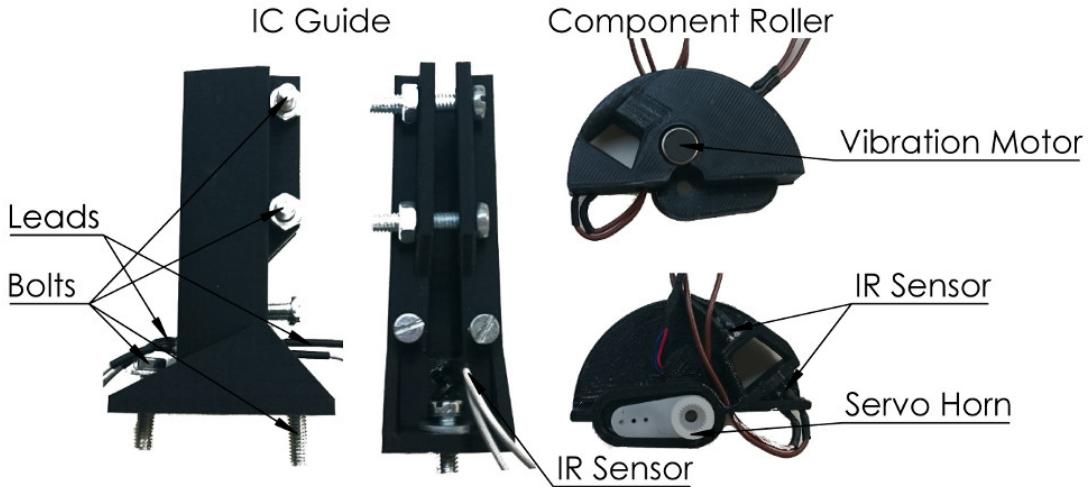


Figure 7.6: Picture of the 3D printed IC guide and roller after assembly.

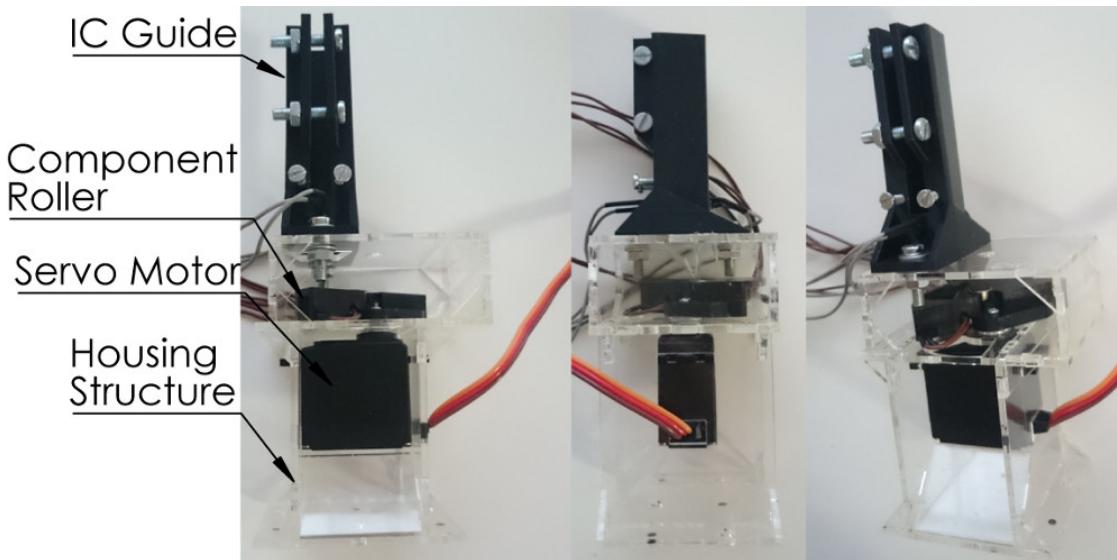


Figure 7.7: Picture of the housing structure for the delivery mechanism.

Initially, the shelves for all the delivery mechanisms were prepared. A drill template, which can be seen in subsection F.4, was made in order to make holes for mounting each delivery mechanism. Included in this template is a template for the size of the a wooden plate that would serve to hold the PCB vertically on the plank. This was done, as specified in the design, to increase the density of the delivery mechanisms allowed on one shelf. Using this template, holes were drilled in the plank and plates were cut from the same wood the shelf was. Holes were drilled in the wooden plates, using the PCBs as a template, to mount the PCBs using metal standoffs. Once all the necessary holes were drilled, the six assembled delivery mechanisms were attached using M4 bolts, washers and nuts. The leads for the motors and sensors were cut and fitted to sockets so they could attach to the PCBs. A test PCB was made to help with wiring of the sensors to correspond with the terminal on the PCB, a picture of its schematic and physical realization can be seen in Figure 7.8.

Once all the connectors had been attached, the PCBs were attached to the plates and the plates were screwed on perpendicularly to the shelf using wood screws. This allowed them to be stacked one behind the other. Wires were made to connect the PCBs together in a linear fashion, ie daisy chaining from one to the next. A picture of the shelf with all the delivery mechanism and PCBs in place can be seen in Figure 7.9.

Part Name	Cost	Quantity
acrylic glass 311.85cm ²	R *	1
3D Printed parts	R 20.00 **	1
IR LED + Phototransistor sensor	R *	2
M4 bolt and nut	R *	4
M4 washer	R *	4
M3 bolt	R *	2
DS-339MG Digital servo	€ 8.04 ***	1

* Obtained from UCT ** Estimated by weight *** Pricing from Hobby King, Obtained from UCT

Table 7.6: Estimated Bill of Materials for the delivery mechanism.

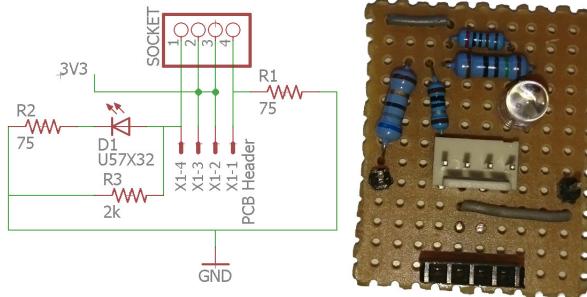


Figure 7.8: Picture of the sensor tester circuit schematic on the left and the physical circuit on the right.

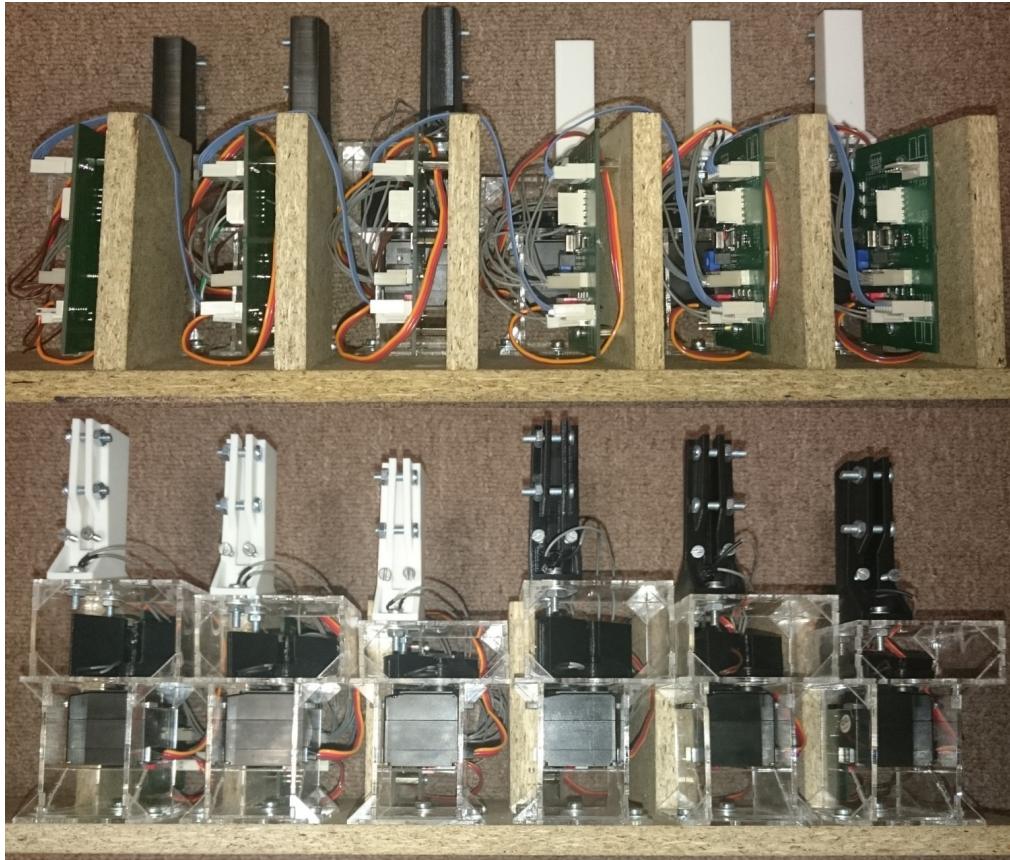


Figure 7.9: Picture from the rear of the shelf, top image, and the front, bottom image.

Next the cover for the RFID tag reader antenna was assembled using the template in Appendix D, sheet number 6 and 7. The bottom plate and middle plate were glued together first. Then bolts were

placed in the cavity, through the designed holes. The antenna was then placed in the center of the middle plate and the top plate was glued in place to hold the antenna and bolts in the cover. This was left to set and a picture of the finished product can be seen in Figure 7.10.



Figure 7.10: Picture of the back and front of the RFID antenna cover.

The RFID cover and LCD module were then mounted on the Front panel door with bolts and nuts to hold them in place. A wire for the LCD module was fashioned with sockets that would enable it to connect to the first MCU module. The RFID module was connected to the antenna and a USB cable was connected and secured to the case with epoxy to prevent it from detaching. The Raspberry Pi and computer PSU were attached next. The Raspberry Pi was attached to the small sheet that put it perpendicular to the rear wall and the PSU attached directly to the rear wall using screws. The PSU was connected to the Pi HAT and the USB from the RFID cover was attached to the Raspberry Pi. The power switch for the Pi HAT was mounted in the wall and the terminals were soldered. The magnetic switch for the Pi HAT was mounted on the frame of the door using double sided tape. Wires were attached from the Pi HAT to the magnetic switch. The shelf, holding all the delivery mechanisms, was placed on the shelf holder in the enclosure and wired to the Pi HAT.

7.6 Additional Notes and Final Assembly

This section will serve to present implementations that were not in the designs.

The sensors for the DIP 20 IC socket were giving false readings during testing and it was found that the components were falling too low. This was only a problem for the DIP sockets and the DIP 20 specifically, as it had a slot in the middle of the component which allowed light to shine through. To help mitigate false readings, a small slice of perfboard was sanded down to place under the bucket when loading components. This helped mitigate false readings. The same was done for the DIP 20 IC dispenser as, it was the most unreliable dispenser. The unreliability of this dispenser will be discussed further in subsection 8.3.

A handle was placed on the door of the enclosure to give an alternative for people pulling on the lock and possibly breaking it. This was done due to the door sagging on the open end, meaning the door needed to be propped up. This was done by placing two holes in the door, and two bolts on the frame which the holes in the door would be propped on to prevent it from sagging when closed.

Finally, the enclosure was painted as can be seen in the pictures of the final assembly. The enclosure door was removed along with the lock, hinges and handle. All the internal components were also removed before painting. Newspaper was laid out on the floor in a well ventilated room. All the walls and the door of the enclosure were spray painted. A layer of primer was first applied to the panels followed by a coating of glossy black. Once all sides were completed, the door was reassembled, with the LCD, RFID cover and module, lock, handle and hinges. The door was then fitted to the frame to make the enclosure. Finally, wires for the LCD module to connect to the first MCU module,

the magnetic switch and the Pi HAT to connect to the MCU modules were fashioned.

The final product can be seen in Figure 7.11, Figure 7.12 and Figure 7.13.



Figure 7.11: Picture of the finished Component Vending Machine from the back, front and with the door open.

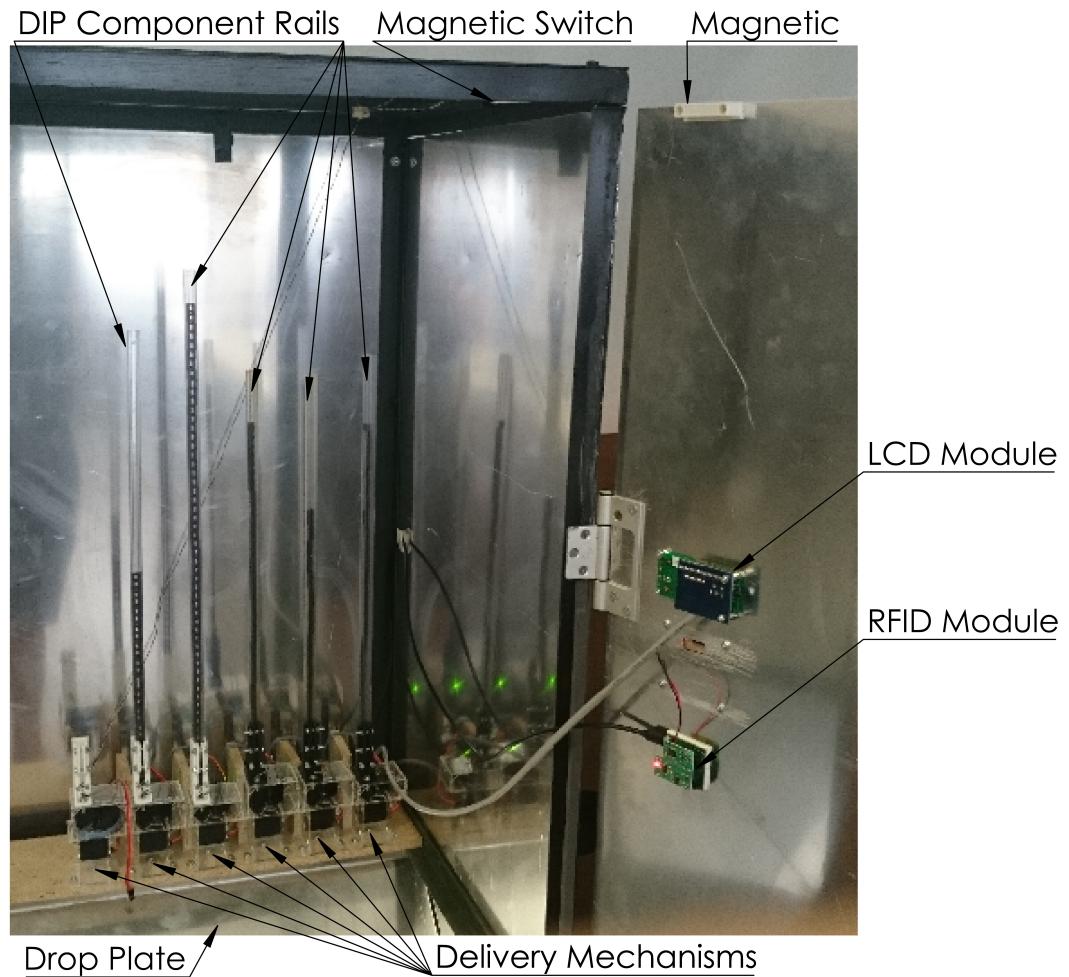


Figure 7.12: Labelled picture of the interior of the Component Vending Machine and its components.

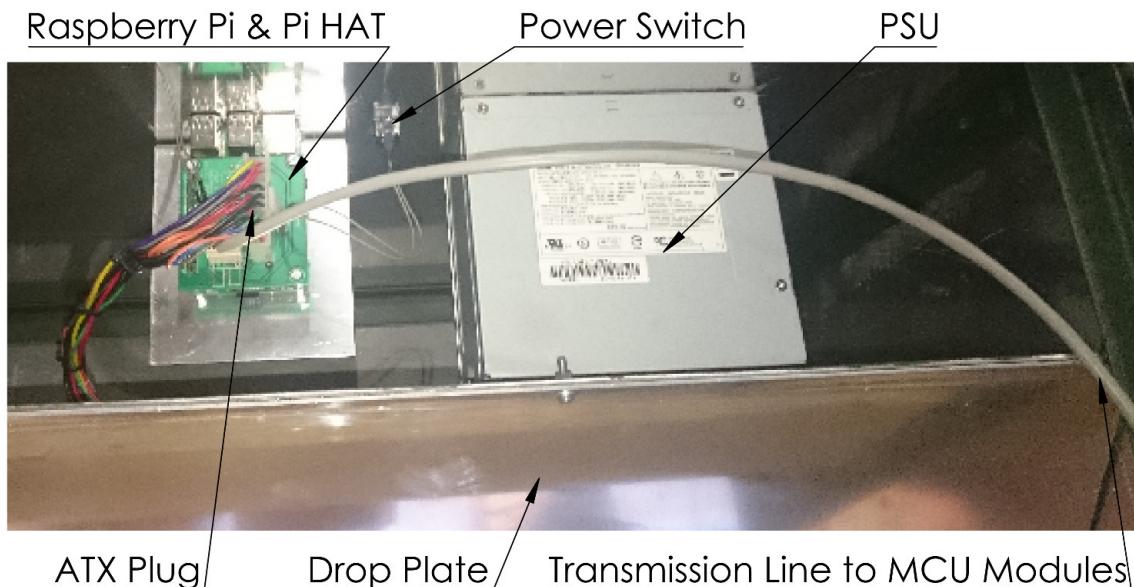


Figure 7.13: Labelled picture of the Raspberry Pi and PSU situated behind the sloped plate and below the wooden shelf.

8 Results and Discussion

Results from various tests were recorded during the process of the project. Some results have been stated previously, but will be discussed in more detail in this section. The three major categories of tests were; Prototyping tests, Comparison test and finally, Final build stress tests. These tests and results will be presented in this section. Also included in this section, will be results from tests done on the communications and final operations test results with the full system assembled.

8.1 Prototyping Tests

The prototyping tests were done in order to validate the design to make sure it was a viable design. This was done to ensure an ineffective design was not used for the final build. The tests were structured in a way to produce comparable results. Each Dispensary test was done with 100 or more individual dispenses to produce a sample to grade the reliability of the design. The actions that were recorded were; a “Success” meant the dispensary action completed without an error, a “Half Load Error” which meant the component to be dispensed caught on the lip of the bucket, a “Full Load Error” meant the component to be dispensed caught halfway down the bucket while loading and a “Release Error” meant the component to be dispensed was caught in the bucket while trying to release. The results can be seen in Figure 8.1. Videos of the tests can be seen at [31, 32, 33, 34], or on the accompanying CD named “Video 1”, “Video 2”, “Video 3” and “Video 4”.

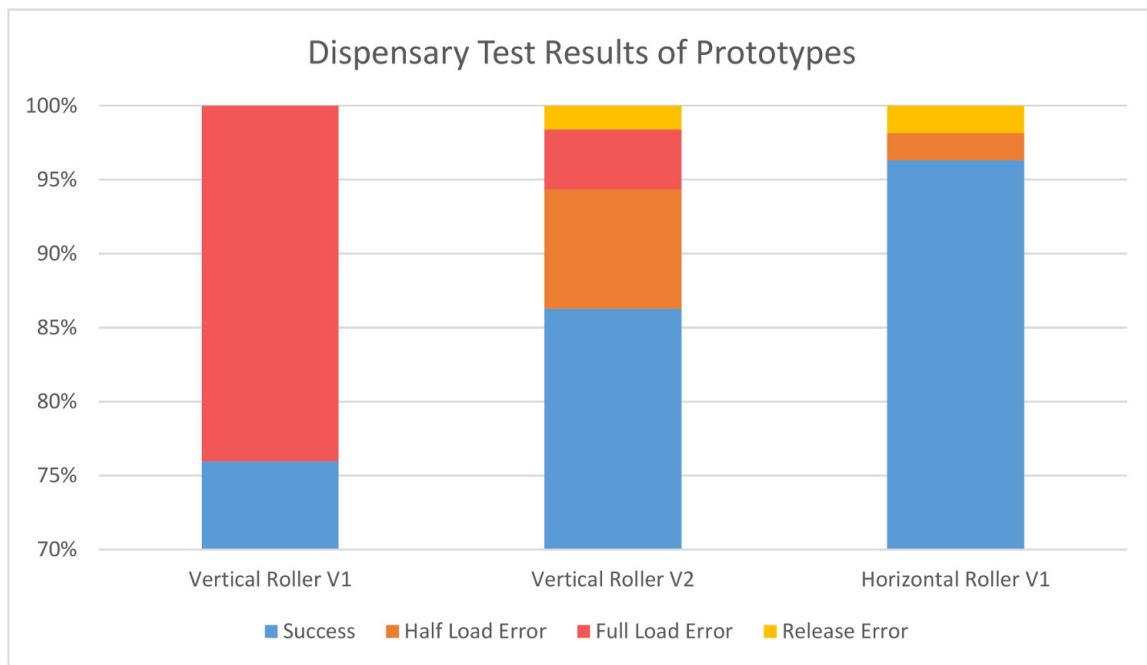


Figure 8.1: Graph showing the results of the prototyping tests done during the design phase of the project.

Comparing results, one can see that with each version of the roller, the reliability rating improved until it met the required 95% rating or above. A major failure in the earliest version of the roller was the full load error. This was a catastrophic failure which caused concern as it could damage components and was present in both versions of the vertical roller. Even with sensors, this problem was present, due to the design of the roller. This error was mitigated in the horizontal roller while also greatly reducing the chance of other failure modes, leading to the choice in using it as the final design.

8.2 Comparison test

A comparison test was done to compare the reliability of the horizontal roller with a vibration motor and without a vibration motor. This was done to validate the use of a vibration motor to assist loading and releasing of components in the bucket. Each test consisted of 105 individual dispenses to get a rough sample of the reliability under each circumstance. The same actions as the prototype testing were used to determine failure modes. The idea was that a vibration motor would help increase the reliability rating of the delivery mechanism. This test was set out to test this theory. The results of the test can be seen Figure 8.2. A video of the test can be seen at [37, 38], or on the accompanying CD names “Video 7” and “Video 8”.

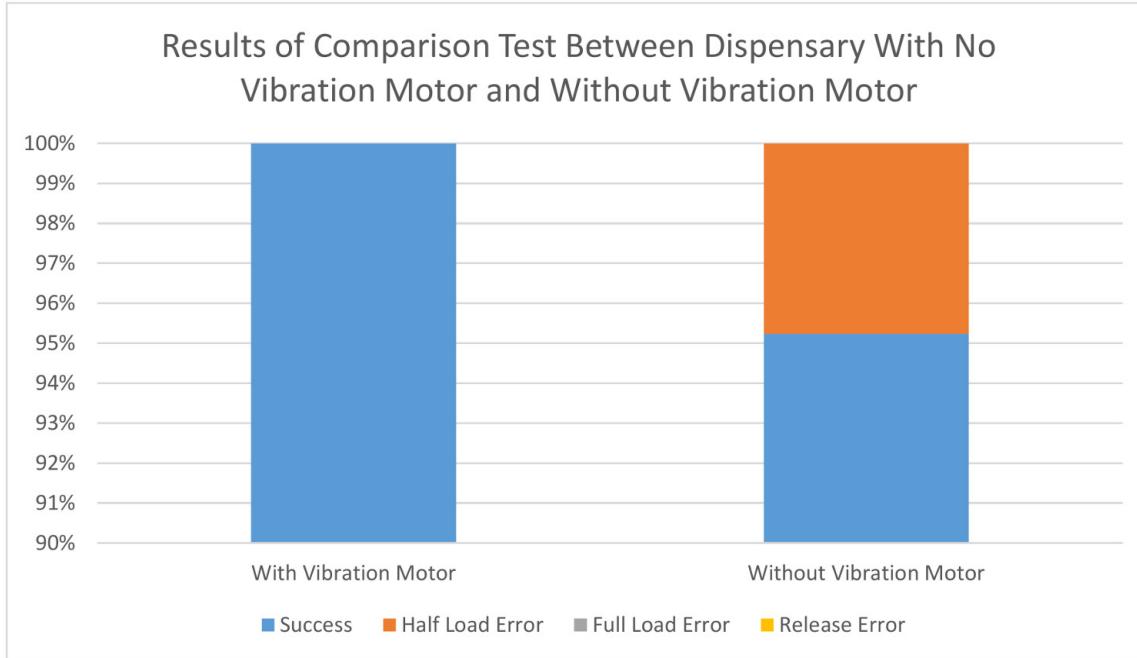


Figure 8.2: Graph showing the results of the prototyping tests done to compare the reliability rating with and without a vibration motor.

As can be seen in the graph, the run with the vibration motor was more reliable than without a vibration motor, as expected. The vibration motor helped with loading and releasing, increasing the reliability to 100%. This resulted in the design on the delivery mechanism to be concluded, as it proved the design was successful.

8.3 Final Build Stress Tests

Stress tests were done on the assembled delivery mechanism to validate that they worked as intended. The test was set up to be repeatable on future delivery mechanisms to validate them before deployment. The tests entailed 20 rounds of 10 components dispensed per round. This amounted to a total of 200 individual dispenses. This was done in order to thoroughly test assembled delivery mechanisms before they were deployed in the White Lab Component Vending Machine. The MCU board was programmed to reduce external errors by dispensing all 10 components with a manual switch of the eighth address bit on the DIP switch. Only three failure modes were recorded, as a half load was no longer a focus of the test. The actions recorded were; a “Success” which meant the dispensary action was completed without an error, a “Load Error” which meant the component to be dispensed failed to fall in the bucket properly, a “Release Error” which meant the component to be dispensed failed to vacate from the bucket when being released. The results from the test done on the assembled delivery mechanism

can be seen in Figure 8.3. A time-lapse video of the testing can be seen at [39], or on the accompanying CD named “Video 9”.

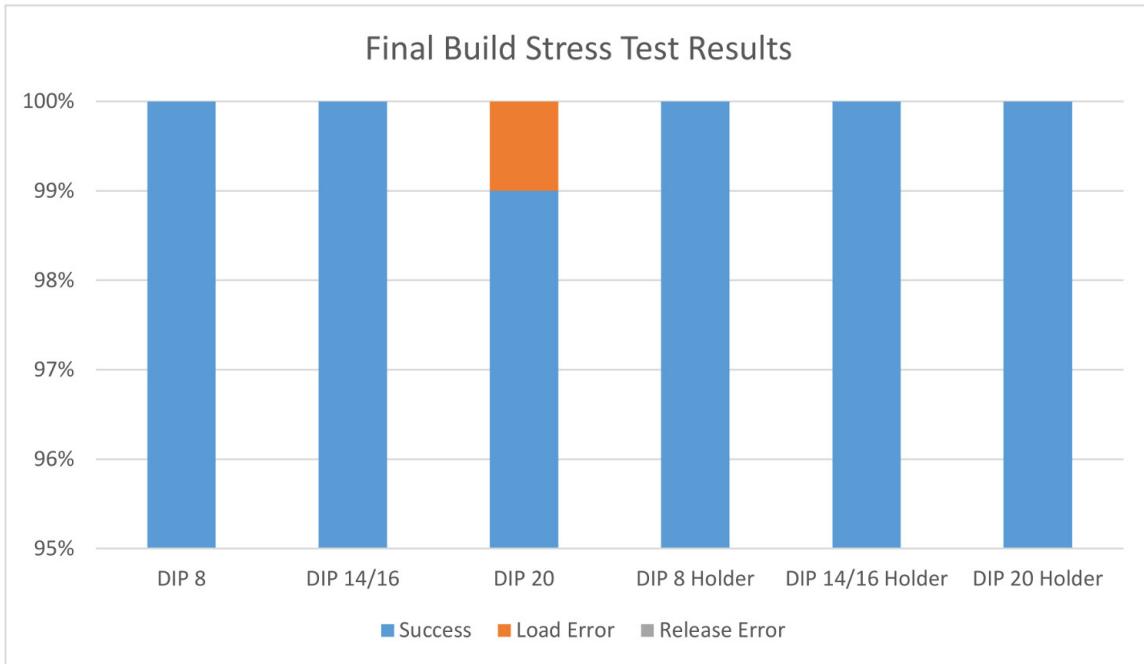


Figure 8.3: Graph showing the results of the stress tests done on all the assembled delivery mechanism for the final build.

As can be seen in the results, all the delivery mechanisms displayed a reliability rating that exceeded expectations and requirements. All achieved a reliability rating of 100% with the exception of the DIP 20 delivery mechanism, with a rating of 99%. The 99% reliability rating of the DIP 20 delivery mechanism was due to the height of the bucket and the component, to be loaded in the next cycle, catching on the filleted slope of the bucket.

8.4 Communications testing

The communications between the devices was tested thoroughly to make sure the system would not fail because of a communications error. While prototyping, tests were done on communications to observe noise on the transmission lines. When using all sensors and motors, the system produced enough noise to corrupt the data when standard UART was used. Take note of Figure 8.4. The image was captured from a logic analyzers output after being connected to the transmission lines and observing the data over the lines.

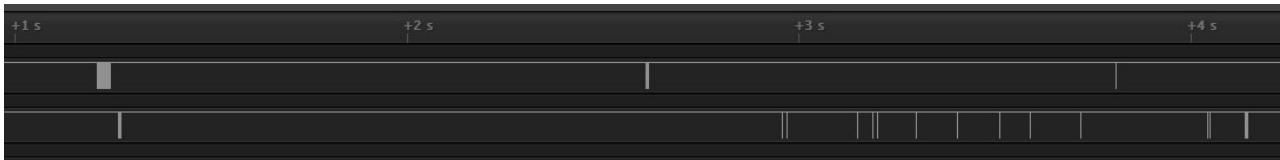


Figure 8.4: Common mode noise observed on the transmission lines.

One can see a message was sent in between the 1 second and 2 second mark in this picture. Another message can be seen in between the 2 second and 3 second mark. This was the expected result. In between the 3 second mark and the 4 second mark, there were abnormal spikes on the transmission line corrupting the data, preventing proper communications. This test was done after communications failed during prototyping leading to the expected need for a more reliable communications protocol.

Test were done with the RS 485 bus transceiver chip and the communications worked as expected. There were problems with communication, but it was found that the limitations of the Raspberry Pi and its' ability to interact with the RS 485 bus transceiver. This was found during debugging of the final assembly. It was isolated to the Raspberry Pi, by testing communications with the MCU modules alone and they were able to communicate with each other flawlessly. After further debugging, a solution was found by changing the way the code interacted with the RS 485 chip, allowing for reliable communications.

8.5 Final build Results and Discussion

The final build as detailed in subsection 7.5 was tested to make sure it operated as expected. The tests that were done, were repeated orders and dispensary requests. After some debugging, it was found that one of the MCU modules were faulty and was taken off the chain. This proved to fix functionality enabling the device to operate properly. Although there were some changes needed to the code base in order to make this so, this was mainly due to problems with communications handling. After these issues were fixed, the Component Vending Machine operated as expected. An excerpt of the log of the Master Program can be seen in Appendix G. This log shows majority successful dispensary actions with some corrupted communication warnings. The corrected code accounted for these unsuccessful communication attempts by attempting to resend the message which would then successfully complete the dispensary action. A video corresponding to this log can be seen at [40] or on the CD that accompanied this report called “Video 10” along with the log file named “Master.log”

9 Conclusion

The aim of this research and report was to investigate the possibility of fabricating a White Lab Component Vending Machine. Designs and construction requirements were set out to define what the Component Vending Machine needed to do and how it should be done. The function of the Component Vending Machine was the reliable dispensing of individual components so as to deliver a 24 hour 7 days a week, store that students could request components from. The system needed to be able to identify students by reading their student cards and interacting with an online store. This was so that students could easily interact with the Component Vending Machine at any time of the day or night. Finally, an enclosure was designed that would be able to house all the parts that made up the internals of the Component Vending Machine while being able to present the components to the students ordering the components.

Designs were theorized and prototypes were made for each version of the design, until an optimal solution was found. The amendments over each version of the delivery mechanism were due to flaws found in the prototypes, which lead to a reliable final design. Additionally, an enclosure and PCBs were designed to work in tandem with the delivery mechanisms. Code was then designed to work on the aforementioned PCBs that would help operate the delivery mechanism and facilitate an internet connected device. After the designs were completed for the White Lab Component Vending Machine, assembly began. The assembly of the Component Vending Machine was successful for the most part with some challenges and failures. In the end, a viable device was produced for further testing.

The enclosure worked for its intended purpose. However, there were inaccuracies due to lack of skill and proper tools. Although it was not the expected outcome, the enclosure was successful enough to house all the parts needed, allowing for a working prototype that could be deployed in the White Lab.

The website functioned as expected and helped to prove the internet capability and that such a device was more than capable of connecting to the internet. This meant that the Component Vending Machine would be able to operate with a single input from the student once they had placed their orders, making it simple and easy to use. This input would be the swipe of their student cards.

With the ability to stock the machine with the component tubes, the machine would be able to stock a large quantity of components. This meant it would be able to supply students with components for extended periods of time without needing to be restocked. This also meant that the machine would be able to supply students with components overnight and reduced the possibility that it would run out of stock on any given night. If it did run out of stock, it would not be a common occurrence due to the quantity of stock the machine was capable of supplying.

The delivery mechanisms all worked better than expected, exceeding the set out reliability standard. Although the delivery mechanisms themselves were successful, the MCU modules were not all functional. This was due to problems encountered in the assembly and not the design. This meant that the design of the MCU module was successful and met the requirements set out at the beginning of the project. With the Pi HAT also working, along with all of the above, the basic functionality of the Component Vending Machine was successful. On top of this success, the Component Vending Machine was able to communicate over the internet and interact with the website to enable online ordering. Paired with the RFID tag reader, the system was able to identify student numbers and match them to orders that were placed on the online store. After matching student numbers with orders on the store, components were successfully delivered to the aforementioned student. The design worked for the intended purpose and therefore, was a successful design and implementation of the White Lab Component Vending Machine.

10 Recommendations

Although the Component Vending Machine worked as intended, there were some potential enhancements identified. The recommendations focus mainly on the build and what could be improved, after experience with assembling the Component Vending Machine. Comments on process and future implementations will also be made and are presented in this final section.

As seen in the research, Apache is fast becoming an outdated software package, with Nginx set to take the lead. This should be taken into consideration. Even though the Component Vending Machine was not expected to have a lot of traffic on the website, future demands may justify a shift to Nginx. Also, support for Apache may be reduced in the future, as Nginx gains popularity, in which case the server should be migrated to a Nginx web server.

Although The website achieved its purpose, when compared to more commercial websites it lacked many features. It was not easily upgradeable and so would reach a point where the website would be unable to meet the requirements of the Component Vending Machine. The website also lacked proper administrator tools, restricting control. It should be heavily upgraded by someone with extensive knowledge in web development. The other alternative was to use a completely different website and integrate the Component Vending Machine with such a website, which would be more capable of meeting student requirements and expansion of the Component Vending Machine and its catalogue.

A mistake was made when soldering the sensor headers on the PCBs and to keep the format of the PCBs uniform, this mistake was replicated. It did not change the functionality of the PCBs, but may cause confusion for future installers when looking at pictures. Future builds should follow the designs to reduce the possibility of mistakes that could cause damage to the sensors or PCB.

The DIP 20 IC and IC socket dispensers were the most unreliable and problematic dispensers. A fix for the issue was implemented but was not practical. The root cause for the IC dispenser was the height of the roller bucket. The bucket was too high, causing the IC to be loaded on the next round to jam up against the filleted slope as discussed in the testing. A redesign should be undertaken with follow up tests. A small height adjustment to the bucket should fix the issue. Th IC socket dispenser was getting false readings due to the slightly elevated sensor position in the bucket. Although this was only a problem on the DIP 20 IC socket dispenser, it could cause problems on the other socket dispensers as well. This issue should be monitored and where possibly be solved preemptively. The sensor for all IC socket dispensers should be redesigned so that the sensor slots are flat, not slanted for 3D printing efficiency. Also, the sensors should be as low as possible, since there was no slot at the bottom of the IC sockets. This should be tested further to find an optimal position.

The MCU modules although operated as expected, lacked capacitors on the supply. This may help with stability issues seen on the bus at times and make the modules more reliable than they already are. It is suggested that the design be amended to add capacitors on the supply lines.

Due to the limitation in scope, only DIP packages were considered. If the Component Vending Machine project is to be successfully advanced, designs for other packages should be designed and implemented. This could also help students expand their knowledge by introducing uncommon components. By doing this, students would have the opportunity to use these rarely requested components they possibly never knew about. This is a possibility, due to the nature of the current White Lab component store being behind doors for the most part. The Component Vending Machines site would allow students to see what would be behind the door and explore a wider variety of components than those they have been told about in lectures.

The enclosure, although it served its' function, was not adequate to serve as the enclosure, permanently. The enclosure was not sturdy enough, due to the way the frame was designed, caused by an oversight

RECOMMENDATIONS

and inaccuracies during the assembly process. One of the major problems, was the fact that the door sagged when open and needed to be propped up when closed. This is due to the weight of the door hanging on the frame and bending the frame, which caused it to sag on the open end. Either improvements need to be made, by strengthening the frame, or a pre-build enclosure should be bought to replace the current enclosure. A pre-bought enclosure was not considered for the purpose of this report due to the cost, that being far greater than making one out of raw materials, as was the one made for this report. This information about a pre-built frame was obtained in passing conversations with technical officers at UCT, which was why it never made its way into the report prior to the recommendations. Additionally the paint work done on the enclosure, although was adequate for the intended purpose, does not have a commercial finish. If a more professional finish were desired the paint work should be outsourced in order to give the Component Vending Machine a more professional look.

The “daisy chain” method of connecting devices could cause strain on the cables currently used. This might occur in the form of overloading them with too much current and causing catastrophic failure. Furthermore, the resistance over the wires might become too great and cause unknown errors on the bus. To prevent this in the current form, thicker inner core transmission lines should be used as the system expands. A twisted shielded pair should be considered if more units were to be added to the chain. This would help reduce noise on the bus and make the communication more reliable. A twisted shielded pair was not used for the transmission lines as the communications worked without it in the current setup of the Component Vending Machine. An alternative would be to use DIN rails in a future implementation, which might make attaching each MCU module to the system easier and more efficient, while eliminating concerns that arose from using a “daisy chain” method.

When programming the STM MCU modules, the programmer should take note of the instructions in subsection H.1. This appendix serves as a guide for the current software. The guide in subsection H.2 provides instructions for restocking and for maintenance of the Component Vending Machine. If new information is gained or the software changes, the corresponding guide should be updated to reflect the changes.

The Colour of the PLA 3D printed material affected the sensor readings. The black print seemed to work the best, as it would not reflect infrared light for the sensor, whereas the white print increased the sampled values because of the reflection. This should be taken into account for future delivery mechanism, as it would affect the sampling code. Either black PLA should be used consistently, or if a different colour is used, then calibration on the sampling code should be done.

The RFID reader proved unreliable due to its close proximity to the aluminium. It was moved further away with a wood piece as a spacer, between the RFID cover and aluminium. The RFID cover should be redone, in order to resolve this issue in a presentable manner. This could be achieved by using 2mm acrylic glass for the front cover, instead of 3mm and a second 3mm sheet on the back of the cover.

Finally, when considering future expansion of this project, it is strongly recommended that it be made open source. Having it open source makes it possible for a community to develop around the project, similar to the RepRap project [29]. Doing so would greatly increase the success of the project, possibly bringing in highly skilled designers and engineers to contribute to the project. Asking other universities to collaborate would be a good idea as well, as this would benefit their students too. Making use of Github to distribute the resources would promote this open source idea. If the project gains more popularity, a shift to an Atmel based system should be considered, as it is more widely used in open source projects. With an Atmel based system, more people might be interested in the project as it is a widely used system. Due to the success of the first assembly of the first White Lab Component Vending Machine, this project should definitely continue. All the resources in this project can be found on the Github repository at [41] and on the CD that accompanied this report.

References

- [1] Y. K. Chelvam, N. Zamin, and G. S. Steele, "A preliminary investigation of m3ditrack3r: A medicine dispensing mobile robot for senior citizens," *Procedia Computer Science*, vol. 42, pp. 240 – 246, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050914014963> [Accessed: Aug. 2016]
- [2] F. Mehboob, "A novel multi purpose robotic dispenser with database management system," in *2012 6th International Symposium on Medical Information and Communication Technology (ISMICT)*, March 2012, pp. 1–4.
- [3] C. Crema, A. Depari, A. Flammini, M. Lavarini, E. Sisinni, and A. Vezzoli, "A smartphone-enhanced pill-dispenser providing patient identification and in-take recognition," in *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*, May 2015, pp. 484–489.
- [4] Portland State University, Maseh College of Engineering & Computer Science: Electrical & Computer Engineering. Engineering parts vending machine. [Online]. Available: <https://www.pdx.edu/ece/engineering-parts-vending-machine> [Accessed: Sept. 2016]
- [5] M. G. Gusso. (August 4th, 2014) Late-night component cravings? [Online]. Available: http://www.latticesemi.com/~/media/LatticeSemi/Documents/WhitePapers/HM/ImprovingNoiseImmunityforSerialInterface.pdf?document_id=50728 [Accessed: Sept. 2016]
- [6] S. G. Patil and T. J. Gale, "Preliminary design of remotely used and monitored medication dispenser," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, Aug 2006, pp. 3616–3619.
- [7] T. J. G. V. Q. Ho and C. R. Stack, "Medication dispenser for narcotic rehabilitation patients," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sept 2009, pp. 1742–1745.
- [8] S. L. J. N. C. K. Toh and Y. O. Tan, "Three dimensional adjustable cavity for flexible singulation of multiple medications in an automated medication dispenser," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2012, pp. 347–352.
- [9] Lattice Semiconductor, "White paper: Improving noise immunity for serial interface," 2014, pp. 2.
- [10] T. K. Texas Instruments, "Application report: The rs-485 design guide," 2008, pp. 1 - 4.
- [11] A. Minor. (2007) Implementing your mcu-based system's serial uart in software. [Online]. Available: <http://www.embedded.com/design/other/4025995/Implementing-your-MCU-based-system-s-serial-UART-in-software> [Accessed: Sept. 2016]
- [12] J. G. Texas Instruments, "Literature report: Low-voltage differential signaling yields megatransfers per second with milliwatts of power," 1996, pp. 3.
- [13] IMPINJ. (2016) The different types of rfid systems. [Online]. Available: <http://www.impinj.com/resources/about-rfid/the-different-types-of-rfid-systems/> [Accessed: Sept. 2016]
- [14] IMPINJ. (2016) How do rfid systems work? [Online]. Available: <http://www.impinj.com/resources/about-rfid/how-do-rfid-systems-work/> [Accessed: Sept. 2016]
- [15] J. Pead. (Sept. 2016). Principal Technical Officer [Online]. Available e-mail: justin.pead@uct.ac.za, Message: RFID Documentation.

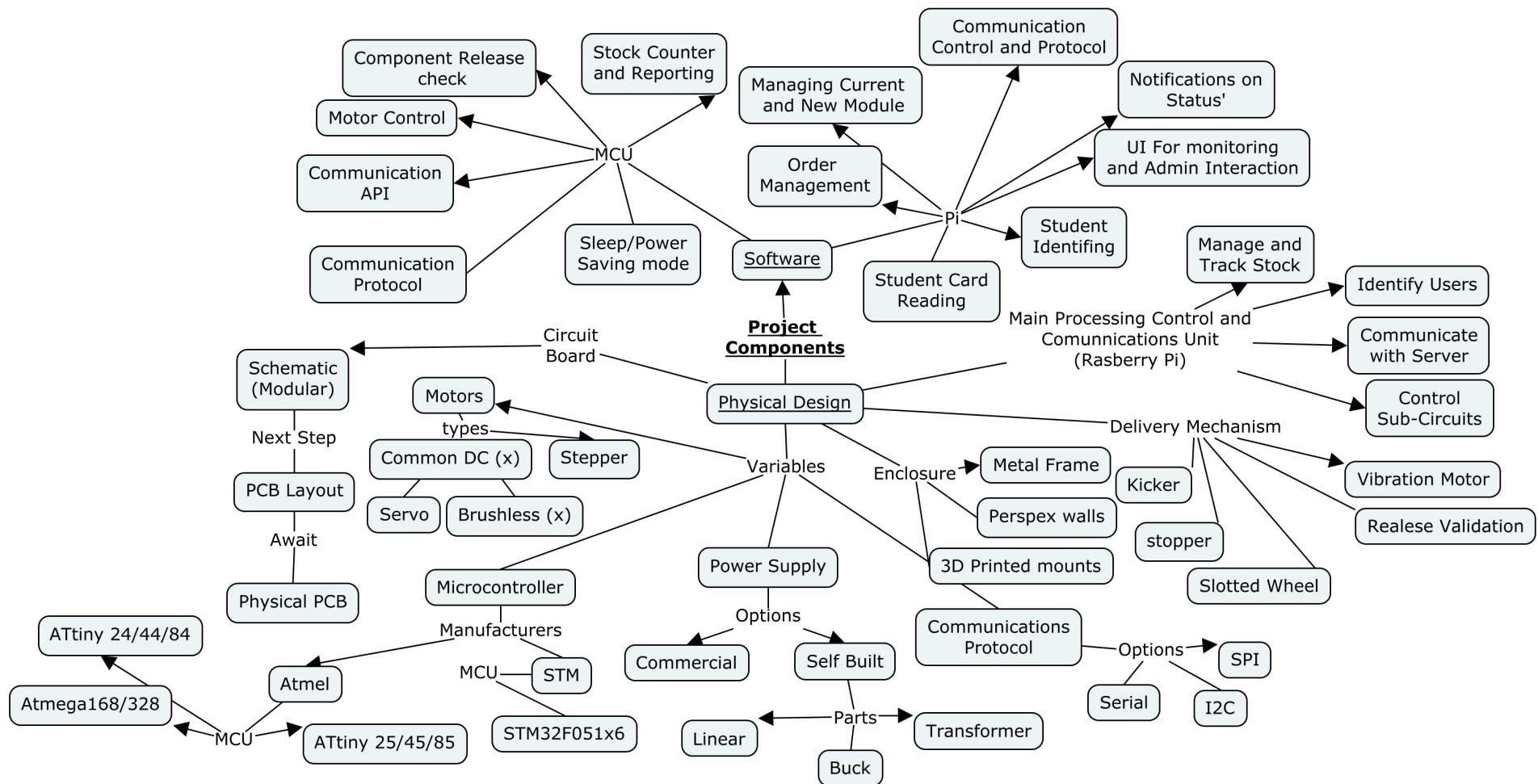
REFERENCES

- [16] Seedwiki. (2016) 125khz rfid module - uart. [Online]. Available: http://wiki.seeedstudio.com/wiki/125Khz_RFID_module_-_UART [Accessed: Sept. 2016]
- [17] Microchip. (2011) Mcp2200 datasheet. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/22228B.pdf> [Accessed: Sept. 2016]
- [18] J. Versluis and J. Versluis, *LAMP Stack for Humans: How to turn a laptop into a web server on your local network.* Jay Versluis(Self Published), 2015.
- [19] O. Garret. (October 9, 2015) Nginx vs. apache: Our view of a decade-old question. [Online]. Available: <https://www.nginx.com/blog/nginx-vs-apache-our-view/> [Accessed: Sept. 2016]
- [20] Netcraft. (2016) August 2016 web server survey. [Online]. Available: <https://news.netcraft.com/archives/2016/08/24/august-2016-web-server-survey.html> [Accessed: Sept. 2016]
- [21] P. L. Ben Laurie and S. St.Laurent, *"Apache: The Definitive Guide" 3rd Edition.* O'Reilly Media, Inc., 2002. pp. 1-5.
- [22] A. Peicevic and M. Maslac, *Apache HTTP Server introduction.* Geek University Press, 2016. pp. 4-6.
- [23] R. Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Learning Php, Mysql, Javascript, Css & Html5) 4th edition.* O'Reilly Media, Inc., 2015. pp 35-37.
- [24] O. C. and/or its affiliates. What is mysql? [Online]. Available: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html> [Accessed: Sept. 2016]
- [25] I. Corporation, “Atx specification version 2.1,” June 2002.
- [26] E. Tutorials. (September 2016) Convert atx psu to bench supply. [Online]. Available: <http://www.electronics-tutorials.ws/blog/convert-atx-psu-to-bench-supply.html> [Accessed: Sept. 2016]
- [27] Mantech. (September 2016) Mantech electronics. [Online]. Available: <http://www.mantech.co.za/> [Accessed: Sept. 2016]
- [28] H. King. (September 2016) Hobbyking. [Online]. Available: <http://www.hobbyking.com/hobbyking/store/index.asp> [Accessed: Sept. 2016]
- [29] A. Bowyer. (September 2016) Reprap. [Online]. Available: <http://reprap.org/> [Accessed: Sept. 2016]
- [30] R. C. (SA). (September 2016) Rs-online. [Online]. Available: <http://za.rs-online.com/> [Accessed: Sept. 2016]
- [31] B. Morgan. (September 2016) Vertical roller v 1 test part 1. [Online]. Available: <https://youtu.be/nNe5YIRPoKM> [Accessed: Sept. 2016]
- [32] B. Morgan. (September 2016) Vertical roller v 1 test part 2. [Online]. Available: <https://youtu.be/4hIJvzhvt88> [Accessed: Sept. 2016]
- [33] B. Morgan. (September 2016) Vertical roller v 2 test. [Online]. Available: <https://youtu.be/DLGEwFN1RbM> [Accessed: Sept. 2016]
- [34] B. Morgan. (September 2016) New horizontal roller test. [Online]. Available: https://youtu.be/_Ol1D7sp83Q [Accessed: Sept. 2016]
- [35] R. P. Foundation. (october 2016) Raspbian download. [Online]. Available: <https://www.raspberrypi.org/downloads/raspbian/> [Accessed: Oct. 2016]

REFERENCES

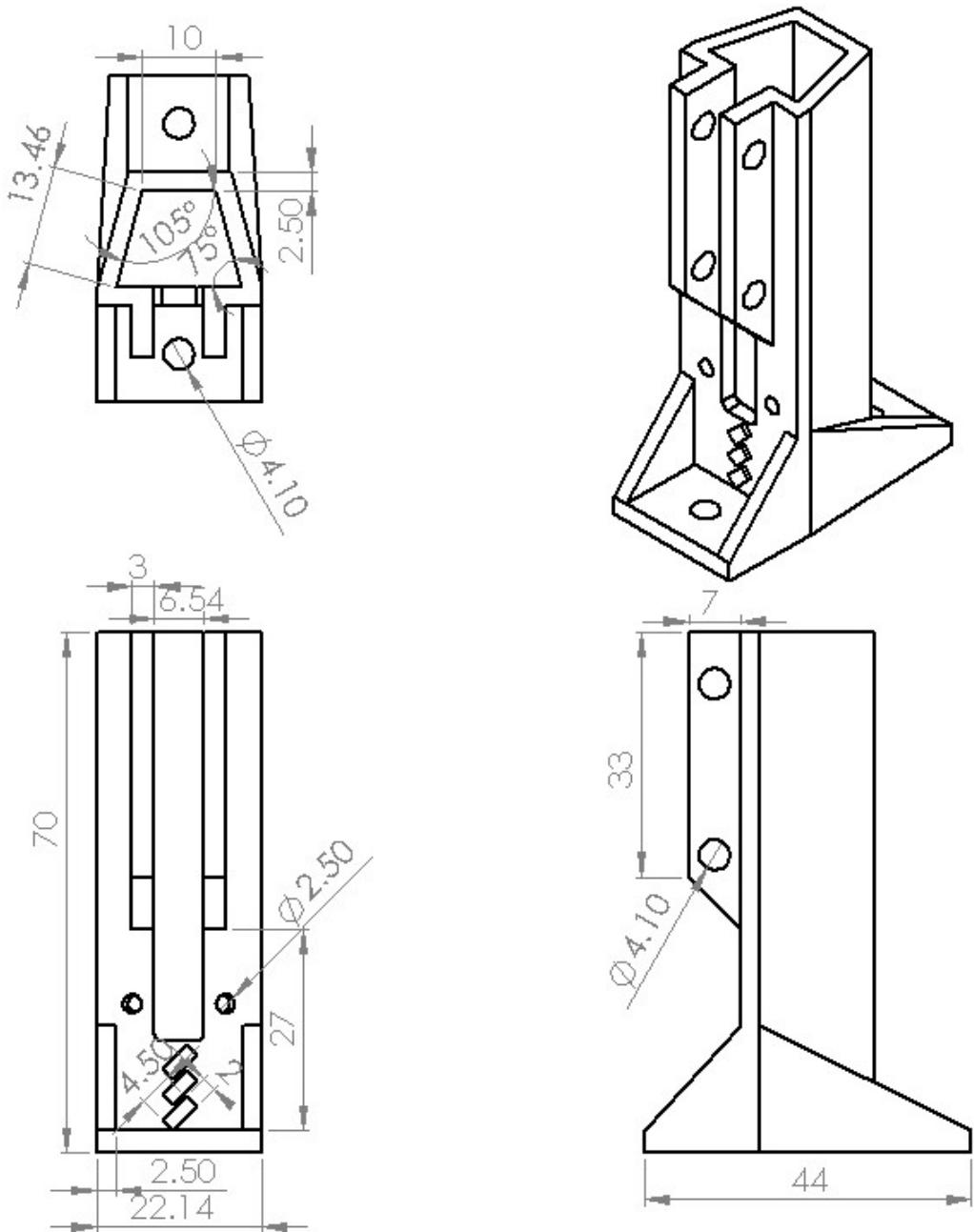
- [36] T. Davis and J. Davis. (october 2016) Win32 disk imager. [Online]. Available: <https://sourceforge.net/projects/win32diskimager/> [Accessed: Oct. 2016]
- [37] B. Morgan. (September 2016) Horizontal roller dip 8 variant test with vibration motor. [Online]. Available: <https://youtu.be/JkFqaCLOxYs> [Accessed: Sept. 2016]
- [38] B. Morgan. (September 2016) Horizontal roller dip 8 test no vibration motor. [Online]. Available: <https://youtu.be/PZypj8374xY> [Accessed: Sept. 2016]
- [39] B. Morgan. (October 2016) Final build stress tests. [Online]. Available: <https://youtu.be/CHXiJHn9fzM> [Accessed: Oct. 2016]
- [40] B. Morgan. (November 2016) Final assembly demonstration. [Online]. Available: <https://youtu.be/3u1VQo9YuXM> [Accessed: Nov. 2016]
- [41] B. Morgan. (Novemeber 2016) Thesis github repository. [Online]. Available: <https://github.com/BadenMorgan/thesis> [Accessed: Nov. 2016]

Appendix A Research Mind Map

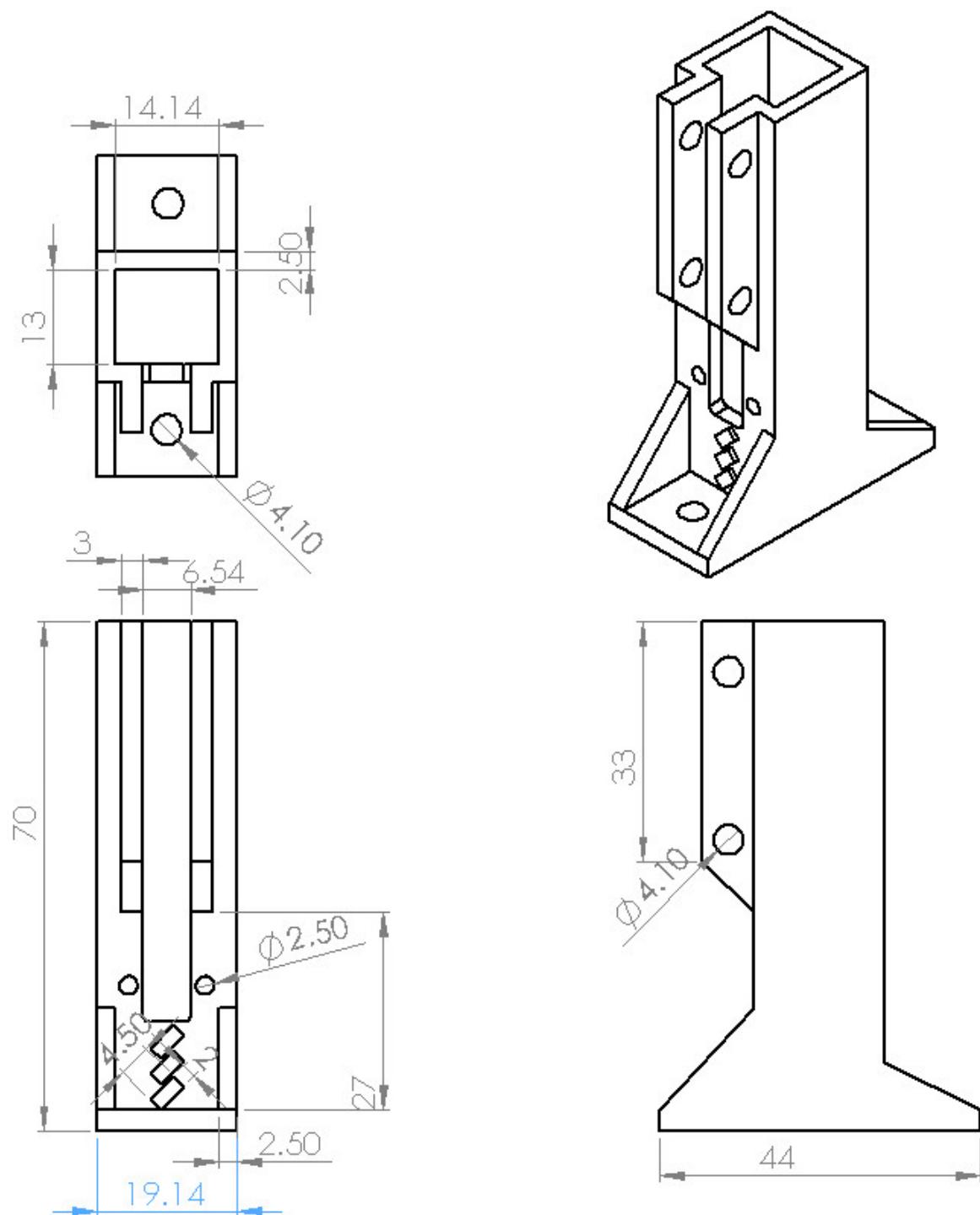


Appendix B Detailed Drawings

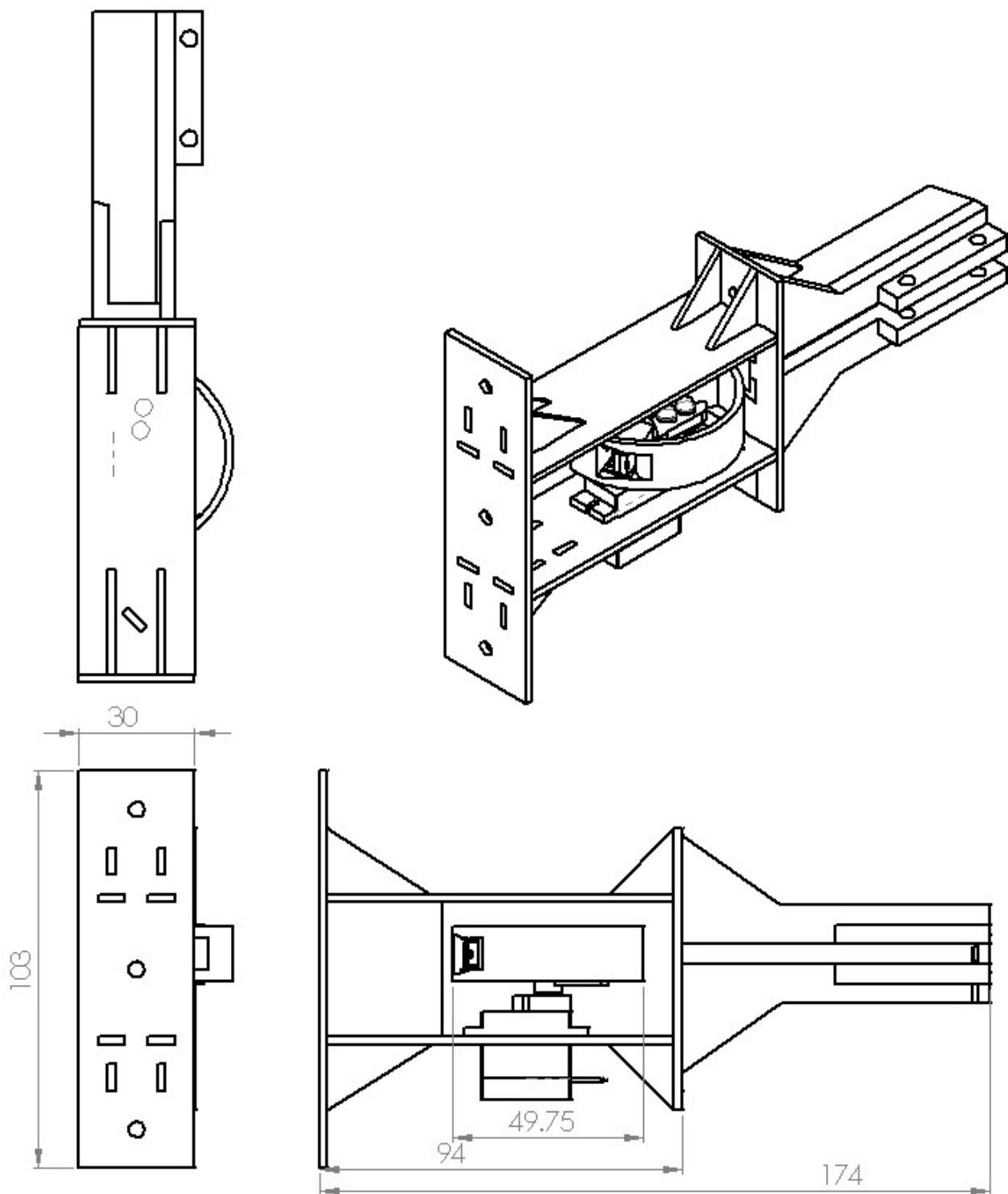
B.1 IC Guide DIP Components



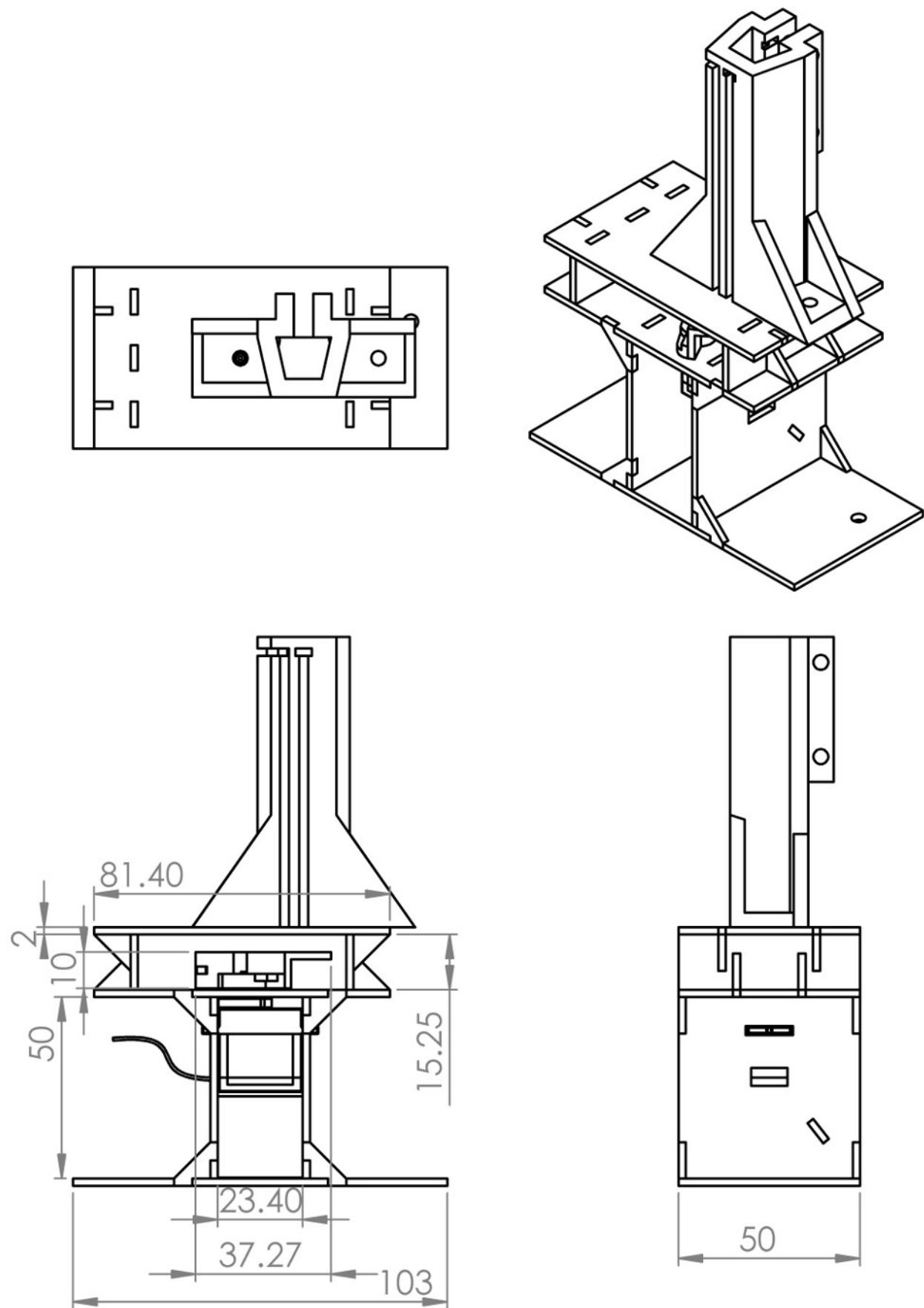
B.2 IC Guide DIP Holder



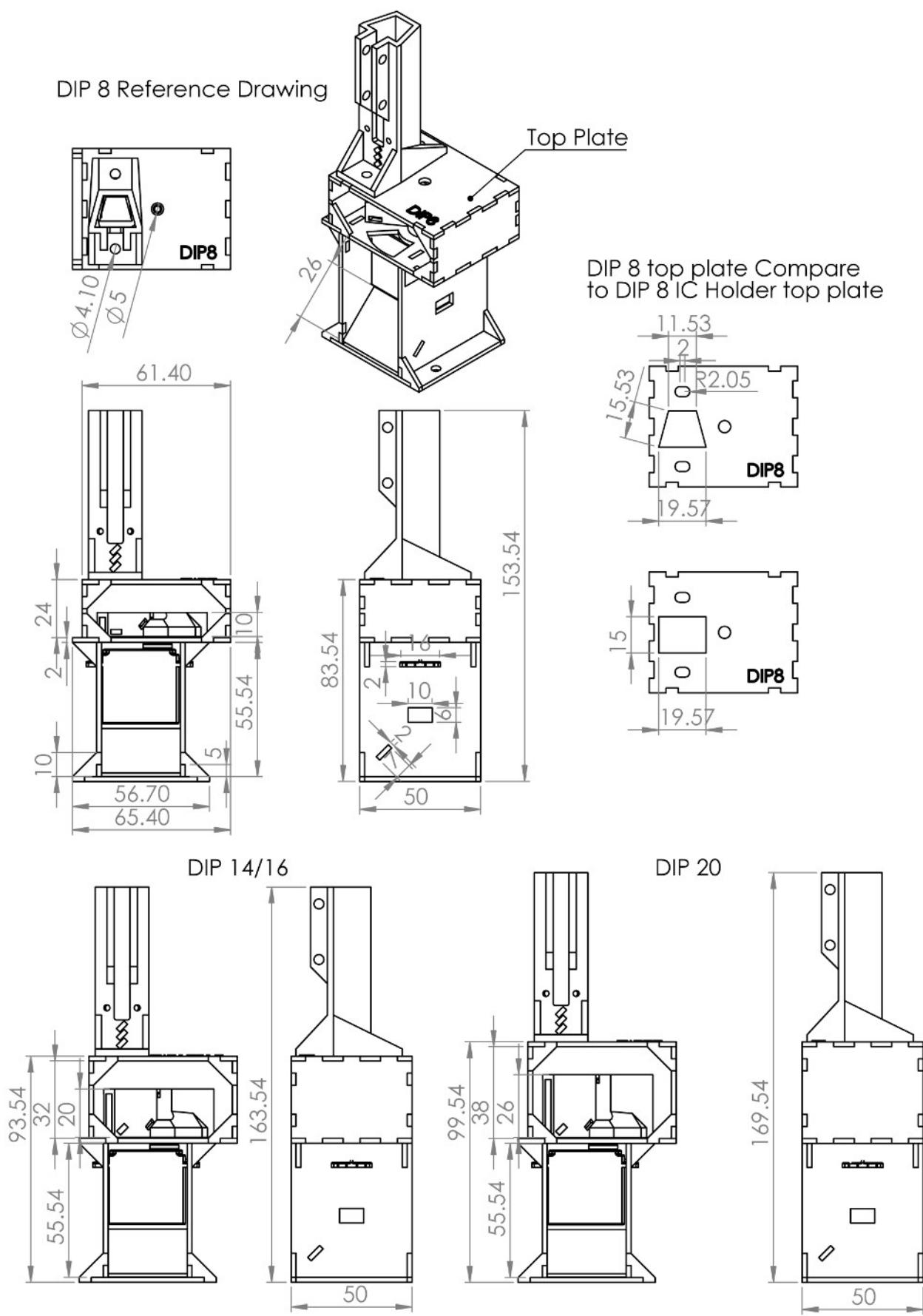
B.3 Vertical Roller & Housing Version 1 & 2



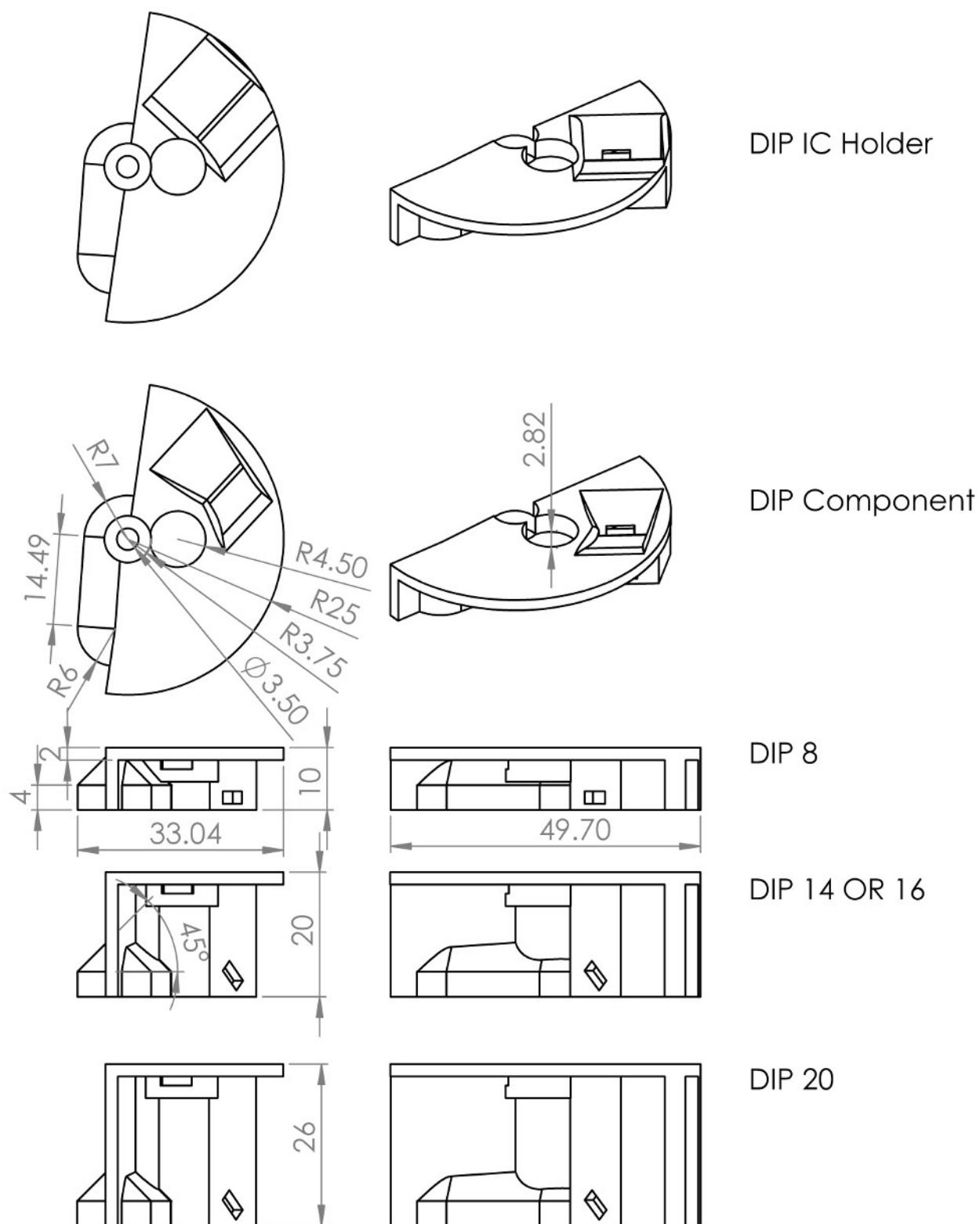
B.4 Horizontal Roller & Housing Version 1



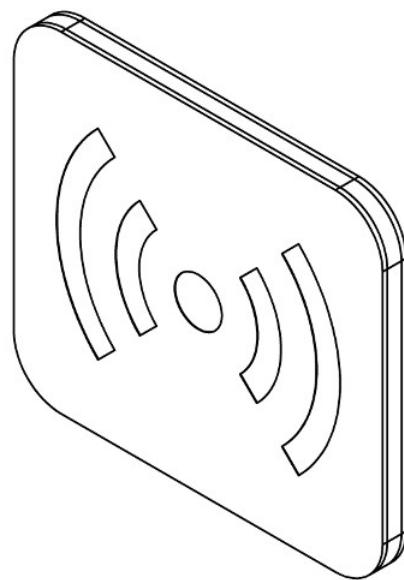
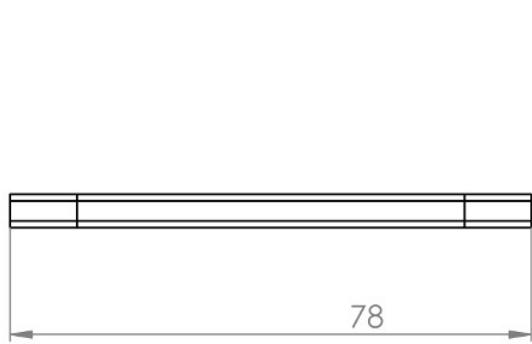
B.5 Horizontal Roller & Housing Final Version



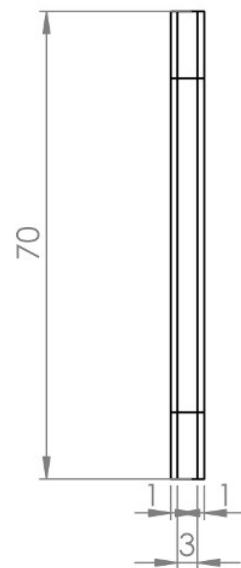
B.6 Horizontal Roller Final Version



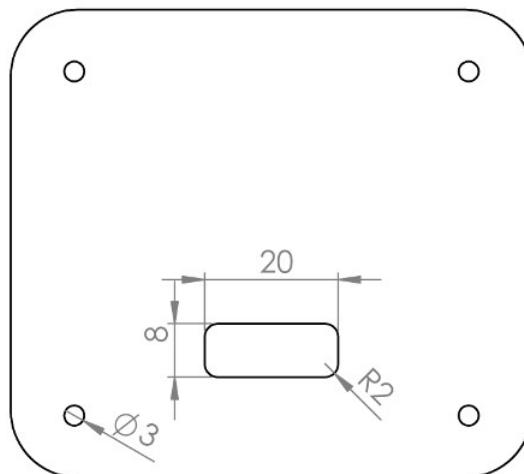
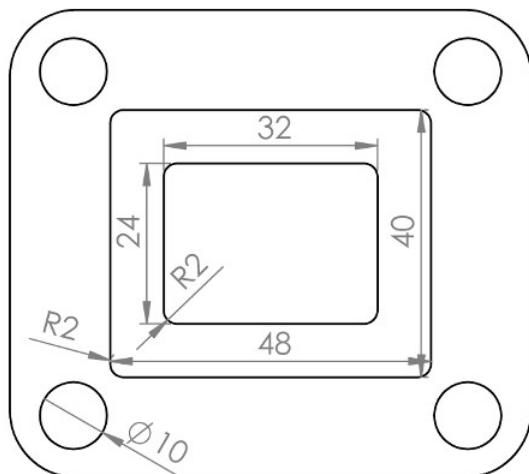
B.7 RFID Coil Cover



Middle Plate



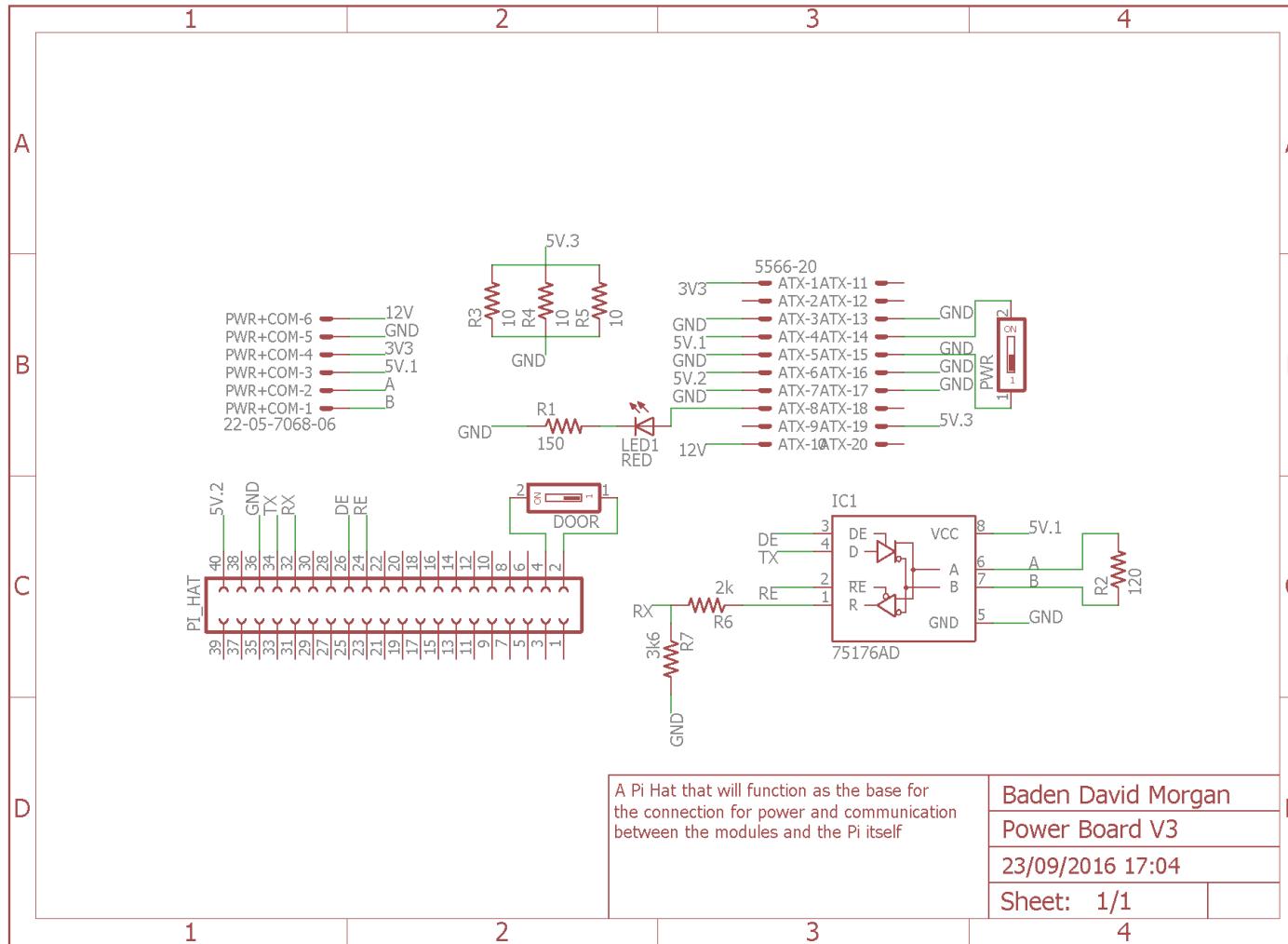
Back Plate



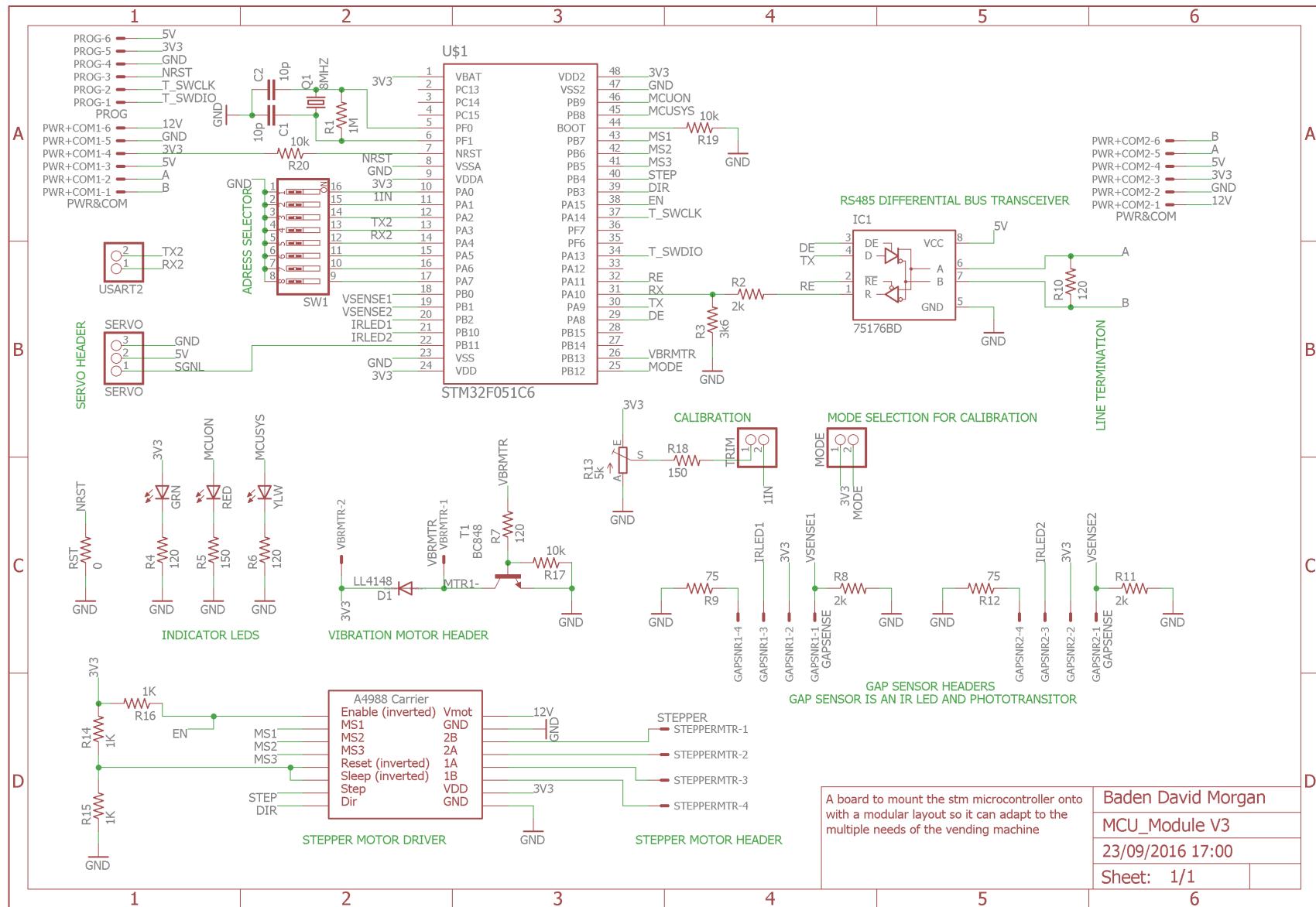
Appendix C PCB Diagrams

C.1 PI HAT Diagram

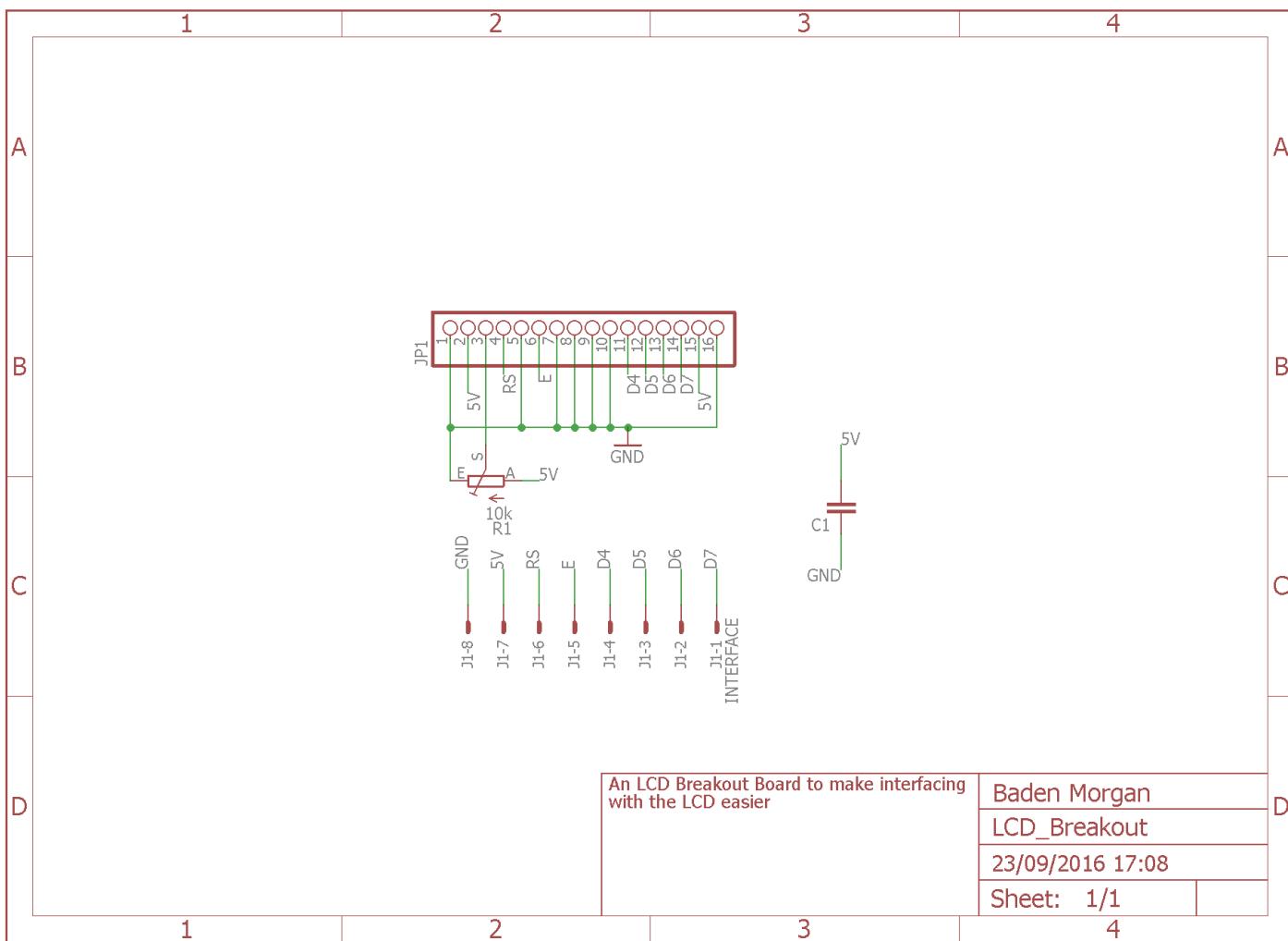
9L



C.2 MCU Module Diagram



C.3 LCD Breakout Diagram



Appendix D Material Request Forms

Aluminium Sheets with thickness 1mm

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION I
DRAWN	SIGNATURE	DATE			
CHK'D					
APPV'D					
MFG					
Q.A.		MATERIAL:	Aluminium	DWG NO.	A4
		WEIGHT:		1:20	SHEET 1 OF 1

TITLE: **Aluminium Sheets
For MRGBAD001**
Sheet 1 v2

4 3 2 1

MATERIAL REQUEST FORMS

4	3	2	1
F	1288	L beam 2mm variant	
F	1288		
F	1288		
F	1288		
E	1100		
D	600		
D	600		
D	705		
D	705		
C			
C			
B			
B			
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR; ANGULAR;		FINISH: DEBURR AND BREAK SHARP EDGES DO NOT SCALE DRAWING REVISION	
DRAWN	NAME	SIGNATURE	DATE
CHKD			
APV'D			
MFG			
QA			
		MATERIAL: Steel DWG NO. TITLE: L Beam Steel Beams 20mm variant for MRGBAD001 Sheet 2 v2 A4 WEIGHT: SCALE: 1:10 SHEET 1 OF 1	
4	3	2	1

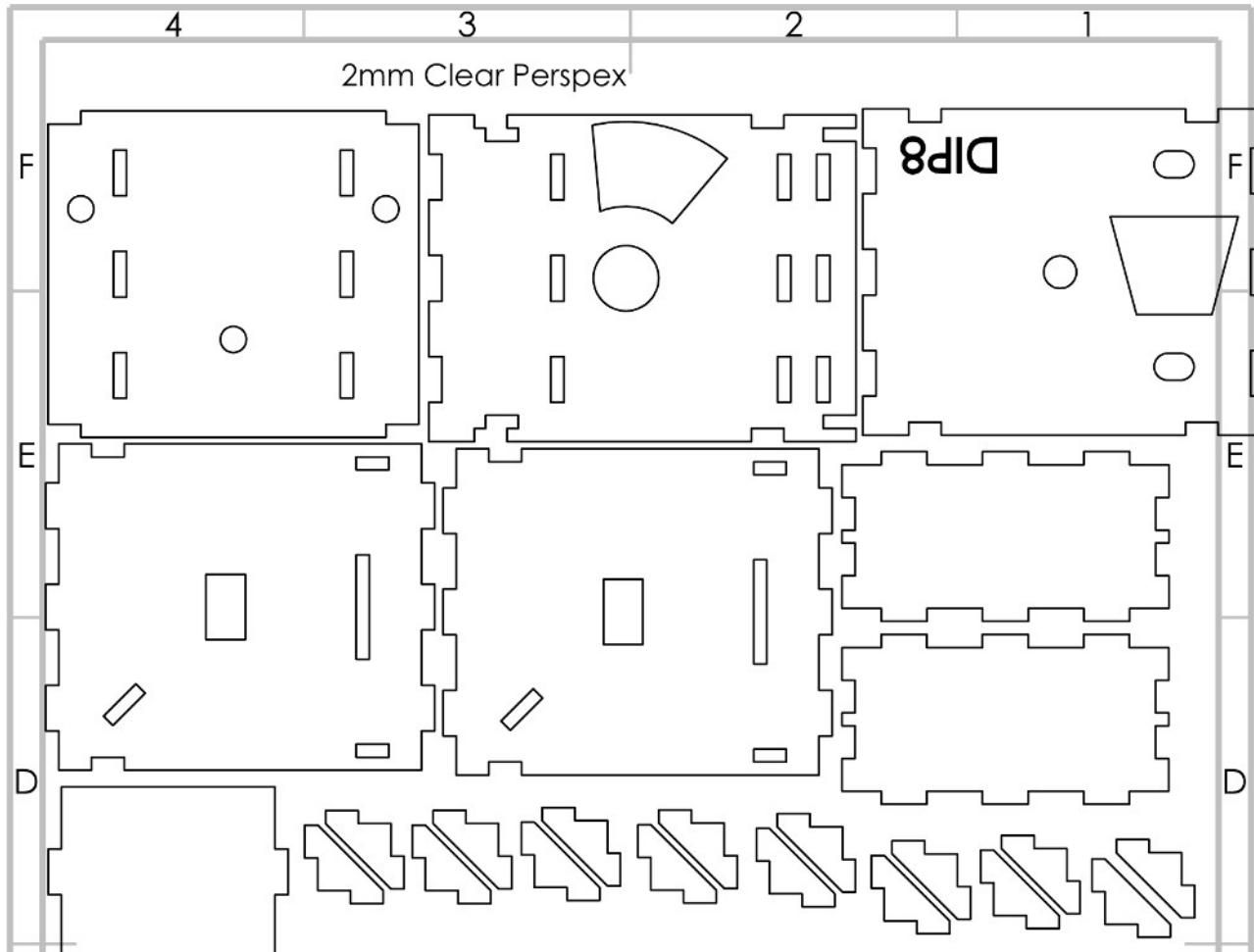
MATERIAL REQUEST FORMS

				4	3	2	1
F							F
E							E
D							D
C							C
B							B
<p>20mm wide steel bars</p> <p>Total of 3269mm (3.269m) 3.4m requested</p>				1055	1088	1088	
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
A	NAME	SIGNATURE	DATE		TITLE:	Steel bars for MRGBAD001 Sheet 3 v2	
DRAWN					DWG NO.		
CHK'D							
APP'D							
MFG							
Q.A.							
				MATERIAL:	Steel		
				WEIGHT:		SCALE:1:5	
						SHEET 1 OF 1	
	4	3	2	1			A4

MATERIAL REQUEST FORMS

				4	3	2	1	
Wooden planks with thickness of 17 or less								
F								F
E								E
D								D
C								C
B								B
A								A
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
DRAWN	NAME	SIGNATURE	DATE					
CHK'D								
APP'D								
MFG								
Q.A.								
				MATERIAL:		DWG NO.		
				MDF or Plywood				
				WEIGHT:		SCALE: 1:10	Sheet 4	
							A4	
							SHEET 1 OF 1	
				4	3	2	1	

MATERIAL REQUEST FORMS



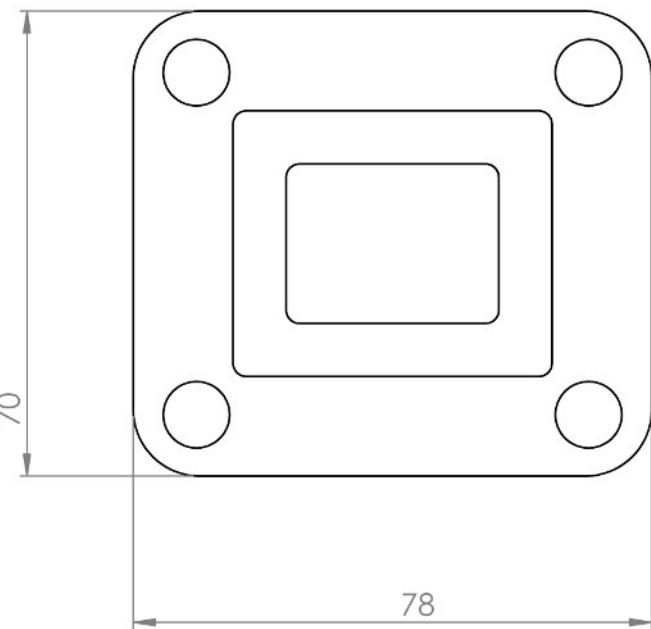
Each Part needs just over half an A4 sheet of perspex with 6 parts needed
that is 4 sheets of A4 2mm clear perspex needed

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
DRAWN	NAME	SIGNATURE	DATE				
CHK'D							
APPV'D							
MFG							
QA							
MATERIAL: 2MM CLEAR PERSPEX					DWG NO.	TITLE: 2mm Clear Perspex for MRGBAD001	
WEIGHT:					SCALE:1:1	sheet 5	
						SHEET 1 OF 1	

MATERIAL REQUEST FORMS

	4	3	2	1	
	1mm White Perspex				
F					F
E					E
D					D
C					C
B					B
<small>UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:</small>		<small>FINISH:</small>		<small>DEBURR AND BREAK SHARP EDGES</small>	<small>DO NOT SCALE DRAWING</small>
					<small>REVISION</small>
A	NAME	SIGNATURE	DATE		<small>TITLE:</small> 1mm White Perspex for MRGBAD001
DRAWN					<small>DWG NO.</small>
CHK'D					
APP'D					
MFG					
QA					
		<small>MATERIAL:</small> Non Transparent 1mm White Perspex			
		<small>WEIGHT:</small>		<small>SCALE:1:1</small>	<small>sheet 6</small>
					<small>A4</small> <small>SHEET 1 OF 1</small>
	4	3	2	1	

MATERIAL REQUEST FORMS

	4	3	2	1		
F	<p>3mm Non Transparent White Perspex</p> 				F	
E					E	
D					D	
C	<p>3mm sheet of Non transparent White perspex 90x90</p>				C	
B					B	
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
DRAWN	NAME	SIGNATURE	DATE		TITLE: 3mm White Perspex for MRGBAD001	
CHK'D						
APP'D					DWG NO.	
MFG						
Q.A.					Sheet 7	
A			MATERIAL: 3mm Non Transparent WhitePerspex		A4	
			WEIGHT:	SCALE:1:1	SHEET 1 OF 1	
	4	3	2	1		

Appendix E Website

E.1 Components Store Page

The screenshot shows the homepage of the UCT White Lab Vending Machine website. At the top, there is a circular logo featuring the University of Cape Town's crest in the center, surrounded by the text "University of Cape Town" in English and Afrikaans. Below the logo, there is a collection of various electronic components, including resistors, capacitors, integrated circuits (e.g., LM358N, LM393, LM741), and transistors. A navigation bar at the bottom left includes links for "Home", "Contact", and "UCT".

UCT White Lab Vending Machine

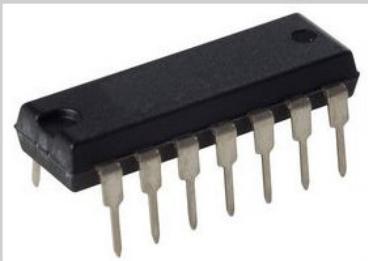
How to use: Simply use the drop down boxes to select the quantity of components you want of each type(maximim of 6 ie the first 6 you pick, the rest will be ignored). Once you have selected your components enter your student number in the field at the bottom of the page and hit submit(Double check it, if it is incorrect you will not get your components). Once submitted proceed to the Vending Machine in White lab and swipe your student card on the same day as ordering to collect your components. Note orders are reset at midnight each day.

Please note you are limited to 3 components of the same type per order, and 6 components in total per order every 24h.

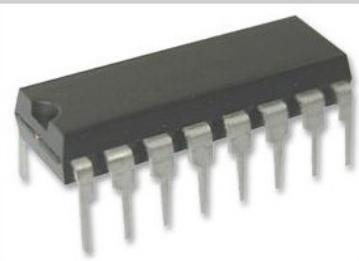
Components:

555 Timer Datasheet quantity: 0	LM311 Datasheet Out of Stock	LM393 Datasheet Out of Stock
LM741 Datasheet Out of Stock	LM358 Datasheet Out of Stock	LM339 Datasheet quantity: 0

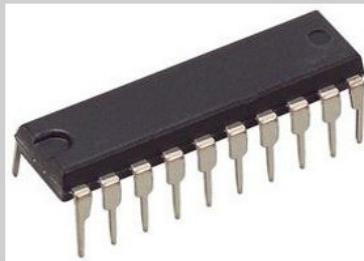
LM324

[Datasheet](#) | Out of Stock

SN7475

[Datasheet](#) | Out of Stock

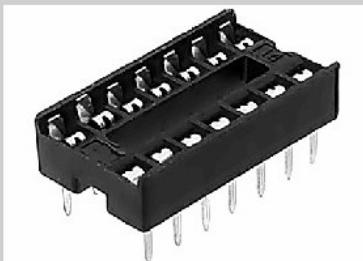
T74LS241B1

[Datasheet](#) | Out of Stock

DIP 8 IC Holder

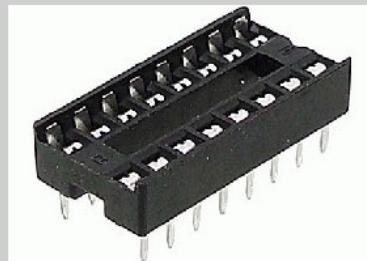
quantity:

DIP 14 IC Holder



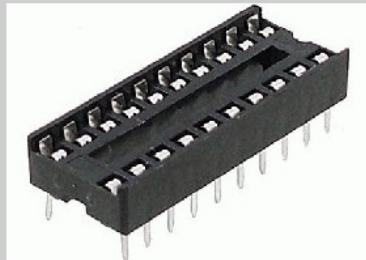
Out of Stock

DIP 16 IC Holder



Out of Stock

DIP 20 IC Holder

quantity: 

*Student Number

submit

E.2 Contacts Page



Home Contact UCT

Contacts Page

Contact us for a query, problems with the vending machine, suggestions or to just say hello.

Please fill in all the details below

*Name

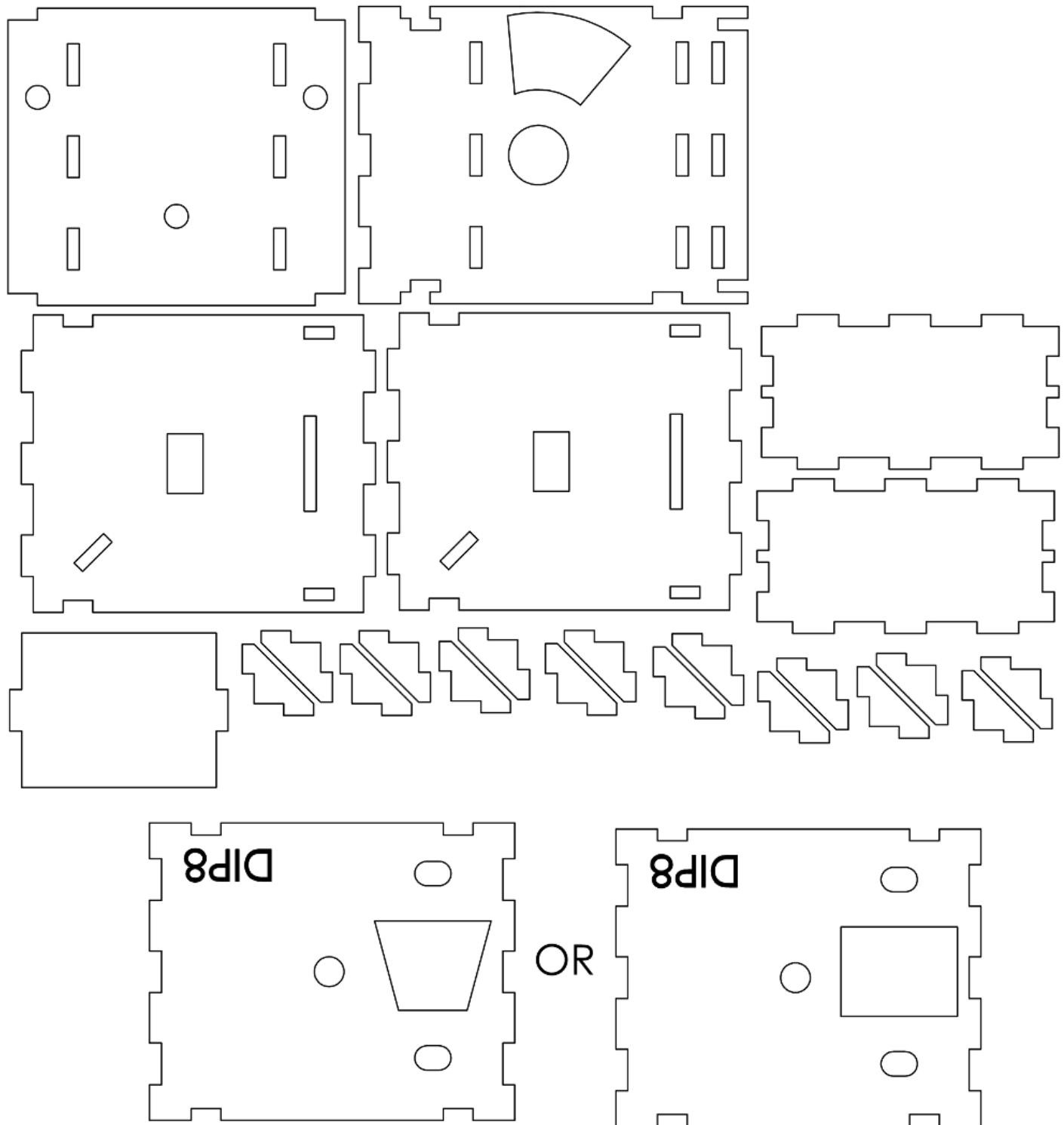
*Student Number

*Email

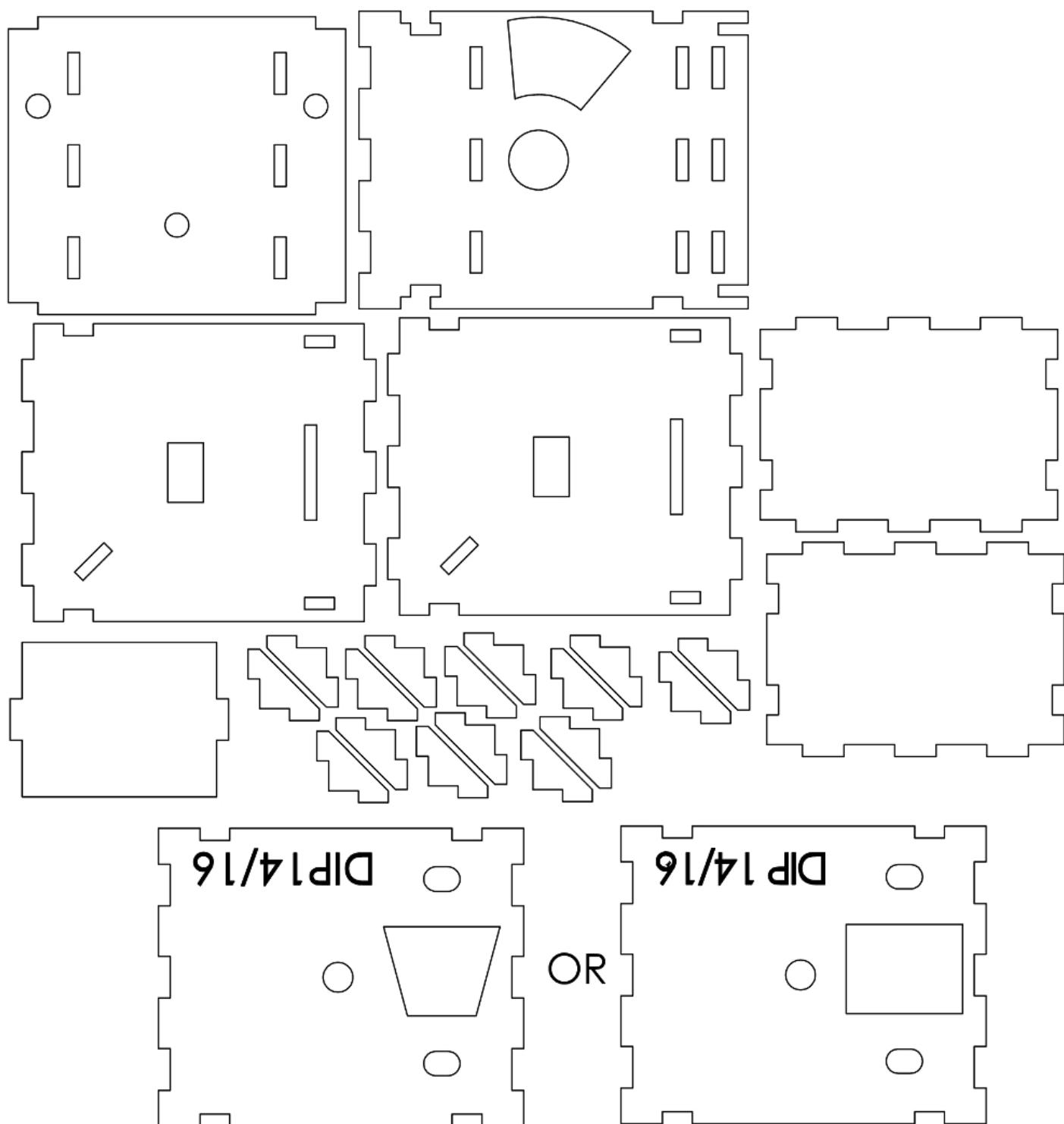
*Message

Appendix F Templates & Drawings for Assembly

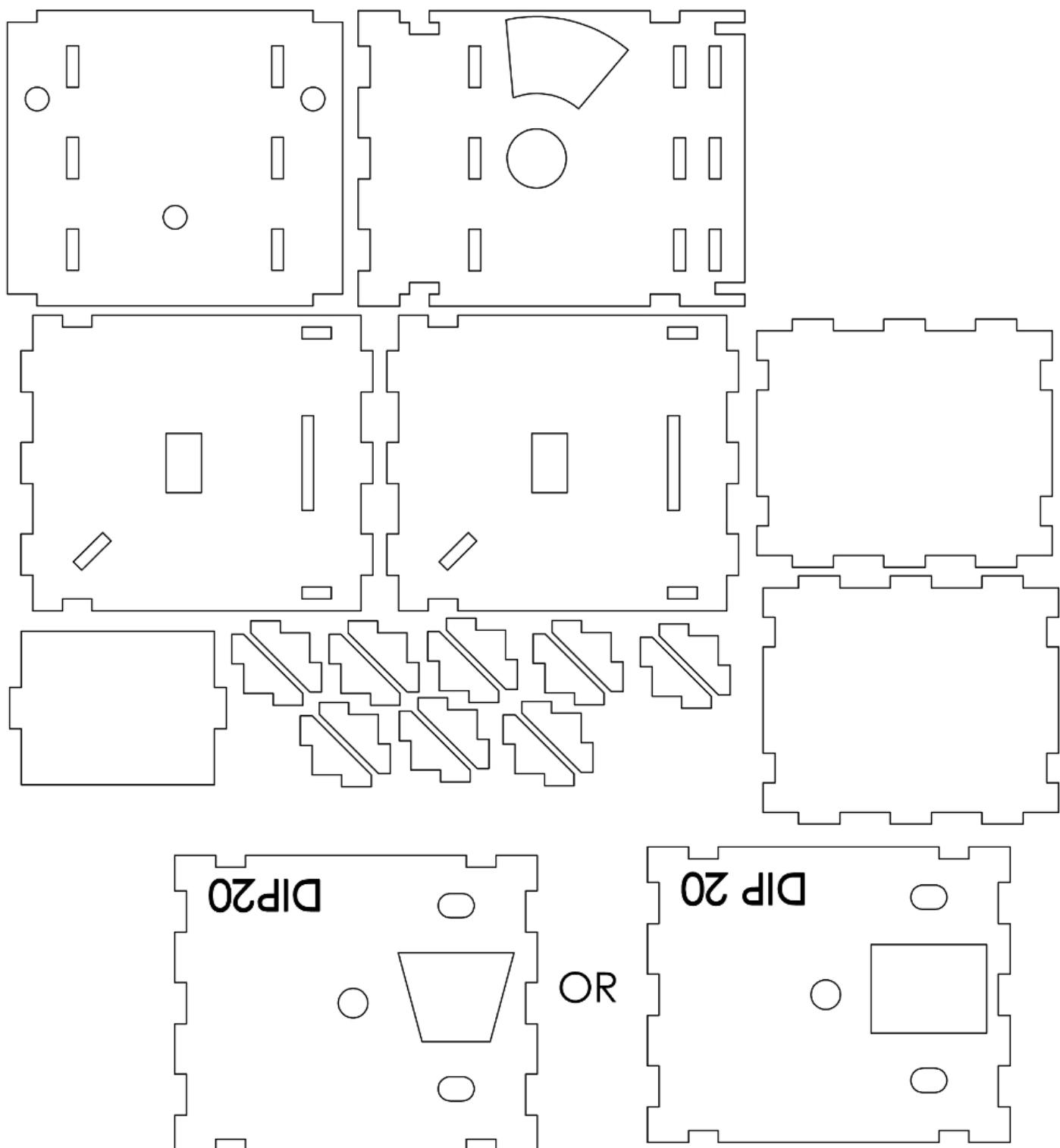
F.1 DIP 8 Template



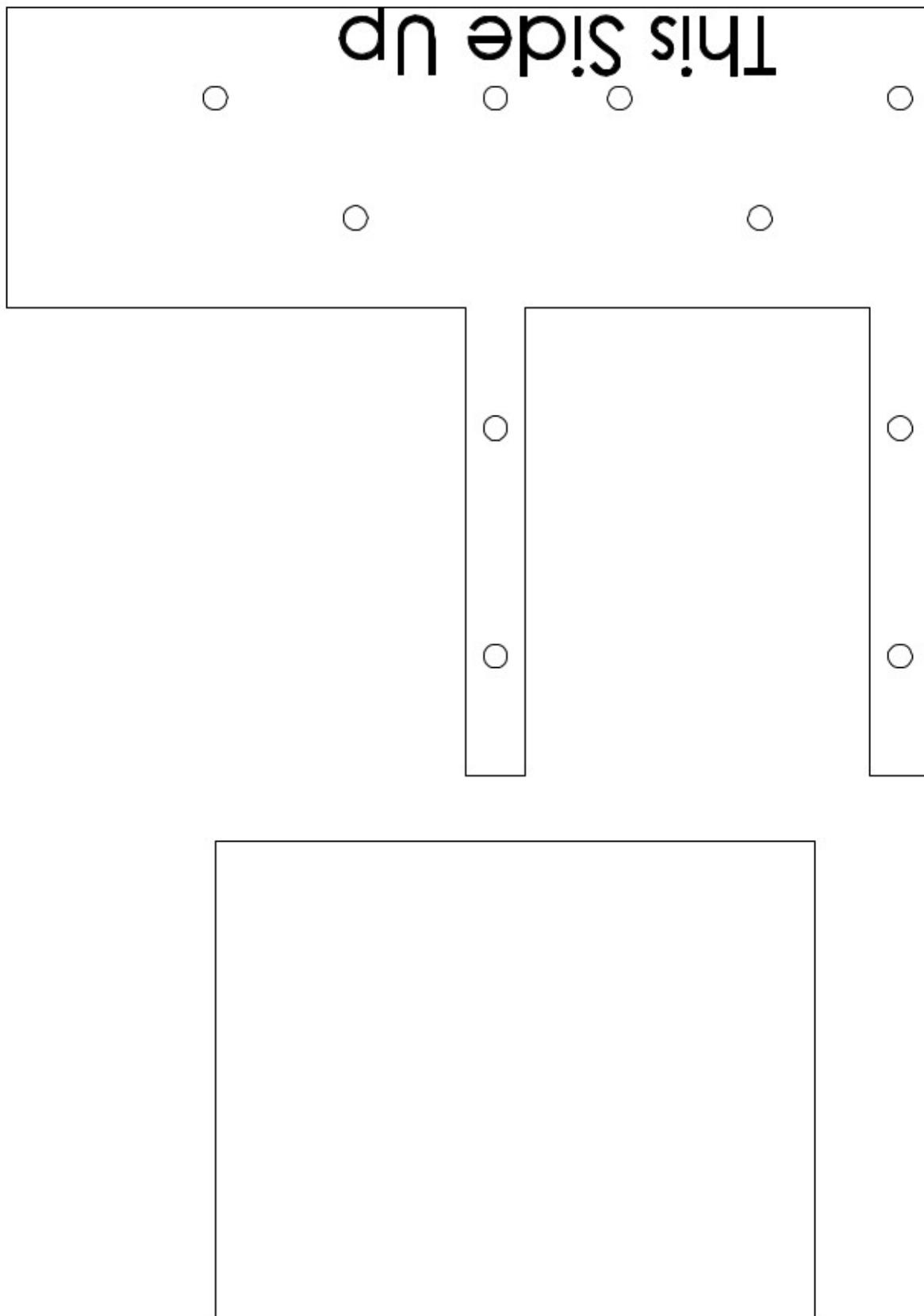
F.2 DIP 14 or 16 Template



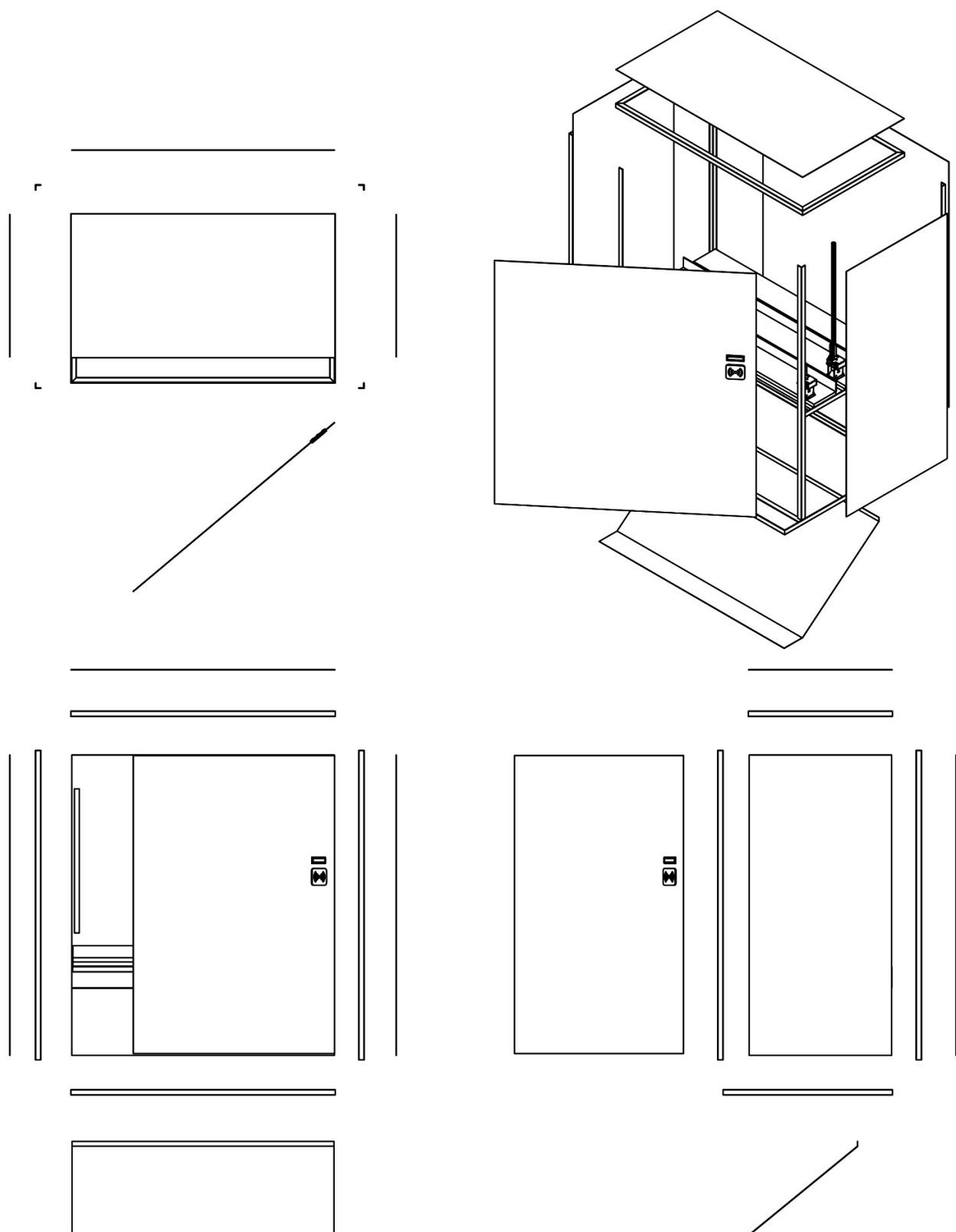
F.3 DIP 20 Template



F.4 Drill Template



F.5 Exploded Enclosure View



Appendix G Master Program Log

11-11 18:50 root	DEBUG	=====
11-11 18:50 root	DEBUG	Program Started
11-11 18:50 root	DEBUG	Reading card...
11-11 18:50 root	DEBUG	Flushing extra reads...
11-11 18:50 root	DEBUG	Done!
11-11 18:50 root	DEBUG	searching db
11-11 18:50 root	DEBUG	freeing up ics
11-11 18:50 root	DEBUG	Flushing extra reads...
11-11 18:50 root	DEBUG	Done!
11-11 18:50 root	DEBUG	display failed Failed
11-11 18:50 root	DEBUG	retrying LCD
11-11 18:50 root	DEBUG	Displayed Successfully
11-11 18:50 root	DEBUG	free
11-11 18:50 root	DEBUG	requesting
11-11 18:50 root	DEBUG	retrying LCD
11-11 18:50 root	DEBUG	Displayed Successfully
11-11 18:50 root	DEBUG	Dispensing
11-11 18:51 root	DEBUG	dispense failed, retrying
11-11 18:51 root	DEBUG	Dispensed Successfully
11-11 18:51 root	DEBUG	Finished Dispensing
11-11 18:51 root	DEBUG	Displayed Successfully
		Demo Log Starts here
11-11 18:53 root	DEBUG	=====
11-11 18:53 root	DEBUG	Program Started
11-11 18:55 root	DEBUG	Reading card...
11-11 18:55 root	DEBUG	Flushing extra reads...
11-11 18:55 root	DEBUG	Done!
11-11 18:55 root	DEBUG	searching db
11-11 18:55 root	DEBUG	requesting
11-11 18:55 root	DEBUG	Flushing extra reads...
11-11 18:55 root	DEBUG	Done!
11-11 18:55 root	DEBUG	display failed Failed
11-11 18:55 root	DEBUG	retrying LCD
11-11 18:55 root	DEBUG	Displayed Successfully
11-11 18:55 root	DEBUG	Dispensing
11-11 18:56 root	DEBUG	dispense failed, retrying
11-11 18:56 root	DEBUG	Dispensed Successfully
11-11 18:56 root	DEBUG	Dispensing
11-11 18:56 root	DEBUG	Dispensed Successfully
11-11 18:56 root	DEBUG	Dispensing
11-11 18:56 root	DEBUG	Dispensed Successfully
11-11 18:56 root	DEBUG	Dispensing
11-11 18:56 root	DEBUG	Dispensed Successfully
11-11 18:56 root	DEBUG	Dispensing
11-11 18:56 root	DEBUG	dispense failed, retrying
11-11 18:56 root	DEBUG	dispense failed, retrying
11-11 18:56 root	DEBUG	dispense failed, retrying
11-11 18:56 root	DEBUG	Dispensed Successfully
11-11 18:56 root	DEBUG	Finished Dispensing
11-11 18:56 root	DEBUG	Displayed Successfully

Appendix H Manuals and Guides

H.1 Programming Guide

This instruction set is to help program the microcontroller for the delivery mechanism that it required, it is presented in list form and should be performed in order. This list should be updated as the project progresses

1. Before programming the STM make sure all the voltage levels on the terminals are correct and the PCB has been assembled correctly.
2. Open EMBlocks.
3. Open the MCU_Module project.
4. Open the main.c file under Sources/src in the menu on the left.
5. At the top on line 19 you will find a definition of macros, uncomment “#define _EXTRA_” and “# define _DEBUG_”.
6. Uncomment the type of delivery mechanism being used. For example, if using the horizontal roller delivery mechanism uncomment “#define _ROLLER_”.
7. Plug in the STM32F051C6 to the computer.
8. Upload the code to the MCU.
9. Optional Calibration:
 - (a) Use an FTDI UART to USB convert or equivalent to connect the USART2 port to the computer for debug output.
 - (b) Take the terminal short off the “Mode” terminal and put one on the terminal next to the potentiometer.
 - (c) Use the Calibration potentiometer to calibrate and update the values for dispensary actions.
 - (d) Play with this action to see what values change to learn what the action does. It is fairly self-explanatory once observed.
 - (e) For the roller dispensary mechanism the first calibration will be the release value. Once the right value has been obtained through the serial monitor and aligning the bucket with the hole in the housing structure, go to definitions.h under Headers/src and go to line 30 and enter the new value for “releasevalINIT”.
 - (f) After 30s calibration for the pickup value will begin.
 - (g) Record the value on the serial monitor and enter it as the new value for “pickupINIT” in definitions.h on line 31.
 - (h) It will change between release and pickup calibration every 15s so if either is missed the first time, just wait for it to go back.
 - (i) Place terminal short back on “Mode” terminal and remove the short on the terminal next to the potentiometer in this order.
10. Test the dispensary action with the 8th address switch on the DIP switch.
11. Plug into the RS485 bus and do a test dispense from the master.

H.2 Restocking and Maintenance Guide

Restocking:

1. Open the Component Vending Machine's door.
2. Take out a shelf or two to get to the shelf needing to be restocked.
3. Tilt the shelf to be restocked 90°towards yourself.
4. Slide the empty rail out of the guide.
5. Take a new rail and unplug one of the ends.
6. Slide the new rail into the guide and tilt the shelf back to its original position.

administrator settings:

The administrator settings can be found in the websites files and in the Master program. This is located on the Raspberry Pi as follows:

```
sudo nano /var/www/html/includes/variables.php
```

In this file you will see the following:

```
<?php
$EmailAdmin = "bmorgan.noodles@gmail.com";
$partlimit = 3;
$complimit = 6;
?>
```

Change EmailAdmin to change the contact email address for the administrator. Change partlimit in order to change the number of non unique components that can be ordered. Change complimit to change the number of components in total that can be ordered.

Additionally, a limit in the Master program is applied to insure the maximum limit of total components is not exceeded by one student on any one day. This is on line **80 of the Master program; VMMaster.py as “complimit”**.

Lastly, The administrator list on the Master program is on **line 83** in an array called “**admins**”. Add or remove administrators through this list. They will have the ability to free up modules from an empty or jammed status.

Assemble and Add New Modules:

1. Assemble the Delivery mechanism:
2. (a) Get the corresponding design cut using one of the templates in Appendix F.
 (b) 3D print the IC guide; subsection B.1 or subsection B.2, and a roller; subsection B.6.
 (c) Assemble as per subsection 7.4 on page no. 51.
 (d) When connecting the gap sensor sockets use the test board referenced in Figure 7.8.
 (e) detailed pictures of the delivery mechanism has been provided in subsection B.5
3. Assemble the MCU Module:

4. (a) Order the PCB through a well known manufacturer.
- (b) Assemble as per subsubsection 7.3.2 on page 49 using the diagram subsection C.2 on page 76.
5. Cut out a plate of wood as per the rectangular template subsection F.4 on page 91.
6. Drill holes in this plate that line up with the holes on the MCU module.
7. Attach the MCU module using 3 10mm metal standoffs.
8. Take the shelf that the new module will be added too.
9. Drill holes on the shelf, make sure there are 2 extra sets of holes on the shelf. This is to ensure the template can be used to make more holes for more modules without removing already in place modules.
10. Attach the new module using bolts and nuts to hold it in place.
11. Attach the wooden plate with the MCU module on it.
12. Connect the sensors and motor to the corresponding headers, use the other existing modules as a template.

Appendix I Terminal Commands

This Appendix serves to present the commands used to create a functional LAMP server to run the website on a Linux based system. The commands are listed in the order they should be executed:

Update software repository:

```
1 sudo apt-get update  
2 suod apt-get upgrade
```

Install Apache web server:

```
3 sudo apt-get install apache2 -y
```

Type in IP address of Pi into the browser to test the Apache server. An Apache information page should appear on the screen. If successful, install PHP compiler and Apache plugin:

```
4 sudo apt-get install php5 libapache2-mod-php5 -y
```

Test the PHP compiler and plugin:

```
5 sudo nano index.php  
6 <?php echo "hello world"; ?>
```

Press Ctrl + X then Y then Enter to save the new file.

```
7 sudo rm index.html  
8 sudo service apache2 restart
```

Wait for the Apache instance to restart. Type in the IP address of Pi into browser to test the PHP plugin. One should see plain text saying “hello world”. Next the MySQL database manager & PHP plugin for MySQL must be installed and the Apache instance must be restarted. While installing the MySQL database manager, the installer will prompt the user for input, follow the instructions on screen:p

```
9 sudo apt-get install mysql-server php5-mysql -y  
10 sudo service apache2 restart
```

Create a database with the name UCTVendingMachine with two tables; Orders and Components with columns specified in the commands below:

```
11 mysql -uroot -prootpassword  
12 CREATE DATABASE UCTVendingMachine;  
13 use UCTVendingMachine;  
14 CREATE TABLE Orders (StudentNo TINYTEXT NOT NULL,  
    PartName TINYTEXT NOT NULL, Quantity TINYINT  
    UNSIGNED NOT NULL, Date DATE NOT NULL, Done  
    TINYINT UNSIGNED NOT NULL, ID INT UNSIGNED NOT  
    NULL);  
15 CREATE TABLE Components (Address TINYINT UNSIGNED  
    NOT NULL, PartName TINYTEXT NOT NULL, Jam  
    TINYINT UNSIGNED NOT NULL, Empty TINYINT  
    UNSIGNED NOT NULL, Low TINYINT UNSIGNED NOT NULL  
    );
```

Press CTRL+C to exit the MySQL shell. Download the Github repository containing the code needed to run the website and redirect the Apache web server to the new HomePage:

TERMINAL COMMANDS

```
16 cd /var/www/  
17 sudo rm -r html  
18 sudo git clone https://github.com/BadenMorgan/  
    VMWebsite.git html  
19 sudo nano /etc/apache2/apache2.conf
```

Add the following to the end of this configuration file:

```
20 <IfModule mod_dir.c>  
21 DirectoryIndex HomePage.php  
22 </IfModule>
```

CTRL+X then Y and Enter to save and exit. Go to the IP in your browser and play around with the website, try to test if it inputs into the databases. To install the mail server to enable administrator notifications via email the following is done:

```
23 sudo apt-get install exim4  
24 sudo dpkg-reconfigure exim4-config
```

Follow the on screen instructions. First select “mail sent by smarthost; received via SMTP or fetchmail”. Leave system mail name as “raspberrypi”. Leave IP as “127.0.0.1 ; ::1”. Leave the next field at “raspberrypi” and the next blank. For outgoing smarthost, type “smtp.gmail.com:587”. Hide local mail name: no. Keep number of DNS: no. Delivery method: “Maildir format in home directory”. Split configuration files: no. Root and postmaster mail recipient: “pi”. Now update the email credentials:

```
25 sudo nano /etc/exim4/passwd.client
```

Add the following to the end of the file with the correct email details, this serves as a template, for gmail accounts, and does not contain actual login details. Note that special characters should be avoided:

```
26 gmail-smtp.l.google.com:email@gmail.com:pass  
27 *.google.com:email@gmail.com:pass  
28 smtp.gmail.com:email@gmail.com:pass
```

Ctrl+x then Y and enter to save and exit. Send a test email on the contacts page of the website. Add the required python libraries and change the permissions for the python script:

```
29 sudo apt-get install python3-dev  
30 sudo apt-get install freetds-dev  
31 sudo pip3 install PyMySQL pymssql httplib2  
32 sudo chmod 775 /var/www/html/VMMaster.py
```

Move the boot time script that will run the python script at start up and give it the proper permissions. Next configure the Systemd to run the boot time script. Reboot to test the script:

```
33 sudo mv /var/www/html/myscript.service /lib/systemd  
    /system/myscript.service  
34 sudo systemctl daemon-reload  
35 sudo systemctl enable myscript.service  
36 sudo reboot now
```

Check the status of the task and the python scripts log file:

```
37 sudo systemctl status myscript.service  
38 tail /var/www/html/MasterLog.log
```

Appendix J Ethics Forms

Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

APPLICATION FORM

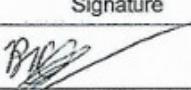
Please Note:

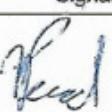
Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form before collecting or analysing data. The objective of submitting this application prior to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook**(available from the UCT EBE, Research Ethics website) prior to completing this application form: <http://www.ebe.uct.ac.za/usr/ebe/research/ethics.pdf>

APPLICANT'S DETAILS			
Name of principal researcher, student or external applicant		Baden David Morgan	
Department		Electrical Engineering	
Preferred email address of applicant:		Mrgbad001@myuct.ac.za	
If a Student	Your Degree: e.g., MSc, PhD, etc.,	Bachelor of Science in Engineering in Mechatronics	
	Name of Supervisor (if supervised):	Justin Pead	
If this is a researchcontract, indicate the source of funding/sponsorship		NA	
Project Title		Design of a White Lab Component Vending Machine	

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

SIGNED BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Baden David Morgan		14-09-16

APPLICATION APPROVED BY	Full name	Signature	Date
Supervisor (where applicable)	Justin Pead		13-09-16
HOD (or delegated nominee) <small>Final authority for all applicants who have answered NO to all questions in Section1; and for all Undergraduate research (Including Honours).</small>	E Boje		11/11/16
Chair : Faculty EIR Committee <small>For applicants other than undergraduate students who have answered YES to any of the above questions.</small>			